

Smart Car Parking Management System

Generated by Doxygen 1.9.1

Author: **Mr. Mohamed Ali Haoufa**

Documentation version: 1.0

Date: September 21th, 2024

@Copyright (c) 2024. All rights reserved.

1 PIR-Sensor-Library-for-Proteus	1
2 Module Index	3
2.1 Modules	3
3 Class Index	9
3.1 Class List	9
4 File Index	13
4.1 File List	13
5 Module Documentation	15
5.1 AT24C1024B EEPROM Driver	15
5.1.1 Detailed Description	15
5.2 EEPROM Constants	15
5.2.1 Detailed Description	16
5.2.2 Macro Definition Documentation	16
5.2.2.1 EEPROM_Slave_address	16
5.3 API Function Prototypes	16
5.3.1 Detailed Description	16
5.3.2 Function Documentation	16
5.3.2.1 EEPROM_Init()	17
5.3.2.2 EEPROM_Read_byte()	17
5.3.2.3 EEPROM_Write_NBytes()	18
5.4 KEYPAD Driver	18
5.4.1 Detailed Description	19
5.4.2 Macro Definition Documentation	19
5.4.2.1 KEYPAD_PORT	19
5.5 API Function Prototypes	20
5.5.1 Detailed Description	20
5.5.2 Function Documentation	20
5.5.2.1 keypad_Get_Pressed_Key()	20
5.5.2.2 keypad_init()	21
5.6 LCD Driver	22
5.6.1 Detailed Description	22
5.6.2 Enumeration Type Documentation	22
5.6.2.1 LCD_MODE_t	22
5.6.2.2 LCD_ROWS_t	23
5.7 LCD Commands	23
5.7.1 Detailed Description	24
5.8 LCD Row Positions	24
5.8.1 Detailed Description	25
5.9 API Function Prototypes	25
5.9.1 Detailed Description	25

5.9.2 Function Documentation	25
5.9.2.1 LCD_Init()	25
5.9.2.2 LCD_Send_Char()	26
5.9.2.3 LCD_Send_Char_Pos()	27
5.9.2.4 LCD_Send_Command()	28
5.9.2.5 LCD_Send_Enable_Signal()	28
5.9.2.6 LCD_Send_String()	29
5.9.2.7 LCD_Send_String_Pos()	30
5.9.2.8 LCD_Set_Cursor()	30
5.10 LED Driver	31
5.10.1 Detailed Description	32
5.11 Enum for specifying the LED mode.	32
5.11.1 Detailed Description	33
5.11.2 Enumeration Type Documentation	33
5.11.2.1 LED_Mode_t	33
5.12 Structure for configuring an LED.	33
5.12.1 Detailed Description	33
5.13 API Function Prototypes	33
5.13.1 Detailed Description	34
5.13.2 Function Documentation	34
5.13.2.1 LED_Init()	34
5.13.2.2 LED_Toggle()	35
5.13.2.3 LED_TurnOff()	36
5.13.2.4 LED_TurnOn()	36
5.14 SERVO MOTOR Driver	37
5.14.1 Detailed Description	37
5.15 Macros for the servo motor directions	38
5.15.1 Detailed Description	38
5.16 API Function Prototypes	38
5.16.1 Detailed Description	39
5.16.2 Function Documentation	39
5.16.2.1 Servo1_Entry_Gate()	39
5.16.2.2 Servo1_Entry_Gate_Init()	40
5.16.2.3 Servo2_Exit_Gate()	40
5.16.2.4 Servo2_Exit_Gate_Init()	41
5.17 CMSIS Global Defines	42
5.18 Defines and Type Definitions	42
5.18.1 Detailed Description	42
5.19 Status and Control Registers	43
5.19.1 Detailed Description	44
5.19.2 Macro Definition Documentation	44
5.19.2.1 APSR_C_Msk	44

5.19.2.2 APSR_C_Pos	44
5.19.2.3 APSR_GE_Msk	44
5.19.2.4 APSR_GE_Pos	45
5.19.2.5 APSR_N_Msk	45
5.19.2.6 APSR_N_Pos	45
5.19.2.7 APSR_Q_Msk	45
5.19.2.8 APSR_Q_Pos	45
5.19.2.9 APSR_V_Msk	45
5.19.2.10 APSR_V_Pos	45
5.19.2.11 APSR_Z_Msk	45
5.19.2.12 APSR_Z_Pos	46
5.19.2.13 CONTROL_FPCA_Msk	46
5.19.2.14 CONTROL_FPCA_Pos	46
5.19.2.15 CONTROL_nPRIV_Msk	46
5.19.2.16 CONTROL_nPRIV_Pos	46
5.19.2.17 CONTROL_SPSEL_Msk	46
5.19.2.18 CONTROL_SPSEL_Pos	46
5.19.2.19 IPSR_ISR_Msk	46
5.19.2.20 IPSR_ISR_Pos	47
5.19.2.21 xPSR_C_Msk	47
5.19.2.22 xPSR_C_Pos	47
5.19.2.23 xPSR_GE_Msk	47
5.19.2.24 xPSR_GE_Pos	47
5.19.2.25 xPSR_ISR_Msk	47
5.19.2.26 xPSR_ISR_Pos	47
5.19.2.27 xPSR_IT_Msk	47
5.19.2.28 xPSR_IT_Pos	48
5.19.2.29 xPSR_N_Msk	48
5.19.2.30 xPSR_N_Pos	48
5.19.2.31 xPSR_Q_Msk	48
5.19.2.32 xPSR_Q_Pos	48
5.19.2.33 xPSR_T_Msk	48
5.19.2.34 xPSR_T_Pos	48
5.19.2.35 xPSR_V_Msk	48
5.19.2.36 xPSR_V_Pos	49
5.19.2.37 xPSR_Z_Msk	49
5.19.2.38 xPSR_Z_Pos	49
5.20 Nested Vectored Interrupt Controller (NVIC)	49
5.20.1 Detailed Description	50
5.20.2 Macro Definition Documentation	50
5.20.2.1 NVIC_STIR_INTID_Msk	50
5.20.2.2 NVIC_STIR_INTID_Pos	50

5.21 System Control Block (SCB)	50
5.21.1 Detailed Description	53
5.21.2 Macro Definition Documentation	53
5.21.2.1 SCB_AIRCR_ENDIANESS_Msk	53
5.21.2.2 SCB_AIRCR_ENDIANESS_Pos	53
5.21.2.3 SCB_AIRCR_PRIGROUP_Msk	53
5.21.2.4 SCB_AIRCR_PRIGROUP_Pos	53
5.21.2.5 SCB_AIRCR_SYSRESETREQ_Msk	54
5.21.2.6 SCB_AIRCR_SYSRESETREQ_Pos	54
5.21.2.7 SCB_AIRCR_VECTCLRACTIVE_Msk	54
5.21.2.8 SCB_AIRCR_VECTCLRACTIVE_Pos	54
5.21.2.9 SCB_AIRCR_VECTKEY_Msk	54
5.21.2.10 SCB_AIRCR_VECTKEY_Pos	54
5.21.2.11 SCB_AIRCR_VECTKEYSTAT_Msk	54
5.21.2.12 SCB_AIRCR_VECTKEYSTAT_Pos	54
5.21.2.13 SCB_AIRCR_VECTRESET_Msk	55
5.21.2.14 SCB_AIRCR_VECTRESET_Pos	55
5.21.2.15 SCB_CCR_BFHFMIGN_Msk	55
5.21.2.16 SCB_CCR_BFHFMIGN_Pos	55
5.21.2.17 SCB_CCR_DIV_0_TRP_Msk	55
5.21.2.18 SCB_CCR_DIV_0_TRP_Pos	55
5.21.2.19 SCB_CCR_NONBASETHRDENA_Msk	55
5.21.2.20 SCB_CCR_NONBASETHRDENA_Pos	55
5.21.2.21 SCB_CCR_STKALIGN_Msk	56
5.21.2.22 SCB_CCR_STKALIGN_Pos	56
5.21.2.23 SCB_CCR_UNALIGN_TRP_Msk	56
5.21.2.24 SCB_CCR_UNALIGN_TRP_Pos	56
5.21.2.25 SCB_CCR_USERSETMPEND_Msk	56
5.21.2.26 SCB_CCR_USERSETMPEND_Pos	56
5.21.2.27 SCB_CFSR_BUSFAULTSR_Msk	56
5.21.2.28 SCB_CFSR_BUSFAULTSR_Pos	56
5.21.2.29 SCB_CFSR_MEMFAULTSR_Msk	57
5.21.2.30 SCB_CFSR_MEMFAULTSR_Pos	57
5.21.2.31 SCB_CFSR_USGFAULTSR_Msk	57
5.21.2.32 SCB_CFSR_USGFAULTSR_Pos	57
5.21.2.33 SCB_CPUID_ARCHITECTURE_Msk	57
5.21.2.34 SCB_CPUID_ARCHITECTURE_Pos	57
5.21.2.35 SCB_CPUID_IMPLEMENTER_Msk	57
5.21.2.36 SCB_CPUID_IMPLEMENTER_Pos	57
5.21.2.37 SCB_CPUID_PARTNO_Msk	58
5.21.2.38 SCB_CPUID_PARTNO_Pos	58
5.21.2.39 SCB_CPUID_REVISION_Msk	58

5.21.2.40 SCB_CPUID_REVISION_Pos	58
5.21.2.41 SCB_CPUID_VARIANT_Msk	58
5.21.2.42 SCB_CPUID_VARIANT_Pos	58
5.21.2.43 SCB_DFSR_BKPT_Msk	58
5.21.2.44 SCB_DFSR_BKPT_Pos	58
5.21.2.45 SCB_DFSR_DWTTRAP_Msk	59
5.21.2.46 SCB_DFSR_DWTTRAP_Pos	59
5.21.2.47 SCB_DFSR_EXTERNAL_Msk	59
5.21.2.48 SCB_DFSR_EXTERNAL_Pos	59
5.21.2.49 SCB_DFSR_HALTED_Msk	59
5.21.2.50 SCB_DFSR_HALTED_Pos	59
5.21.2.51 SCB_DFSR_VCATCH_Msk	59
5.21.2.52 SCB_DFSR_VCATCH_Pos	59
5.21.2.53 SCB_HFSR_DEBUGEVT_Msk	60
5.21.2.54 SCB_HFSR_DEBUGEVT_Pos	60
5.21.2.55 SCB_HFSR_FORCED_Msk	60
5.21.2.56 SCB_HFSR_FORCED_Pos	60
5.21.2.57 SCB_HFSR_VECTTBL_Msk	60
5.21.2.58 SCB_HFSR_VECTTBL_Pos	60
5.21.2.59 SCB_ICSR_ISRPENDING_Msk	60
5.21.2.60 SCB_ICSR_ISRPENDING_Pos	60
5.21.2.61 SCB_ICSR_ISRPREEMPT_Msk	61
5.21.2.62 SCB_ICSR_ISRPREEMPT_Pos	61
5.21.2.63 SCB_ICSR_NMIPENDSET_Msk	61
5.21.2.64 SCB_ICSR_NMIPENDSET_Pos	61
5.21.2.65 SCB_ICSR_PENDSTCLR_Msk	61
5.21.2.66 SCB_ICSR_PENDSTCLR_Pos	61
5.21.2.67 SCB_ICSR_PENDSTSET_Msk	61
5.21.2.68 SCB_ICSR_PENDSTSET_Pos	61
5.21.2.69 SCB_ICSR_PENDSVCLR_Msk	62
5.21.2.70 SCB_ICSR_PENDSVCLR_Pos	62
5.21.2.71 SCB_ICSR_PENDSVSET_Msk	62
5.21.2.72 SCB_ICSR_PENDSVSET_Pos	62
5.21.2.73 SCB_ICSR_RETTOBASE_Msk	62
5.21.2.74 SCB_ICSR_RETTOBASE_Pos	62
5.21.2.75 SCB_ICSR_VECTACTIVE_Msk	62
5.21.2.76 SCB_ICSR_VECTACTIVE_Pos	62
5.21.2.77 SCB_ICSR_VECTPENDING_Msk	63
5.21.2.78 SCB_ICSR_VECTPENDING_Pos	63
5.21.2.79 SCB_SCR_SEVONPEND_Msk	63
5.21.2.80 SCB_SCR_SEVONPEND_Pos	63
5.21.2.81 SCB_SCR_SLEEPDEEP_Msk	63

5.21.2.82 SCB_SCR_SLEEPDEEP_Pos	63
5.21.2.83 SCB_SCR_SLEEPONEXIT_Msk	63
5.21.2.84 SCB_SCR_SLEEPONEXIT_Pos	63
5.21.2.85 SCB_SHCSR_BUSFAULTACT_Msk	64
5.21.2.86 SCB_SHCSR_BUSFAULTACT_Pos	64
5.21.2.87 SCB_SHCSR_BUSFAULTENA_Msk	64
5.21.2.88 SCB_SHCSR_BUSFAULTENA_Pos	64
5.21.2.89 SCB_SHCSR_BUSFAULTPENDED_Msk	64
5.21.2.90 SCB_SHCSR_BUSFAULTPENDED_Pos	64
5.21.2.91 SCB_SHCSR_MEMFAULTACT_Msk	64
5.21.2.92 SCB_SHCSR_MEMFAULTACT_Pos	64
5.21.2.93 SCB_SHCSR_MEMFAULTENA_Msk	65
5.21.2.94 SCB_SHCSR_MEMFAULTENA_Pos	65
5.21.2.95 SCB_SHCSR_MEMFAULTPENDED_Msk	65
5.21.2.96 SCB_SHCSR_MEMFAULTPENDED_Pos	65
5.21.2.97 SCB_SHCSR_MONITORACT_Msk	65
5.21.2.98 SCB_SHCSR_MONITORACT_Pos	65
5.21.2.99 SCB_SHCSR_PENDSVACT_Msk	65
5.21.2.100 SCB_SHCSR_PENDSVACT_Pos	65
5.21.2.101 SCB_SHCSR_SVCALLACT_Msk	66
5.21.2.102 SCB_SHCSR_SVCALLACT_Pos	66
5.21.2.103 SCB_SHCSR_SVCALLPENDED_Msk	66
5.21.2.104 SCB_SHCSR_SVCALLPENDED_Pos	66
5.21.2.105 SCB_SHCSR_SYSTICKACT_Msk	66
5.21.2.106 SCB_SHCSR_SYSTICKACT_Pos	66
5.21.2.107 SCB_SHCSR_USGFAULTACT_Msk	66
5.21.2.108 SCB_SHCSR_USGFAULTACT_Pos	66
5.21.2.109 SCB_SHCSR_USGFAULTENA_Msk	67
5.21.2.110 SCB_SHCSR_USGFAULTENA_Pos	67
5.21.2.111 SCB_SHCSR_USGFAULTPENDED_Msk	67
5.21.2.112 SCB_SHCSR_USGFAULTPENDED_Pos	67
5.21.2.113 SCB_VTOR_TBLOFF_Msk	67
5.21.2.114 SCB_VTOR_TBLOFF_Pos	67
5.22 System Controls not in SCB (SCnSCB)	67
5.22.1 Detailed Description	68
5.22.2 Macro Definition Documentation	68
5.22.2.1 SCnSCB_ACTLR_DISDEFWBUF_Msk	68
5.22.2.2 SCnSCB_ACTLR_DISDEFWBUF_Pos	68
5.22.2.3 SCnSCB_ACTLR_DISFOLD_Msk	68
5.22.2.4 SCnSCB_ACTLR_DISFOLD_Pos	69
5.22.2.5 SCnSCB_ACTLR_DISFPCA_Msk	69
5.22.2.6 SCnSCB_ACTLR_DISFPCA_Pos	69

5.22.2.7 SCnSCB_ACTLR_DISMCYCINT_Msk	69
5.22.2.8 SCnSCB_ACTLR_DISMCYCINT_Pos	69
5.22.2.9 SCnSCB_ACTLR_DISOOFP_Msk	69
5.22.2.10 SCnSCB_ACTLR_DISOOFP_Pos	69
5.22.2.11 SCnSCB_ICTR_INTLINESNUM_Msk	69
5.22.2.12 SCnSCB_ICTR_INTLINESNUM_Pos	70
5.23 System Tick Timer (SysTick)	70
5.23.1 Detailed Description	71
5.23.2 Macro Definition Documentation	71
5.23.2.1 SysTick_CALIB_NOREF_Msk	71
5.23.2.2 SysTick_CALIB_NOREF_Pos	71
5.23.2.3 SysTick_CALIB_SKEW_Msk	71
5.23.2.4 SysTick_CALIB_SKEW_Pos	71
5.23.2.5 SysTick_CALIB_TENMS_Msk	71
5.23.2.6 SysTick_CALIB_TENMS_Pos	71
5.23.2.7 SysTick_CTRL_CLKSOURCE_Msk	72
5.23.2.8 SysTick_CTRL_CLKSOURCE_Pos	72
5.23.2.9 SysTick_CTRL_COUNTFLAG_Msk	72
5.23.2.10 SysTick_CTRL_COUNTFLAG_Pos	72
5.23.2.11 SysTick_CTRL_ENABLE_Msk	72
5.23.2.12 SysTick_CTRL_ENABLE_Pos	72
5.23.2.13 SysTick_CTRL_TICKINT_Msk	72
5.23.2.14 SysTick_CTRL_TICKINT_Pos	72
5.23.2.15 SysTick_LOAD_RELOAD_Msk	73
5.23.2.16 SysTick_LOAD_RELOAD_Pos	73
5.23.2.17 SysTick_VAL_CURRENT_Msk	73
5.23.2.18 SysTick_VAL_CURRENT_Pos	73
5.24 Instrumentation Trace Macrocell (ITM)	73
5.24.1 Detailed Description	74
5.24.2 Macro Definition Documentation	74
5.24.2.1 ITM_IMCR_INTEGRATION_Msk	74
5.24.2.2 ITM_IMCR_INTEGRATION_Pos	75
5.24.2.3 ITM_IRR_ATREADYM_Msk	75
5.24.2.4 ITM_IRR_ATREADYM_Pos	75
5.24.2.5 ITM_IWR_ATVALIDM_Msk	75
5.24.2.6 ITM_IWR_ATVALIDM_Pos	75
5.24.2.7 ITM_LSR_Access_Msk	75
5.24.2.8 ITM_LSR_Access_Pos	75
5.24.2.9 ITM_LSR_BytAcc_Msk	75
5.24.2.10 ITM_LSR_BytAcc_Pos	76
5.24.2.11 ITM_LSR_Present_Msk	76
5.24.2.12 ITM_LSR_Present_Pos	76

5.24.2.13 ITM_TCR_BUSY_Msk	76
5.24.2.14 ITM_TCR_BUSY_Pos	76
5.24.2.15 ITM_TCR_DWTENA_Msk	76
5.24.2.16 ITM_TCR_DWTENA_Pos	76
5.24.2.17 ITM_TCR_GTSFREQ_Msk	76
5.24.2.18 ITM_TCR_GTSFREQ_Pos	77
5.24.2.19 ITM_TCR_ITMENA_Msk	77
5.24.2.20 ITM_TCR_ITMENA_Pos	77
5.24.2.21 ITM_TCR_SWOENA_Msk	77
5.24.2.22 ITM_TCR_SWOENA_Pos	77
5.24.2.23 ITM_TCR_SYNCENA_Msk	77
5.24.2.24 ITM_TCR_SYNCENA_Pos	77
5.24.2.25 ITM_TCR_TraceBusID_Msk	77
5.24.2.26 ITM_TCR_TraceBusID_Pos	78
5.24.2.27 ITM_TCR_TSENA_Msk	78
5.24.2.28 ITM_TCR_TSENA_Pos	78
5.24.2.29 ITM_TCR_TSPrescale_Msk	78
5.24.2.30 ITM_TCR_TSPrescale_Pos	78
5.24.2.31 ITM_TPR_PRIVMASK_Msk	78
5.24.2.32 ITM_TPR_PRIVMASK_Pos	78
5.25 Data Watchpoint and Trace (DWT)	79
5.25.1 Detailed Description	80
5.25.2 Macro Definition Documentation	80
5.25.2.1 DWT_CPICNT_CPICNT_Msk	81
5.25.2.2 DWT_CPICNT_CPICNT_Pos	81
5.25.2.3 DWT_CTRL_CPIEVTEA_Msk	81
5.25.2.4 DWT_CTRL_CPIEVTEA_Pos	81
5.25.2.5 DWT_CTRL_CYCCNTENA_Msk	81
5.25.2.6 DWT_CTRL_CYCCNTENA_Pos	81
5.25.2.7 DWT_CTRL_CYCEVTEA_Msk	81
5.25.2.8 DWT_CTRL_CYCEVTEA_Pos	81
5.25.2.9 DWT_CTRL_CYCTAP_Msk	82
5.25.2.10 DWT_CTRL_CYCTAP_Pos	82
5.25.2.11 DWT_CTRL_EXCEVTENA_Msk	82
5.25.2.12 DWT_CTRL_EXCEVTENA_Pos	82
5.25.2.13 DWT_CTRL_EXCTRCENA_Msk	82
5.25.2.14 DWT_CTRL_EXCTRCENA_Pos	82
5.25.2.15 DWT_CTRL_FOLDEVTENA_Msk	82
5.25.2.16 DWT_CTRL_FOLDEVTENA_Pos	82
5.25.2.17 DWT_CTRL_LSUEVTENA_Msk	83
5.25.2.18 DWT_CTRL_LSUEVTENA_Pos	83
5.25.2.19 DWT_CTRL_NOCYCCNT_Msk	83

5.25.2.20 DWT_CTRL_NOCYCCNT_Pos	83
5.25.2.21 DWT_CTRL_NOEXTTRIG_Msk	83
5.25.2.22 DWT_CTRL_NOEXTTRIG_Pos	83
5.25.2.23 DWT_CTRL_NOPRFCNT_Msk	83
5.25.2.24 DWT_CTRL_NOPRFCNT_Pos	83
5.25.2.25 DWT_CTRL_NOTRCPKT_Msk	84
5.25.2.26 DWT_CTRL_NOTRCPKT_Pos	84
5.25.2.27 DWT_CTRL_NUMCOMP_Msk	84
5.25.2.28 DWT_CTRL_NUMCOMP_Pos	84
5.25.2.29 DWT_CTRL_PCSAMPLENA_Msk	84
5.25.2.30 DWT_CTRL_PCSAMPLENA_Pos	84
5.25.2.31 DWT_CTRL_POSTINIT_Msk	84
5.25.2.32 DWT_CTRL_POSTINIT_Pos	84
5.25.2.33 DWT_CTRL_POSTPRESET_Msk	85
5.25.2.34 DWT_CTRL_POSTPRESET_Pos	85
5.25.2.35 DWT_CTRL_SLEEPEVTENA_Msk	85
5.25.2.36 DWT_CTRL_SLEEPEVTENA_Pos	85
5.25.2.37 DWT_CTRL_SYNCTAP_Msk	85
5.25.2.38 DWT_CTRL_SYNCTAP_Pos	85
5.25.2.39 DWT_EXCCNT_EXCCNT_Msk	85
5.25.2.40 DWT_EXCCNT_EXCCNT_Pos	85
5.25.2.41 DWT_FOLDCNT_FOLDCNT_Msk	86
5.25.2.42 DWT_FOLDCNT_FOLDCNT_Pos	86
5.25.2.43 DWT_FUNCTION_CYCMATCH_Msk	86
5.25.2.44 DWT_FUNCTION_CYCMATCH_Pos	86
5.25.2.45 DWT_FUNCTION_DATAVADDR0_Msk	86
5.25.2.46 DWT_FUNCTION_DATAVADDR0_Pos	86
5.25.2.47 DWT_FUNCTION_DATAVADDR1_Msk	86
5.25.2.48 DWT_FUNCTION_DATAVADDR1_Pos	86
5.25.2.49 DWT_FUNCTION_DATAVMATCH_Msk	87
5.25.2.50 DWT_FUNCTION_DATAVMATCH_Pos	87
5.25.2.51 DWT_FUNCTION_DATAVSIZE_Msk	87
5.25.2.52 DWT_FUNCTION_DATAVSIZE_Pos	87
5.25.2.53 DWT_FUNCTION_EMITRANGE_Msk	87
5.25.2.54 DWT_FUNCTION_EMITRANGE_Pos	87
5.25.2.55 DWT_FUNCTION_FUNCTION_Msk	87
5.25.2.56 DWT_FUNCTION_FUNCTION_Pos	87
5.25.2.57 DWT_FUNCTION_LNK1ENA_Msk	88
5.25.2.58 DWT_FUNCTION_LNK1ENA_Pos	88
5.25.2.59 DWT_FUNCTION_MATCHED_Msk	88
5.25.2.60 DWT_FUNCTION_MATCHED_Pos	88
5.25.2.61 DWT_LSUCNT_LSUCNT_Msk	88

5.25.2.62 DWT_LSUCNT_LSUCNT_Pos	88
5.25.2.63 DWT_MASK_MASK_Msk	88
5.25.2.64 DWT_MASK_MASK_Pos	88
5.25.2.65 DWT_SLEEPCNT_SLEEPCNT_Msk	89
5.25.2.66 DWT_SLEEPCNT_SLEEPCNT_Pos	89
5.26 Trace Port Interface (TPI)	89
5.26.1 Detailed Description	91
5.26.2 Macro Definition Documentation	91
5.26.2.1 TPI_ACPR_PRESCALER_Msk	91
5.26.2.2 TPI_ACPR_PRESCALER_Pos	91
5.26.2.3 TPI_DEVID_AsynClkIn_Msk	91
5.26.2.4 TPI_DEVID_AsynClkIn_Pos	91
5.26.2.5 TPI_DEVID_MANCVALID_Msk	91
5.26.2.6 TPI_DEVID_MANCVALID_Pos	91
5.26.2.7 TPI_DEVID_MinBufSz_Msk	92
5.26.2.8 TPI_DEVID_MinBufSz_Pos	92
5.26.2.9 TPI_DEVID_NrTraceInput_Msk	92
5.26.2.10 TPI_DEVID_NrTraceInput_Pos	92
5.26.2.11 TPI_DEVID_NRZVALID_Msk	92
5.26.2.12 TPI_DEVID_NRZVALID_Pos	92
5.26.2.13 TPI_DEVID_PTINVALID_Msk	92
5.26.2.14 TPI_DEVID_PTINVALID_Pos	92
5.26.2.15 TPI_DEVTYPE_MajorType_Msk	93
5.26.2.16 TPI_DEVTYPE_MajorType_Pos	93
5.26.2.17 TPI_DEVTYPE_SubType_Msk	93
5.26.2.18 TPI_DEVTYPE_SubType_Pos	93
5.26.2.19 TPI_FFCR_EnFCont_Msk	93
5.26.2.20 TPI_FFCR_EnFCont_Pos	93
5.26.2.21 TPI_FFCR_TrigIn_Msk	93
5.26.2.22 TPI_FFCR_TrigIn_Pos	93
5.26.2.23 TPI_FFSR_FInProg_Msk	94
5.26.2.24 TPI_FFSR_FInProg_Pos	94
5.26.2.25 TPI_FFSR_FtNonStop_Msk	94
5.26.2.26 TPI_FFSR_FtNonStop_Pos	94
5.26.2.27 TPI_FFSR_FtStopped_Msk	94
5.26.2.28 TPI_FFSR_FtStopped_Pos	94
5.26.2.29 TPI_FFSR_TCPresent_Msk	94
5.26.2.30 TPI_FFSR_TCPresent_Pos	94
5.26.2.31 TPI_FIFO0_ETM0_Msk	95
5.26.2.32 TPI_FIFO0_ETM0_Pos	95
5.26.2.33 TPI_FIFO0_ETM1_Msk	95
5.26.2.34 TPI_FIFO0_ETM1_Pos	95

5.26.2.35 TPI_FIFO0_ETM2_Msk	95
5.26.2.36 TPI_FIFO0_ETM2_Pos	95
5.26.2.37 TPI_FIFO0_ETM_ATVALID_Msk	95
5.26.2.38 TPI_FIFO0_ETM_ATVALID_Pos	95
5.26.2.39 TPI_FIFO0_ETM_bytecount_Msk	96
5.26.2.40 TPI_FIFO0_ETM_bytecount_Pos	96
5.26.2.41 TPI_FIFO0_ITM_ATVALID_Msk	96
5.26.2.42 TPI_FIFO0_ITM_ATVALID_Pos	96
5.26.2.43 TPI_FIFO0_ITM_bytecount_Msk	96
5.26.2.44 TPI_FIFO0_ITM_bytecount_Pos	96
5.26.2.45 TPI_FIFO1_ETM_ATVALID_Msk	96
5.26.2.46 TPI_FIFO1_ETM_ATVALID_Pos	96
5.26.2.47 TPI_FIFO1_ETM_bytecount_Msk	97
5.26.2.48 TPI_FIFO1_ETM_bytecount_Pos	97
5.26.2.49 TPI_FIFO1_ITM0_Msk	97
5.26.2.50 TPI_FIFO1_ITM0_Pos	97
5.26.2.51 TPI_FIFO1_ITM1_Msk	97
5.26.2.52 TPI_FIFO1_ITM1_Pos	97
5.26.2.53 TPI_FIFO1_ITM2_Msk	97
5.26.2.54 TPI_FIFO1_ITM2_Pos	97
5.26.2.55 TPI_FIFO1_ITM_ATVALID_Msk	98
5.26.2.56 TPI_FIFO1_ITM_ATVALID_Pos	98
5.26.2.57 TPI_FIFO1_ITM_bytecount_Msk	98
5.26.2.58 TPI_FIFO1_ITM_bytecount_Pos	98
5.26.2.59 TPI_ITATBCTR0_ATREADY_Msk	98
5.26.2.60 TPI_ITATBCTR0_ATREADY_Pos	98
5.26.2.61 TPI_ITATBCTR2_ATREADY_Msk	98
5.26.2.62 TPI_ITATBCTR2_ATREADY_Pos	98
5.26.2.63 TPI_ITCTRL_Mode_Msk	99
5.26.2.64 TPI_ITCTRL_Mode_Pos	99
5.26.2.65 TPI_SPPR_TXMODE_Msk	99
5.26.2.66 TPI_SPPR_TXMODE_Pos	99
5.26.2.67 TPI_TRIGGER_TRIGGER_Msk	99
5.26.2.68 TPI_TRIGGER_TRIGGER_Pos	99
5.27 Core Debug Registers (CoreDebug)	99
5.27.1 Detailed Description	101
5.27.2 Macro Definition Documentation	101
5.27.2.1 CoreDebug_DCRSR_REGSEL_Msk	101
5.27.2.2 CoreDebug_DCRSR_REGSEL_Pos	101
5.27.2.3 CoreDebug_DCRSR_REGWnR_Msk	101
5.27.2.4 CoreDebug_DCRSR_REGWnR_Pos	101
5.27.2.5 CoreDebug_DEMCR_MON_EN_Msk	102

5.27.2.6 CoreDebug_DEMCR_MON_EN_Pos	102
5.27.2.7 CoreDebug_DEMCR_MON_PEND_Msk	102
5.27.2.8 CoreDebug_DEMCR_MON_PEND_Pos	102
5.27.2.9 CoreDebug_DEMCR_MON_REQ_Msk	102
5.27.2.10 CoreDebug_DEMCR_MON_REQ_Pos	102
5.27.2.11 CoreDebug_DEMCR_MON_STEP_Msk	102
5.27.2.12 CoreDebug_DEMCR_MON_STEP_Pos	102
5.27.2.13 CoreDebug_DEMCR_TRCENA_Msk	103
5.27.2.14 CoreDebug_DEMCR_TRCENA_Pos	103
5.27.2.15 CoreDebug_DEMCR_VC_BUSERR_Msk	103
5.27.2.16 CoreDebug_DEMCR_VC_BUSERR_Pos	103
5.27.2.17 CoreDebug_DEMCR_VC_CHKERR_Msk	103
5.27.2.18 CoreDebug_DEMCR_VC_CHKERR_Pos	103
5.27.2.19 CoreDebug_DEMCR_VC_CORERESET_Msk	103
5.27.2.20 CoreDebug_DEMCR_VC_CORERESET_Pos	103
5.27.2.21 CoreDebug_DEMCR_VC_HARDERR_Msk	104
5.27.2.22 CoreDebug_DEMCR_VC_HARDERR_Pos	104
5.27.2.23 CoreDebug_DEMCR_VC_INTERR_Msk	104
5.27.2.24 CoreDebug_DEMCR_VC_INTERR_Pos	104
5.27.2.25 CoreDebug_DEMCR_VC_MMERR_Msk	104
5.27.2.26 CoreDebug_DEMCR_VC_MMERR_Pos	104
5.27.2.27 CoreDebug_DEMCR_VC_NOCPERR_Msk	104
5.27.2.28 CoreDebug_DEMCR_VC_NOCPERR_Pos	104
5.27.2.29 CoreDebug_DEMCR_VC_STATERR_Msk	105
5.27.2.30 CoreDebug_DEMCR_VC_STATERR_Pos	105
5.27.2.31 CoreDebug_DHCSR_C_DEBUGEN_Msk	105
5.27.2.32 CoreDebug_DHCSR_C_DEBUGEN_Pos	105
5.27.2.33 CoreDebug_DHCSR_C_HALT_Msk	105
5.27.2.34 CoreDebug_DHCSR_C_HALT_Pos	105
5.27.2.35 CoreDebug_DHCSR_C_MASKINTS_Msk	105
5.27.2.36 CoreDebug_DHCSR_C_MASKINTS_Pos	105
5.27.2.37 CoreDebug_DHCSR_C_SNAPSTALL_Msk	106
5.27.2.38 CoreDebug_DHCSR_C_SNAPSTALL_Pos	106
5.27.2.39 CoreDebug_DHCSR_C_STEP_Msk	106
5.27.2.40 CoreDebug_DHCSR_C_STEP_Pos	106
5.27.2.41 CoreDebug_DHCSR_DBGKEY_Msk	106
5.27.2.42 CoreDebug_DHCSR_DBGKEY_Pos	106
5.27.2.43 CoreDebug_DHCSR_S_HALT_Msk	106
5.27.2.44 CoreDebug_DHCSR_S_HALT_Pos	106
5.27.2.45 CoreDebug_DHCSR_S_LOCKUP_Msk	107
5.27.2.46 CoreDebug_DHCSR_S_LOCKUP_Pos	107
5.27.2.47 CoreDebug_DHCSR_S_REGRDY_Msk	107

5.27.2.48 CoreDebug_DHCSR_S_REGRDY_Pos	107
5.27.2.49 CoreDebug_DHCSR_S_RESET_ST_Msk	107
5.27.2.50 CoreDebug_DHCSR_S_RESET_ST_Pos	107
5.27.2.51 CoreDebug_DHCSR_S_RETIRE_ST_Msk	107
5.27.2.52 CoreDebug_DHCSR_S_RETIRE_ST_Pos	107
5.27.2.53 CoreDebug_DHCSR_S_SLEEP_Msk	108
5.27.2.54 CoreDebug_DHCSR_S_SLEEP_Pos	108
5.28 Core Definitions	108
5.28.1 Detailed Description	109
5.28.2 Macro Definition Documentation	109
5.28.2.1 CoreDebug	109
5.28.2.2 CoreDebug_BASE	109
5.28.2.3 DWT	109
5.28.2.4 DWT_BASE	109
5.28.2.5 ITM	109
5.28.2.6 ITM_BASE	110
5.28.2.7 NVIC	110
5.28.2.8 NVIC_BASE	110
5.28.2.9 SCB	110
5.28.2.10 SCB_BASE	110
5.28.2.11 SCnSCB	110
5.28.2.12 SCS_BASE	110
5.28.2.13 SysTick	111
5.28.2.14 SysTick_BASE	111
5.28.2.15 TPI	111
5.28.2.16 TPI_BASE	111
5.29 Functions and Instructions Reference	111
5.29.1 Detailed Description	112
5.30 NVIC Functions	112
5.30.1 Detailed Description	113
5.30.2 Function Documentation	113
5.30.2.1 NVIC_ClearPendingIRQ()	113
5.30.2.2 NVIC_DecodePriority()	113
5.30.2.3 NVIC_DisableIRQ()	114
5.30.2.4 NVIC_EnableIRQ()	114
5.30.2.5 NVIC_EncodePriority()	114
5.30.2.6 NVIC_GetActive()	115
5.30.2.7 NVIC_GetPendingIRQ()	115
5.30.2.8 NVIC_GetPriority()	115
5.30.2.9 NVIC_GetPriorityGrouping()	116
5.30.2.10 NVIC_SetPendingIRQ()	116
5.30.2.11 NVIC_SetPriority()	116

5.30.2.12 NVIC_SetPriorityGrouping()	117
5.30.2.13 NVIC_SystemReset()	117
5.31 SysTick Functions	118
5.31.1 Detailed Description	118
5.31.2 Function Documentation	118
5.31.2.1 SysTick_Config()	118
5.32 ITM Functions	119
5.32.1 Detailed Description	123
5.32.2 Macro Definition Documentation	123
5.32.2.1 ITM_RXBUFFER_EMPTY	123
5.32.3 Function Documentation	123
5.32.3.1 ITM_CheckChar()	123
5.32.3.2 ITM_ReceiveChar()	124
5.32.3.3 ITM_SendChar()	124
5.32.4 Variable Documentation	124
5.32.4.1 _reserved0 [1/8]	124
5.32.4.2 [2/8]	125
5.32.4.3 _reserved0 [3/8]	125
5.32.4.4 [4/8]	125
5.32.4.5 _reserved0 [5/8]	125
5.32.4.6 [6/8]	125
5.32.4.7 _reserved0 [7/8]	125
5.32.4.8 [8/8]	125
5.32.4.9 _reserved1 [1/4]	126
5.32.4.10 [2/4]	126
5.32.4.11 [3/4]	126
5.32.4.12 _reserved1 [4/4]	126
5.32.4.13 ACPR	126
5.32.4.14 ACTLR	126
5.32.4.15 ADR	126
5.32.4.16 AFSR	127
5.32.4.17 AIRCR	127
5.32.4.18 [1/4]	127
5.32.4.19 [2/4]	127
5.32.4.20 [3/4]	127
5.32.4.21 [4/4]	127
5.32.4.22 BFAR	127
5.32.4.23 C [1/4]	128
5.32.4.24 [2/4]	128
5.32.4.25 C [3/4]	128
5.32.4.26 [4/4]	128
5.32.4.27 CALIB	128

5.32.4.28 CCR	128
5.32.4.29 CFSR	128
5.32.4.30 CID0	129
5.32.4.31 CID1	129
5.32.4.32 CID2	129
5.32.4.33 CID3	129
5.32.4.34 CLAIMCLR	129
5.32.4.35 CLAIMSET	129
5.32.4.36 COMP0	129
5.32.4.37 COMP1	130
5.32.4.38 COMP2	130
5.32.4.39 COMP3	130
5.32.4.40 CPACR	130
5.32.4.41 CPICNT	130
5.32.4.42 CPUID	130
5.32.4.43 CSPSR	130
5.32.4.44 CTRL [1/2]	131
5.32.4.45 CTRL [2/2]	131
5.32.4.46 CYCCNT	131
5.32.4.47 DCRDR	131
5.32.4.48 DCRSR	131
5.32.4.49 DEMCR	131
5.32.4.50 DEVID	131
5.32.4.51 DEVTYPE	132
5.32.4.52 DFR	132
5.32.4.53 DFSR	132
5.32.4.54 DHCSR	132
5.32.4.55 EXCCNT	132
5.32.4.56 FFCR	132
5.32.4.57 FFSR	132
5.32.4.58 FIFO0	133
5.32.4.59 FIFO1	133
5.32.4.60 FOLDCNT	133
5.32.4.61 FPCA [1/2]	133
5.32.4.62 [2/2]	133
5.32.4.63 FSCR	133
5.32.4.64 FUNCTION0	133
5.32.4.65 FUNCTION1	134
5.32.4.66 FUNCTION2	134
5.32.4.67 FUNCTION3	134
5.32.4.68 [1/4]	134
5.32.4.69 GE [2/4]	134

5.32.4.70 GE [3/4]	134
5.32.4.71 [4/4]	134
5.32.4.72 HFSR	135
5.32.4.73 IABR	135
5.32.4.74 ICER	135
5.32.4.75 ICPR	135
5.32.4.76 ICSR	135
5.32.4.77 ICTR	135
5.32.4.78 IMCR	135
5.32.4.79 IP	136
5.32.4.80 IRR	136
5.32.4.81 ISAR	136
5.32.4.82 ISER	136
5.32.4.83 ISPR	136
5.32.4.84 ISR [1/4]	136
5.32.4.85 [2/4]	136
5.32.4.86 ISR [3/4]	137
5.32.4.87 [4/4]	137
5.32.4.88 IT [1/2]	137
5.32.4.89 [2/2]	137
5.32.4.90 ITATBCTR0	137
5.32.4.91 ITATBCTR2	137
5.32.4.92 ITCTRL	137
5.32.4.93 ITM_RxBuffer	138
5.32.4.94 IWR	138
5.32.4.95 LAR	138
5.32.4.96 LOAD	138
5.32.4.97 LSR	138
5.32.4.98 LSUCNT	138
5.32.4.99 MASK0	138
5.32.4.100 MASK1	139
5.32.4.101 MASK2	139
5.32.4.102 MASK3	139
5.32.4.103 MMFAR	139
5.32.4.104 MMFR	139
5.32.4.105 [1/4]	139
5.32.4.106 N [2/4]	139
5.32.4.107 [3/4]	140
5.32.4.108 N [4/4]	140
5.32.4.109 nPRIV [1/2]	140
5.32.4.110 [2/2]	140
5.32.4.111 PCSR	140

5.32.4.112 PFR	140
5.32.4.113 PID0	140
5.32.4.114 PID1	141
5.32.4.115 PID2	141
5.32.4.116 PID3	141
5.32.4.117 PID4	141
5.32.4.118 PID5	141
5.32.4.119 PID6	141
5.32.4.120 PID7	141
5.32.4.121	141
5.32.4.122 [1/4]	142
5.32.4.123 Q [2/4]	142
5.32.4.124 [3/4]	142
5.32.4.125 Q [4/4]	142
5.32.4.126 SCR	142
5.32.4.127 SHCSR	142
5.32.4.128 SHP	142
5.32.4.129 SLEEPCNT	143
5.32.4.130 SPPR	143
5.32.4.131 SPSEL [1/2]	143
5.32.4.132 [2/2]	143
5.32.4.133 SSPSR	143
5.32.4.134 STIR	143
5.32.4.135 [1/2]	143
5.32.4.136 T [2/2]	144
5.32.4.137 TCR	144
5.32.4.138 TER	144
5.32.4.139 TPR	144
5.32.4.140 TRIGGER	144
5.32.4.141 u16 [1/2]	144
5.32.4.142 [2/2]	144
5.32.4.143 u32 [1/2]	145
5.32.4.144 [2/2]	145
5.32.4.145 [1/2]	145
5.32.4.146 u8 [2/2]	145
5.32.4.147 [1/4]	145
5.32.4.148 V [2/4]	145
5.32.4.149 [3/4]	145
5.32.4.150 V [4/4]	146
5.32.4.151 VAL	146
5.32.4.152 VTOR	146
5.32.4.153 w [1/4]	146

5.32.4.154 w [2/4]	146
5.32.4.155 w [3/4]	146
5.32.4.156 w [4/4]	146
5.32.4.157 Z [1/4]	147
5.32.4.158 [2/4]	147
5.32.4.159 [3/4]	147
5.32.4.160 Z [4/4]	147
5.33 CMSIS Core Register Access Functions	147
5.34 CMSIS Core Instruction Interface	148
5.35 CMSIS SIMD Intrinsics	148
5.36 Platform Standard Types	148
5.36.1 Detailed Description	149
5.36.2 Macro Definition Documentation	150
5.36.2.1 CPU_BIT_ORDER	150
5.36.2.2 CPU_BYTE_ORDER	150
5.36.2.3 CPU_TYPE	150
5.36.2.4 CPU_TYPE_16	150
5.36.2.5 CPU_TYPE_32	150
5.36.2.6 CPU_TYPE_64	150
5.36.2.7 CPU_TYPE_8	151
5.36.2.8 FALSE	151
5.36.2.9 FLAG_RESET	151
5.36.2.10 FLAG_SET	151
5.36.2.11 HIGH_BYTE_FIRST	151
5.36.2.12 LOW_BYTE_FIRST	151
5.36.2.13 LSB_FIRST	151
5.36.2.14 MSB_FIRST	152
5.36.2.15 RESET	152
5.36.2.16 SET	152
5.36.2.17 TRUE	152
5.36.3 Typedef Documentation	152
5.36.3.1 boolean	152
5.36.3.2 ConstVoidPtr	152
5.36.3.3 float32	153
5.36.3.4 float64	153
5.36.3.5 sint16	153
5.36.3.6 sint16_least	153
5.36.3.7 sint32	153
5.36.3.8 sint32_least	153
5.36.3.9 sint64	153
5.36.3.10 sint8	154
5.36.3.11 sint8_least	154

5.36.3.12 uint16	154
5.36.3.13 uint16_least	154
5.36.3.14 uint32	154
5.36.3.15 uint32_least	154
5.36.3.16 uint64	154
5.36.3.17 uint8	154
5.36.3.18 uint8_least	155
5.36.3.19 VoidPtr	155
5.36.3.20 vuint16_t	155
5.36.3.21 vuint32_t	155
5.36.3.22 vuint8_t	155
5.37 STM32F401xx MCU Header File	156
5.37.1 Detailed Description	157
5.38 Memory Base Addresses	157
5.38.1 Detailed Description	157
5.39 Cortex-M4 Peripherals Base Addresses	157
5.39.1 Detailed Description	158
5.40 AHB Peripherals Base Addresses	158
5.40.1 Detailed Description	158
5.41 APB2 Peripherals Base Addresses	158
5.41.1 Detailed Description	159
5.42 APB1 Peripherals Base Addresses	159
5.42.1 Detailed Description	160
5.43 Peripheral Register Definitions	160
5.43.1 Detailed Description	161
5.44 NVIC Register Map	161
5.44.1 Detailed Description	161
5.45 SCB Register Map	162
5.45.1 Detailed Description	162
5.46 SysTick Timer Register Map	162
5.46.1 Detailed Description	162
5.47 GPIO Register Map	163
5.47.1 Detailed Description	163
5.48 RCC Register Map	163
5.48.1 Detailed Description	163
5.49 EXTI Register Map	164
5.49.1 Detailed Description	164
5.50 SYSCFG Register Map	164
5.50.1 Detailed Description	164
5.51 USART Register Map	165
5.51.1 Detailed Description	165
5.52 SPI Register Map	165

5.52.1 Detailed Description	165
5.53 I2C Register Map	166
5.53.1 Detailed Description	166
5.54 CRC Register Map	166
5.54.1 Detailed Description	166
5.55 TIM1 Register Map	167
5.55.1 Detailed Description	167
5.56 Peripheral Instances	167
5.56.1 Detailed Description	169
5.57 NVIC Instance	169
5.57.1 Detailed Description	169
5.57.2 Macro Definition Documentation	169
5.57.2.1 NVIC	169
5.58 SCB Instance	169
5.58.1 Detailed Description	170
5.58.2 Macro Definition Documentation	170
5.58.2.1 SCB	170
5.59 SysTick Timer Instance	170
5.59.1 Detailed Description	170
5.59.2 Macro Definition Documentation	170
5.59.2.1 STK	171
5.60 GPIO Instances	171
5.60.1 Detailed Description	171
5.60.2 Macro Definition Documentation	171
5.60.2.1 GPIOA	171
5.60.2.2 GPIOB	172
5.60.2.3 GPIOC	172
5.60.2.4 GPIOD	172
5.60.2.5 GPIOE	172
5.61 RCC Instance	172
5.61.1 Detailed Description	172
5.61.2 Macro Definition Documentation	173
5.61.2.1 RCC	173
5.62 EXTI Instance	173
5.62.1 Detailed Description	173
5.62.2 Macro Definition Documentation	173
5.62.2.1 EXTI	173
5.63 SYSCFG Instance	174
5.63.1 Detailed Description	174
5.63.2 Macro Definition Documentation	174
5.63.2.1 SYSCFG	174
5.64 Timer Instances	174

5.64.1 Detailed Description	175
5.64.2 Macro Definition Documentation	175
5.64.2.1 TIM1	175
5.64.2.2 TIM2	175
5.65 USART Instances	175
5.65.1 Detailed Description	176
5.65.2 Macro Definition Documentation	176
5.65.2.1 USART1	176
5.65.2.2 USART2	176
5.65.2.3 USART6	176
5.66 SPI Instances	176
5.66.1 Detailed Description	177
5.66.2 Macro Definition Documentation	177
5.66.2.1 SPI1	177
5.66.2.2 SPI2	177
5.67 I2C Instances	177
5.67.1 Detailed Description	177
5.67.2 Macro Definition Documentation	178
5.67.2.1 I2C1	178
5.67.2.2 I2C2	178
5.68 CRC Instance	178
5.68.1 Detailed Description	178
5.68.2 Macro Definition Documentation	178
5.68.2.1 CRC	179
5.69 I2C Registers	179
5.69.1 Detailed Description	179
5.70 I2C CR1 Register Bit Definitions	180
5.70.1 Detailed Description	181
5.70.2 Macro Definition Documentation	181
5.70.2.1 I2C_CR1_ACK	181
5.70.2.2 I2C_CR1_ACK_Msk	181
5.70.2.3 I2C_CR1_ACK_Pos	182
5.70.2.4 I2C_CR1_ALERT	182
5.70.2.5 I2C_CR1_ALERT_Msk	182
5.70.2.6 I2C_CR1_ALERT_Pos	182
5.70.2.7 I2C_CR1_ENARP	182
5.70.2.8 I2C_CR1_ENARP_Msk	182
5.70.2.9 I2C_CR1_ENARP_Pos	182
5.70.2.10 I2C_CR1_ENGC	183
5.70.2.11 I2C_CR1_ENGC_Msk	183
5.70.2.12 I2C_CR1_ENGC_Pos	183
5.70.2.13 I2C_CR1_ENPEC	183

5.70.2.14 I2C_CR1_ENPEC_Msk	183
5.70.2.15 I2C_CR1_ENPEC_Pos	183
5.70.2.16 I2C_CR1_NOSTRETCH	183
5.70.2.17 I2C_CR1_NOSTRETCH_Msk	184
5.70.2.18 I2C_CR1_NOSTRETCH_Pos	184
5.70.2.19 I2C_CR1_PE	184
5.70.2.20 I2C_CR1_PE_Msk	184
5.70.2.21 I2C_CR1_PE_Pos	184
5.70.2.22 I2C_CR1_PEC	184
5.70.2.23 I2C_CR1_PEC_Msk	184
5.70.2.24 I2C_CR1_PEC_Pos	185
5.70.2.25 I2C_CR1_POS	185
5.70.2.26 I2C_CR1_POS_Msk	185
5.70.2.27 I2C_CR1_POS_Pos	185
5.70.2.28 I2C_CR1_SMBTYPE	185
5.70.2.29 I2C_CR1_SMBTYPE_Msk	185
5.70.2.30 I2C_CR1_SMBTYPE_Pos	185
5.70.2.31 I2C_CR1_SMBUS	186
5.70.2.32 I2C_CR1_SMBUS_Msk	186
5.70.2.33 I2C_CR1_SMBUS_Pos	186
5.70.2.34 I2C_CR1_START	186
5.70.2.35 I2C_CR1_START_Msk	186
5.70.2.36 I2C_CR1_START_Pos	186
5.70.2.37 I2C_CR1_STOP	186
5.70.2.38 I2C_CR1_STOP_Msk	187
5.70.2.39 I2C_CR1_STOP_Pos	187
5.70.2.40 I2C_CR1_SWRST	187
5.70.2.41 I2C_CR1_SWRST_Msk	187
5.70.2.42 I2C_CR1_SWRST_Pos	187
5.71 I2C CR2 Register Bit Definitions	187
5.71.1 Detailed Description	188
5.71.2 Macro Definition Documentation	188
5.71.2.1 I2C_CR2_DMAEN	188
5.71.2.2 I2C_CR2_DMAEN_Msk	188
5.71.2.3 I2C_CR2_DMAEN_Pos	189
5.71.2.4 I2C_CR2_FREQ	189
5.71.2.5 I2C_CR2_FREQ_Msk	189
5.71.2.6 I2C_CR2_FREQ_Pos	189
5.71.2.7 I2C_CR2_ITBUFEN	189
5.71.2.8 I2C_CR2_ITBUFEN_Msk	189
5.71.2.9 I2C_CR2_ITBUFEN_Pos	189
5.71.2.10 I2C_CR2_ITERREN	190

5.71.2.11 I2C_CR2_ITERREN_Msk	190
5.71.2.12 I2C_CR2_ITERREN_Pos	190
5.71.2.13 I2C_CR2 ITEVTEN	190
5.71.2.14 I2C_CR2 ITEVTEN_Msk	190
5.71.2.15 I2C_CR2 ITEVTEN_Pos	190
5.71.2.16 I2C_CR2_LAST	190
5.71.2.17 I2C_CR2_LAST_Msk	191
5.71.2.18 I2C_CR2_LAST_Pos	191
5.72 I2C OAR2 Register Bit Definitions	191
5.72.1 Detailed Description	191
5.72.2 Macro Definition Documentation	191
5.72.2.1 I2C_OAR2_ADD2	192
5.72.2.2 I2C_OAR2_ADD2_Msk	192
5.72.2.3 I2C_OAR2_ADD2_Pos	192
5.72.2.4 I2C_OAR2_ENDUAL	192
5.72.2.5 I2C_OAR2_ENDUAL_Msk	192
5.72.2.6 I2C_OAR2_ENDUAL_Pos	192
5.73 I2C SR1 Register Bit Definitions	192
5.73.1 Detailed Description	193
5.73.2 Macro Definition Documentation	193
5.73.2.1 I2C_SR1_ADD10	194
5.73.2.2 I2C_SR1_ADD10_Msk	194
5.73.2.3 I2C_SR1_ADD10_Pos	194
5.73.2.4 I2C_SR1_ADDR	194
5.73.2.5 I2C_SR1_ADDR_Msk	194
5.73.2.6 I2C_SR1_ADDR_Pos	194
5.73.2.7 I2C_SR1_AF	194
5.73.2.8 I2C_SR1_AF_Msk	194
5.73.2.9 I2C_SR1_AF_Pos	195
5.73.2.10 I2C_SR1_ARLO	195
5.73.2.11 I2C_SR1_ARLO_Msk	195
5.73.2.12 I2C_SR1_ARLO_Pos	195
5.73.2.13 I2C_SR1_BERR	195
5.73.2.14 I2C_SR1_BERR_Msk	195
5.73.2.15 I2C_SR1_BERR_Pos	195
5.73.2.16 I2C_SR1_BTF	195
5.73.2.17 I2C_SR1_BTF_Msk	196
5.73.2.18 I2C_SR1_BTF_Pos	196
5.73.2.19 I2C_SR1_OVR	196
5.73.2.20 I2C_SR1_OVR_Msk	196
5.73.2.21 I2C_SR1_OVR_Pos	196
5.73.2.22 I2C_SR1_PECERR	196

5.73.2.23 I2C_SR1_PECERR_Msk	196
5.73.2.24 I2C_SR1_PECERR_Pos	196
5.73.2.25 I2C_SR1_RXNE	197
5.73.2.26 I2C_SR1_RXNE_Msk	197
5.73.2.27 I2C_SR1_RXNE_Pos	197
5.73.2.28 I2C_SR1_SB	197
5.73.2.29 I2C_SR1_SB_Msk	197
5.73.2.30 I2C_SR1_SB_Pos	197
5.73.2.31 I2C_SR1_SMBALERT	197
5.73.2.32 I2C_SR1_SMBALERT_Msk	198
5.73.2.33 I2C_SR1_SMBALERT_Pos	198
5.73.2.34 I2C_SR1_STOPF	198
5.73.2.35 I2C_SR1_STOPF_Msk	198
5.73.2.36 I2C_SR1_STOPF_Pos	198
5.73.2.37 I2C_SR1_TIMEOUT	198
5.73.2.38 I2C_SR1_TIMEOUT_Msk	198
5.73.2.39 I2C_SR1_TIMEOUT_Pos	198
5.73.2.40 I2C_SR1_TXE	199
5.73.2.41 I2C_SR1_TXE_Msk	199
5.73.2.42 I2C_SR1_TXE_Pos	199
5.74 I2C SR2 Register Bit Definitions	199
5.74.1 Detailed Description	200
5.74.2 Macro Definition Documentation	200
5.74.2.1 I2C_SR2_BUSY	200
5.74.2.2 I2C_SR2_BUSY_Msk	200
5.74.2.3 I2C_SR2_BUSY_Pos	200
5.74.2.4 I2C_SR2_DUALF	200
5.74.2.5 I2C_SR2_DUALF_Msk	200
5.74.2.6 I2C_SR2_DUALF_Pos	201
5.74.2.7 I2C_SR2_GENCALL	201
5.74.2.8 I2C_SR2_GENCALL_Msk	201
5.74.2.9 I2C_SR2_GENCALL_Pos	201
5.74.2.10 I2C_SR2_MSL	201
5.74.2.11 I2C_SR2_MSL_Msk	201
5.74.2.12 I2C_SR2_MSL_Pos	201
5.74.2.13 I2C_SR2_PEC	202
5.74.2.14 I2C_SR2_PEC_Msk	202
5.74.2.15 I2C_SR2_PEC_Pos	202
5.74.2.16 I2C_SR2_SMBDEFAULT	202
5.74.2.17 I2C_SR2_SMBDEFAULT_Msk	202
5.74.2.18 I2C_SR2_SMBDEFAULT_Pos	202
5.74.2.19 I2C_SR2_SMBHOST	202

5.74.2.20 I2C_SR2_SMBHOST_Msk	202
5.74.2.21 I2C_SR2_SMBHOST_Pos	203
5.74.2.22 I2C_SR2_TRA	203
5.74.2.23 I2C_SR2_TRA_Msk	203
5.74.2.24 I2C_SR2_TRA_Pos	203
5.75 I2C CCR Register Bit Definitions	203
5.75.1 Detailed Description	204
5.75.2 Macro Definition Documentation	204
5.75.2.1 I2C_CCR_CCR	204
5.75.2.2 I2C_CCR_CCR_Msk	204
5.75.2.3 I2C_CCR_CCR_Pos	204
5.75.2.4 I2C_CCR_DUTY	204
5.75.2.5 I2C_CCR_DUTY_Msk	204
5.75.2.6 I2C_CCR_DUTY_Pos	204
5.76 I2C TRISE Register Bit Definitions	205
5.76.1 Detailed Description	205
5.76.2 Macro Definition Documentation	205
5.76.2.1 I2C_TRISE_TRISE	205
5.76.2.2 I2C_TRISE_TRISE_Msk	205
5.76.2.3 I2C_TRISE_TRISE_Pos	205
5.77 RCC Bit Position Definitions	206
5.77.1 Detailed Description	207
5.78 RCC_CR Bit Position Definitions	207
5.78.1 Detailed Description	208
5.78.2 Macro Definition Documentation	208
5.78.2.1 RCC_CR_CSSON	208
5.78.2.2 RCC_CR_HSEBYP	208
5.78.2.3 RCC_CR_HSEON	208
5.78.2.4 RCC_CR_HSERDY	208
5.78.2.5 RCC_CR_HSICAL	209
5.78.2.6 RCC_CR_HSION	209
5.78.2.7 RCC_CR_HSIRDY	209
5.78.2.8 RCC_CR_HSITRIM	209
5.78.2.9 RCC_CR_PLLI2SON	209
5.78.2.10 RCC_CR_PLLI2SRDY	209
5.78.2.11 RCC_CR_PLLON	209
5.78.2.12 RCC_CR_PLLRDY	209
5.78.2.13 RCC_CR_PLLSAION	210
5.78.2.14 RCC_CR_PLLSAIRDY	210
5.79 RCC_PLLCFGR Bit Position Definitions	210
5.79.1 Detailed Description	210
5.79.2 Macro Definition Documentation	210

5.79.2.1 RCC_PLLCFGR_PLLM	211
5.79.2.2 RCC_PLLCFGR_PLLN	211
5.79.2.3 RCC_PLLCFGR_PLLP	211
5.79.2.4 RCC_PLLCFGR_PLLQ	211
5.79.2.5 RCC_PLLCFGR_PLLSRC	211
5.80 RCC_CFGR Bit Position Definitions	211
5.80.1 Detailed Description	212
5.80.2 Macro Definition Documentation	212
5.80.2.1 RCC_CFGR_HPRE	212
5.80.2.2 RCC_CFGR_I2SSRC	212
5.80.2.3 RCC_CFGR_MCO1	212
5.80.2.4 RCC_CFGR_MCO1PRE	212
5.80.2.5 RCC_CFGR_MCO2	213
5.80.2.6 RCC_CFGR_MCO2PRE	213
5.80.2.7 RCC_CFGR_PPRE1	213
5.80.2.8 RCC_CFGR_PPRE2	213
5.80.2.9 RCC_CFGR_RTCPRE	213
5.80.2.10 RCC_CFGR_SW	213
5.80.2.11 RCC_CFGR_SWS	213
5.81 RCC_CIR Bit Position Definitions	214
5.81.1 Detailed Description	214
5.81.2 Macro Definition Documentation	214
5.81.2.1 RCC_CIR_CSSF	215
5.81.2.2 RCC_CIR_HSERDYC	215
5.81.2.3 RCC_CIR_HSERDYF	215
5.81.2.4 RCC_CIR_HSERDYIE	215
5.81.2.5 RCC_CIR_HSIRDYC	215
5.81.2.6 RCC_CIR_HSIRDYF	215
5.81.2.7 RCC_CIR_HSIRDYIE	215
5.81.2.8 RCC_CIR_LSERDYC	215
5.81.2.9 RCC_CIR_LSERDYF	216
5.81.2.10 RCC_CIR_LSERDYIE	216
5.81.2.11 RCC_CIR_LSIRDYC	216
5.81.2.12 RCC_CIR_LSIRDYF	216
5.81.2.13 RCC_CIR_LSIRDYIE	216
5.81.2.14 RCC_CIR_PLLI2SRDYC	216
5.81.2.15 RCC_CIR_PLLI2SRDYF	216
5.81.2.16 RCC_CIR_PLLI2SRDYIE	216
5.81.2.17 RCC_CIR_PLLRDYC	217
5.81.2.18 RCC_CIR_PLLRDYF	217
5.81.2.19 RCC_CIR_PLLRDYIE	217
5.81.2.20 RCC_CIR_PLLSAIRDYC	217

5.81.2.21 RCC_CIR_PLLSAIRDYF	217
5.81.2.22 RCC_CIR_PLLSAIRDYIE	217
5.82 RCC_AHB1RSTR Bit Position Definitions	217
5.82.1 Detailed Description	218
5.82.2 Macro Definition Documentation	218
5.82.2.1 RCC_AHB1RSTR_CRC	218
5.82.2.2 RCC_AHB1RSTR_DMA1	218
5.82.2.3 RCC_AHB1RSTR_DMA2	218
5.82.2.4 RCC_AHB1RSTR_ETHMAC	219
5.82.2.5 RCC_AHB1RSTR_GPIOA	219
5.82.2.6 RCC_AHB1RSTR_GPIOB	219
5.82.2.7 RCC_AHB1RSTR_GPIOC	219
5.82.2.8 RCC_AHB1RSTR_GPIOD	219
5.82.2.9 RCC_AHB1RSTR_GPIOE	219
5.82.2.10 RCC_AHB1RSTR_GPIOF	219
5.82.2.11 RCC_AHB1RSTR_GPIOG	219
5.82.2.12 RCC_AHB1RSTR_GPIOH	220
5.82.2.13 RCC_AHB1RSTR_GPIOI	220
5.82.2.14 RCC_AHB1RSTR_OTGHS	220
5.82.2.15 RCC_AHB1RSTR_OTGHSULPI	220
5.83 RCC_AHB2RSTR Bit Position Definitions	220
5.83.1 Detailed Description	221
5.83.2 Macro Definition Documentation	221
5.83.2.1 RCC_AHB2RSTR_CRYP	221
5.83.2.2 RCC_AHB2RSTR_DCMI	221
5.83.2.3 RCC_AHB2RSTR_HASH	221
5.83.2.4 RCC_AHB2RSTR_OTGFS	221
5.83.2.5 RCC_AHB2RSTR_RNG	221
5.84 RCC_AHB3RSTR Bit Position Definitions	222
5.84.1 Detailed Description	222
5.84.2 Macro Definition Documentation	222
5.84.2.1 RCC_AHB3RSTR_FSMC	222
5.85 RCC_APB1RSTR Bit Position Definitions	222
5.85.1 Detailed Description	223
5.85.2 Macro Definition Documentation	223
5.85.2.1 RCC_APB1RSTR_CAN1	223
5.85.2.2 RCC_APB1RSTR_CAN2	223
5.85.2.3 RCC_APB1RSTR_DAC	224
5.85.2.4 RCC_APB1RSTR_I2C1	224
5.85.2.5 RCC_APB1RSTR_I2C2	224
5.85.2.6 RCC_APB1RSTR_I2C3	224
5.85.2.7 RCC_APB1RSTR_PWR	224

5.85.2.8 RCC_APB1RSTR_SPI2	224
5.85.2.9 RCC_APB1RSTR_SPI3	224
5.85.2.10 RCC_APB1RSTR_TIM12	224
5.85.2.11 RCC_APB1RSTR_TIM13	225
5.85.2.12 RCC_APB1RSTR_TIM14	225
5.85.2.13 RCC_APB1RSTR_TIM2	225
5.85.2.14 RCC_APB1RSTR_TIM3	225
5.85.2.15 RCC_APB1RSTR_TIM4	225
5.85.2.16 RCC_APB1RSTR_TIM5	225
5.85.2.17 RCC_APB1RSTR_TIM6	225
5.85.2.18 RCC_APB1RSTR_TIM7	225
5.85.2.19 RCC_APB1RSTR_UART4	226
5.85.2.20 RCC_APB1RSTR_UART5	226
5.85.2.21 RCC_APB1RSTR_UART7	226
5.85.2.22 RCC_APB1RSTR_UART8	226
5.85.2.23 RCC_APB1RSTR_USART2	226
5.85.2.24 RCC_APB1RSTR_USART3	226
5.85.2.25 RCC_APB1RSTR_WWDG	226
5.86 RCC_APB2RSTR Bit Position Definitions	227
5.86.1 Detailed Description	227
5.86.2 Macro Definition Documentation	227
5.86.2.1 RCC_APB2RSTR_ADC	227
5.86.2.2 RCC_APB2RSTR_SDIO	227
5.86.2.3 RCC_APB2RSTR_SPI1	228
5.86.2.4 RCC_APB2RSTR_SYSCFG	228
5.86.2.5 RCC_APB2RSTR_TIM1	228
5.86.2.6 RCC_APB2RSTR_TIM10	228
5.86.2.7 RCC_APB2RSTR_TIM11	228
5.86.2.8 RCC_APB2RSTR_TIM8	228
5.86.2.9 RCC_APB2RSTR_TIM9	228
5.86.2.10 RCC_APB2RSTR_USART1	228
5.86.2.11 RCC_APB2RSTR_USART6	229
5.87 RCC_AHB1LPENR Bit Position Definitions	229
5.87.1 Detailed Description	229
5.87.2 Macro Definition Documentation	229
5.87.2.1 RCC_AHB1LPENR_CRCEN	230
5.87.2.2 RCC_AHB1LPENR_DMA1LPEN	230
5.87.2.3 RCC_AHB1LPENR_DMA2LPEN	230
5.87.2.4 RCC_AHB1LPENR_ETHMACLPEN	230
5.87.2.5 RCC_AHB1LPENR_ETHMACPTPLPEN	230
5.87.2.6 RCC_AHB1LPENR_ETHMACRXLPEN	230
5.87.2.7 RCC_AHB1LPENR_ETHMACTXLPEN	230

5.87.2.8 RCC_AHB1LPENR_GPIOALPEN	230
5.87.2.9 RCC_AHB1LPENR_GPIOBLPEN	231
5.87.2.10 RCC_AHB1LPENR_GPIOCLPEN	231
5.87.2.11 RCC_AHB1LPENR_GPIODLPEN	231
5.87.2.12 RCC_AHB1LPENR_GPIOELPEN	231
5.87.2.13 RCC_AHB1LPENR_GPIOFLPEN	231
5.87.2.14 RCC_AHB1LPENR_GPIOGLPEN	231
5.87.2.15 RCC_AHB1LPENR_GPIOHLPEN	231
5.87.2.16 RCC_AHB1LPENR_GPIOILPEN	231
5.87.2.17 RCC_AHB1LPENR_OTGHSHULPI	232
5.87.2.18 RCC_AHB1LPENR_OTGHSHPEN	232
5.88 RCC_AHB2LPENR Bit Position Definitions	232
5.88.1 Detailed Description	232
5.88.2 Macro Definition Documentation	232
5.88.2.1 RCC_AHB2LPENR_CRYPLPEN	232
5.88.2.2 RCC_AHB2LPENR_DCMILPEN	233
5.88.2.3 RCC_AHB2LPENR_HASHLPEN	233
5.88.2.4 RCC_AHB2LPENR_OTGFSLPEN	233
5.88.2.5 RCC_AHB2LPENR_RNGLPEN	233
5.89 RCC_AHB3LPENR Bit Position Definitions	233
5.89.1 Detailed Description	233
5.89.2 Macro Definition Documentation	234
5.89.2.1 RCC_AHB3LPENR_FSMCLPEN	234
5.90 RCC_APB1LPENR Bit Position Definitions	234
5.90.1 Detailed Description	235
5.90.2 Macro Definition Documentation	235
5.90.2.1 RCC_APB1LPENR_CAN1LPEN	235
5.90.2.2 RCC_APB1LPENR_CAN2LPEN	235
5.90.2.3 RCC_APB1LPENR_DACLPEN	235
5.90.2.4 RCC_APB1LPENR_I2C1LPEN	235
5.90.2.5 RCC_APB1LPENR_I2C2LPEN	235
5.90.2.6 RCC_APB1LPENR_I2C3LPEN	235
5.90.2.7 RCC_APB1LPENR_PWRLPEN	236
5.90.2.8 RCC_APB1LPENR_SPI2LPEN	236
5.90.2.9 RCC_APB1LPENR_SPI3LPEN	236
5.90.2.10 RCC_APB1LPENR_TIM12LPEN	236
5.90.2.11 RCC_APB1LPENR_TIM13LPEN	236
5.90.2.12 RCC_APB1LPENR_TIM14LPEN	236
5.90.2.13 RCC_APB1LPENR_TIM2LPEN	236
5.90.2.14 RCC_APB1LPENR_TIM3LPEN	236
5.90.2.15 RCC_APB1LPENR_TIM4LPEN	237
5.90.2.16 RCC_APB1LPENR_TIM5LPEN	237

5.90.2.17 RCC_APB1LPENR_TIM6LPEN	237
5.90.2.18 RCC_APB1LPENR_TIM7LPEN	237
5.90.2.19 RCC_APB1LPENR_UART4LPEN	237
5.90.2.20 RCC_APB1LPENR_UART5LPEN	237
5.90.2.21 RCC_APB1LPENR_UART7LPEN	237
5.90.2.22 RCC_APB1LPENR_UART8LPEN	237
5.90.2.23 RCC_APB1LPENR_USART2LPEN	238
5.90.2.24 RCC_APB1LPENR_USART3LPEN	238
5.90.2.25 RCC_APB1LPENR_WWDGLPEN	238
5.91 RCC_APB2LPENR Bit Position Definitions	238
5.91.1 Detailed Description	239
5.91.2 Macro Definition Documentation	239
5.91.2.1 RCC_APB2LPENR_ADCLPEN	239
5.91.2.2 RCC_APB2LPENR_SDIOLPEN	239
5.91.2.3 RCC_APB2LPENR_SPI1LPEN	239
5.91.2.4 RCC_APB2LPENR_SYSCFGLPEN	239
5.91.2.5 RCC_APB2LPENR_TIM10LPEN	239
5.91.2.6 RCC_APB2LPENR_TIM11LPEN	239
5.91.2.7 RCC_APB2LPENR_TIM1LPEN	240
5.91.2.8 RCC_APB2LPENR_TIM8LPEN	240
5.91.2.9 RCC_APB2LPENR_TIM9LPEN	240
5.91.2.10 RCC_APB2LPENR_USART1LPEN	240
5.91.2.11 RCC_APB2LPENR_USART6LPEN	240
5.92 RCC_BDCR Bit Position Definitions	240
5.92.1 Detailed Description	241
5.92.2 Macro Definition Documentation	241
5.92.2.1 RCC_BDCR_BDRST	241
5.92.2.2 RCC_BDCR_LSEBYP	241
5.92.2.3 RCC_BDCR_LSEON	241
5.92.2.4 RCC_BDCR_LSERDY	241
5.92.2.5 RCC_BDCR_RTCEN	241
5.92.2.6 RCC_BDCR_RTCSEL	242
5.93 RCC_CSR Bit Position Definitions	242
5.93.1 Detailed Description	242
5.93.2 Macro Definition Documentation	242
5.93.2.1 RCC_CSR_IWDGRSTF	242
5.93.2.2 RCC_CSR_LPWRRSTF	243
5.93.2.3 RCC_CSR_LSION	243
5.93.2.4 RCC_CSR_LSIRDY	243
5.93.2.5 RCC_CSR_OBLRSTF	243
5.93.2.6 RCC_CSR_PINRSTF	243
5.93.2.7 RCC_CSR_PORRSTF	243

5.93.2.8 RCC_CSR_RMVF	243
5.93.2.9 RCC_CSR_SFTRSTF	243
5.93.2.10 RCC_CSR_WWDGRSTF	244
5.94 RCC_SSCGR Bit Position Definitions	244
5.94.1 Detailed Description	244
5.94.2 Macro Definition Documentation	244
5.94.2.1 RCC_SSCGR_INCSTEP	244
5.94.2.2 RCC_SSCGR_MODPER	244
5.94.2.3 RCC_SSCGR_SPREADSEL	245
5.94.2.4 RCC_SSCGR_SSCGEN	245
5.95 RCC_PLLI2SCFGR Bit Position Definitions	245
5.95.1 Detailed Description	245
5.95.2 Macro Definition Documentation	245
5.95.2.1 RCC_PLLI2SCFGR_PLLI2SN	245
5.95.2.2 RCC_PLLI2SCFGR_PLLI2SR	246
5.96 GPIO Base Address to Code Conversion Macros	246
5.96.1 Detailed Description	246
5.96.2 Macro Definition Documentation	246
5.96.2.1 GPIO_BASEADDR_TO_CODE	246
5.97 GPIO Driver	247
5.97.1 Detailed Description	248
5.98 GPIO Pin Numbers	248
5.98.1 Detailed Description	249
5.99 GPIO Pin Modes	249
5.99.1 Detailed Description	250
5.99.2 Macro Definition Documentation	250
5.99.2.1 GPIO_MODE_AF_INPUT	250
5.99.2.2 GPIO_MODE_INPUT_FLO	250
5.99.2.3 GPIO_MODE_INPUT_PD	250
5.99.2.4 GPIO_MODE_INPUT_PU	250
5.99.2.5 GPIO_MODE_OUTPUT_AF_OD	250
5.99.2.6 GPIO_MODE_OUTPUT_AF_PP	250
5.99.2.7 GPIO_MODE_OUTPUT_OD	251
5.99.2.8 GPIO_MODE_OUTPUT_PP	251
5.100 GPIO Pin Speeds	251
5.100.1 Detailed Description	251
5.100.2 Macro Definition Documentation	251
5.100.2.1 GPIO_SPEED_10M	251
5.100.2.2 GPIO_SPEED_2M	252
5.100.2.3 GPIO_SPEED_50M	252
5.101 GPIO Pin States	252
5.101.1 Detailed Description	252

5.101.2 Macro Definition Documentation	252
5.101.2.1 GPIO_PIN_RESET	252
5.101.2.2 GPIO_PIN_SET	253
5.102 GPIO Return Lock Status	253
5.102.1 Detailed Description	253
5.102.2 Macro Definition Documentation	253
5.102.2.1 GPIO_RETURN_LOCK_ERROR	253
5.102.2.2 GPIO_RETURN_LOCK_OK	253
5.103 GPIO Pin Modes	254
5.103.1 Detailed Description	254
5.103.2 Macro Definition Documentation	254
5.103.2.1 GPIO_MODE_ALTFN	254
5.103.2.2 GPIO_MODE_ANALOG	254
5.103.2.3 GPIO_MODE_IN	255
5.103.2.4 GPIO_MODE_IT_FT	255
5.103.2.5 GPIO_MODE_IT_RFT	255
5.103.2.6 GPIO_MODE_IT_RT	255
5.103.2.7 GPIO_MODE_OUT	255
5.104 GPIO Output Speeds	255
5.104.1 Detailed Description	256
5.104.2 Macro Definition Documentation	256
5.104.2.1 GPIO_SPEED_FAST	256
5.104.2.2 GPIO_SPEED_HIGH	256
5.104.2.3 GPIO_SPEED_LOW	256
5.104.2.4 GPIO_SPEED_MEDIUM	256
5.105 GPIO Pull-up/Pull-down Configurations	257
5.105.1 Detailed Description	257
5.105.2 Macro Definition Documentation	257
5.105.2.1 GPIO_NO_PUPD	257
5.105.2.2 GPIO_PIN_PD	257
5.105.2.3 GPIO_PIN_PU	257
5.106 GPIO Output Types	258
5.106.1 Detailed Description	258
5.106.2 Macro Definition Documentation	258
5.106.2.1 GPIO_OP_TYPE_OD	258
5.106.2.2 GPIO_OP_TYPE_PP	258
5.107 API Function Prototypes	259
5.107.1 Detailed Description	259
5.107.2 Function Documentation	259
5.107.2.1 MCAL_GPIO_DelInit()	259
5.107.2.2 MCAL_GPIO_Init()	260
5.107.2.3 MCAL_GPIO_LockPin()	261

5.107.2.4 MCAL_GPIO_ReadPin()	262
5.107.2.5 MCAL_GPIO_ReadPort()	263
5.107.2.6 MCAL_GPIO_TogglePin()	264
5.107.2.7 MCAL_GPIO_TogglePort()	265
5.107.2.8 MCAL_GPIO_WritePin()	265
5.107.2.9 MCAL_GPIO_WritePort()	266
5.108 I2C Driver	267
5.108.1 Detailed Description	269
5.108.2 Enumeration Type Documentation	269
5.108.2.1 FlagStatus	269
5.108.2.2 Functional_State	269
5.108.2.3 I2C_Direction	270
5.108.2.4 Repeated_Start	270
5.108.2.5 Slave_State	270
5.108.2.6 Status	271
5.108.2.7 StopCondition	271
5.109 I2C Speed Definitions	271
5.109.1 Detailed Description	272
5.109.2 Macro Definition Documentation	272
5.109.2.1 I2C_SCK_FM_200K	272
5.109.2.2 I2C_SCK_FM_400K	272
5.109.2.3 I2C_SCK_SM_100K	272
5.109.2.4 I2C_SCK_SM_50K	272
5.110 I2C Stretch Mode Definitions	273
5.110.1 Detailed Description	273
5.110.2 Macro Definition Documentation	273
5.110.2.1 I2C_StretchMode_disabled	273
5.110.2.2 I2C_StretchMode_enabled	273
5.111 I2C Mode Definitions	273
5.111.1 Detailed Description	274
5.111.2 Macro Definition Documentation	274
5.111.2.1 I2C_Mode_I2C	274
5.111.2.2 I2C_Mode_SMBus	274
5.112 I2C Addressing Modes	274
5.112.1 Detailed Description	275
5.112.2 Macro Definition Documentation	275
5.112.2.1 I2C_Addressing_Slave_10bits	275
5.112.2.2 I2C_Addressing_Slave_7bits	275
5.113 I2C Acknowledge Control Definitions	275
5.113.1 Detailed Description	275
5.113.2 Macro Definition Documentation	276
5.113.2.1 I2C_Ack_Control_Disable	276

5.113.2.2 I2C_Ack_Control_Enable	276
5.114 I2C General Call Enable Definitions	276
5.114.1 Detailed Description	276
5.114.2 Macro Definition Documentation	276
5.114.2.1 I2C_ENGC_Disable	277
5.114.2.2 I2C_ENGC_Enable	277
5.115 API Function Prototypes	277
5.115.1 Detailed Description	278
5.115.2 Function Documentation	278
5.115.2.1 I2C_ACKnowledgeConfig()	278
5.115.2.2 I2C_Generate_Start()	278
5.115.2.3 I2C_Generate_Stop()	279
5.115.2.4 I2C_Get_FlagStatus()	279
5.115.2.5 I2C_SendAddress()	280
5.115.2.6 MCAL_I2C_Deinit()	281
5.115.2.7 MCAL_I2C_Init()	281
5.115.2.8 MCAL_I2C_Master_Rx()	282
5.115.2.9 MCAL_I2C_Master_Tx()	282
5.115.2.10 MCAL_I2C_Set_GPIO()	283
5.115.2.11 MCAL_I2C_Slave_SendData()	284
5.115.2.12 MCAL_I2C_SlaveReceiveData()	284
5.116 NVIC Driver	285
5.116.1 Detailed Description	286
5.117 NVIC Register Definitions	287
5.117.1 Detailed Description	287
5.118 NVIC Configuration Macros	287
5.118.1 Detailed Description	288
5.119 Interrupt Status	288
5.119.1 Detailed Description	288
5.120 Priority Groups	288
5.120.1 Detailed Description	289
5.121 Interrupt Priorities	289
5.121.1 Detailed Description	290
5.122 Interrupt Requests	290
5.122.1 Detailed Description	291
5.123 API Function Prototypes	291
5.123.1 Detailed Description	292
5.123.2 Function Documentation	292
5.123.2.1 MCAL_NVIC_ClearPendingIRQ()	292
5.123.2.2 MCAL_NVIC_DisableIRQ()	293
5.123.2.3 MCAL_NVIC_EnableIRQ()	293
5.123.2.4 MCAL_NVIC_GetActive()	294

5.123.2.5 MCAL_NVIC_GetPendingIRQ()	295
5.123.2.6 MCAL_NVIC_GetPriority()	295
5.123.2.7 MCAL_NVIC_GetPriorityGrouping()	296
5.123.2.8 MCAL_NVIC_SetPendingIRQ()	297
5.123.2.9 MCAL_NVIC_SetPriority()	298
5.123.2.10 MCAL_NVIC_SetPriorityGrouping()	299
5.123.2.11 MCAL_NVIC_SystemReset()	299
5.124 NVIC IRQ Enable/Disable Macros	300
5.124.1 Detailed Description	301
5.124.2 Macro Definition Documentation	301
5.124.2.1 NVIC IRQ10_EXTI4_Disable	301
5.124.2.2 NVIC IRQ10_EXTI4_Enable	301
5.124.2.3 NVIC IRQ23_EXTI5_9_Disable	302
5.124.2.4 NVIC IRQ23_EXTI5_9_Enable	302
5.124.2.5 NVIC IRQ31_I2C1_EV_Disable	302
5.124.2.6 NVIC IRQ31_I2C1_EV_Enable	302
5.124.2.7 NVIC IRQ32_I2C1_ER_Disable	302
5.124.2.8 NVIC IRQ32_I2C1_ER_Enable	302
5.124.2.9 NVIC IRQ33_I2C2_EV_Disable	302
5.124.2.10 NVIC IRQ33_I2C2_EV_Enable	303
5.124.2.11 NVIC IRQ34_I2C2_ER_Enable	303
5.124.2.12 NVIC IRQ35_SPI1_Disable	303
5.124.2.13 NVIC IRQ35_SPI1_Enable	303
5.124.2.14 NVIC IRQ36_SPI2_Enable	303
5.124.2.15 NVIC IRQ37_USART1_Disable	303
5.124.2.16 NVIC IRQ37_USART1_Enable	303
5.124.2.17 NVIC IRQ38_USART2_Disable	304
5.124.2.18 NVIC IRQ38_USART2_Enable	304
5.124.2.19 NVIC IRQ39_USART3_Enable	304
5.124.2.20 NVIC IRQ40_EXTI10_15_Enable	304
5.124.2.21 NVIC IRQ6_EXTI0_Disable	304
5.124.2.22 NVIC IRQ6_EXTI0_Enable	304
5.124.2.23 NVIC IRQ7_EXTI1_Disable	304
5.124.2.24 NVIC IRQ7_EXTI1_Enable	305
5.124.2.25 NVIC IRQ8_EXTI2_Disable	305
5.124.2.26 NVIC IRQ8_EXTI2_Enable	305
5.124.2.27 NVIC IRQ9_EXTI3_Disable	305
5.124.2.28 NVIC IRQ9_EXTI3_Enable	305
5.125 RCC Driver	305
5.125.1 Detailed Description	306
5.126 Clock Source Selection	306
5.126.1 Detailed Description	306

5.126.2 Macro Definition Documentation	306
5.126.2.1 HSE_CLK	306
5.126.2.2 HSE_VALUE	307
5.126.2.3 HSI_RC_CLK	307
5.126.2.4 HSI_VALUE	307
5.126.2.5 RCC_PLLCFGR_PLLSRC_HSE	307
5.126.2.6 RCC_SELECT_HSE	307
5.126.2.7 RCC_SELECT_HSI	307
5.126.2.8 RCC_SELECT_PLL	307
5.127 Peripheral Selection	308
5.127.1 Detailed Description	308
5.127.2 Macro Definition Documentation	308
5.127.2.1 RCC_CRC	308
5.127.2.2 RCC_GPIOA	308
5.127.2.3 RCC_GPIOB	309
5.127.2.4 RCC_GPIOC	309
5.127.2.5 RCC_GPIOD	309
5.127.2.6 RCC_GPIOE	309
5.127.2.7 RCC_I2C1	309
5.127.2.8 RCC_I2C2	309
5.127.2.9 RCC_SPI1	309
5.127.2.10 RCC_SPI2	309
5.127.2.11 RCC_SYSCFG	310
5.127.2.12 RCC_TIM2	310
5.127.2.13 RCC_USART1	310
5.127.2.14 RCC_USART2	310
5.127.2.15 RCC_USART6	310
5.128 API Function Prototypes	310
5.128.1 Detailed Description	311
5.128.2 Function Documentation	311
5.128.2.1 MCAL_RCC_Enable_Peripheral()	311
5.128.2.2 MCAL_RCC_GetHCLKFreq()	311
5.128.2.3 MCAL_RCC_GetPCLK1Freq()	312
5.128.2.4 MCAL_RCC_GetPCLK2Freq()	313
5.128.2.5 MCAL_RCC_GetSYS_CLKFreq()	313
5.128.2.6 MCAL_RCC_Reset_Peripheral()	314
5.128.2.7 MCAL_RCC_Select_Clock()	314
5.128.2.8 RCC_GetPLLOutputClock()	314
5.129 SPI Driver	315
5.129.1 Detailed Description	316
5.129.2 Enumeration Type Documentation	316
5.129.2.1 SPI_PollingMechanism	316

5.130 SPI Mode Definitions	317
5.130.1 Detailed Description	317
5.130.2 Macro Definition Documentation	317
5.130.2.1 SPI_Mode_Master	317
5.130.2.2 SPI_Mode_Slave	317
5.131 SPI Communication Mode Definitions	317
5.131.1 Detailed Description	318
5.131.2 Macro Definition Documentation	318
5.131.2.1 SPI_Direction_1line_Receive_only	318
5.131.2.2 SPI_Direction_1line_Transmit_only	318
5.131.2.3 SPI_Direction_2lines	318
5.131.2.4 SPI_Direction_2lines_RX_Only	318
5.132 SPI Payload Length Definitions	318
5.132.1 Detailed Description	319
5.132.2 Macro Definition Documentation	319
5.132.2.1 SPI_Payload_Length_16bit	319
5.132.2.2 SPI_Payload_Length_8bit	319
5.133 SPI Frame Format Definitions	319
5.133.1 Detailed Description	319
5.133.2 Macro Definition Documentation	319
5.133.2.1 SPI_Frame_Format_LSB	320
5.133.2.2 SPI_Frame_Format_MSB	320
5.134 SPI Clock Polarity Definitions	320
5.134.1 Detailed Description	320
5.134.2 Macro Definition Documentation	320
5.134.2.1 SPI_Clock_Polarity_High_Idle	320
5.134.2.2 SPI_Clock_Polarity_Low_Idle	321
5.135 SPI Clock Phase Definitions	321
5.135.1 Detailed Description	321
5.135.2 Macro Definition Documentation	321
5.135.2.1 SPI_Clock_Phase_Leading	321
5.135.2.2 SPI_Clock_Phase_Trailing	321
5.136 SPI NSS Definitions	322
5.136.1 Detailed Description	322
5.136.2 Macro Definition Documentation	322
5.136.2.1 SPI_NSS_Hard_Master_SS_No_output	322
5.136.2.2 SPI_NSS_Hard_Master_SS_Output_Enable	322
5.136.2.3 SPI_NSS_Hard_Slave	322
5.136.2.4 SPI_NSS_Internal_Soft_Reset	323
5.136.2.5 SPI_NSS_Internal_Soft_Set	323
5.137 SPI Prescaler Definitions	323
5.137.1 Detailed Description	323

5.137.2 Macro Definition Documentation	323
5.137.2.1 SPI_Prescaler_By128	323
5.137.2.2 SPI_Prescaler_By16	324
5.137.2.3 SPI_Prescaler_By2	324
5.137.2.4 SPI_Prescaler_By256	324
5.137.2.5 SPI_Prescaler_By32	324
5.137.2.6 SPI_Prescaler_By4	324
5.137.2.7 SPI_Prescaler_By64	324
5.137.2.8 SPI_Prescaler_By8	324
5.138 SPI IRQ Enable Definitions	325
5.138.1 Detailed Description	325
5.138.2 Macro Definition Documentation	325
5.138.2.1 SPI_IRQ_Enable_Err	325
5.138.2.2 SPI_IRQ_Enable_None	325
5.138.2.3 SPI_IRQ_Enable_Rx_Only	325
5.138.2.4 SPI_IRQ_Enable_Tx_Only	326
5.139 API Function Prototypes	326
5.139.1 Detailed Description	326
5.139.2 Function Documentation	326
5.139.2.1 MCAL_SPI_DelInit()	326
5.139.2.2 MCAL_SPI_GPIO_Set_Pins()	327
5.139.2.3 MCAL_SPI_Init()	328
5.139.2.4 MCAL_SPI_ReceiveData()	329
5.139.2.5 MCAL_SPI_SendData()	329
5.139.2.6 MCAL_SPI_Tx_Rx()	330
5.140 SYSTICK Driver	331
5.140.1 Detailed Description	332
5.140.2 Macro Definition Documentation	332
5.140.2.1 STK_FCPU	332
5.141 SysTick Interrupt Configuration	332
5.141.1 Detailed Description	332
5.141.2 Macro Definition Documentation	332
5.141.2.1 STK_INTERRUPT_DISABLED	332
5.141.2.2 STK_INTERRUPT_ENABLED	333
5.142 SysTick Clock Configuration	333
5.142.1 Detailed Description	333
5.142.2 Macro Definition Documentation	333
5.142.2.1 STK_CLK_AHB	333
5.142.2.2 STK_CLK_AHB_8	333
5.143 SysTick Running Mode Configuration	334
5.143.1 Detailed Description	334
5.143.2 Macro Definition Documentation	334

5.143.2.1 STK_ONE_SHOT_MODE	334
5.143.2.2 STK_PERIODIC_MODE	334
5.144 API Function Prototypes	334
5.144.1 Detailed Description	335
5.144.2 Function Documentation	335
5.144.2.1 MCAL_STK_Config()	335
5.144.2.2 MCAL_STK_Delay()	336
5.144.2.3 MCAL_STK_Delay1ms()	337
5.144.2.4 MCAL_STK_SetCallback()	337
5.144.2.5 MCAL_STK_SetReload()	338
5.144.2.6 MCAL_STK_StartTimer()	338
5.144.2.7 MCAL_STK_StopTimer()	339
5.145 TIMER Driver	339
5.145.1 Detailed Description	340
5.146 TIM2 Register group	340
5.146.1 Detailed Description	341
5.146.2 Macro Definition Documentation	341
5.146.2.1 TIM2_TIMER_BASE	341
5.147 API Function Prototypes	341
5.147.1 Detailed Description	342
5.147.2 Function Documentation	342
5.147.2.1 dms()	342
5.147.2.2 dus()	342
5.147.2.3 Timer2_init()	343
5.148 USART Driver	344
5.148.1 Detailed Description	344
5.149 Enumeration and Structures for USART	344
5.149.1 Detailed Description	345
5.150 Macros for USART Configuration	345
5.150.1 Detailed Description	346
5.151 API Function Prototypes	346
5.151.1 Detailed Description	347
5.151.2 Function Documentation	347
5.151.2.1 MCAL_USART_DeInit()	347
5.151.2.2 MCAL_USART_GPIO_Set_Pins()	347
5.151.2.3 MCAL_USART_Init()	348
5.151.2.4 MCAL_USART_ReceiveBuffer()	349
5.151.2.5 MCAL_USART_ReceiveData()	349
5.151.2.6 MCAL_USART_SendData()	349
5.151.2.7 MCAL_USART_SendString()	350
5.151.2.8 MCAL_USART_Wait_TC()	350
5.152 Application States	350

5.152.1 Detailed Description	352
5.152.2 Enumeration Type Documentation	352
5.152.2.1 STATES	352
5.152.3 Function Documentation	352
5.152.3.1 STATE_API() [1/6]	353
5.152.3.2 STATE_API() [2/6]	353
5.152.3.3 STATE_API() [3/6]	353
5.152.3.4 STATE_API() [4/6]	353
5.152.3.5 STATE_API() [5/6]	353
5.152.3.6 STATE_API() [6/6]	353
5.152.4 Variable Documentation	354
5.152.4.1 fp_App_State_Handler	354
5.152.4.2 Free_Slots	354
5.153 Includes	354
5.154 State Macros	355
5.154.1 Detailed Description	355
5.154.2 Macro Definition Documentation	355
5.154.2.1 STATE_API	355
5.154.2.2 STATE_NAME	356
5.155 Prototypes	356
5.155.1 Detailed Description	356
5.155.2 Variable Documentation	356
5.155.2.1 fp_App_State_Handler	357
5.156 State Handlers	357
5.156.1 Detailed Description	357
5.156.2 Function Documentation	358
5.156.2.1 STATE_API() [1/6]	358
5.156.2.2 STATE_API() [2/6]	358
5.156.2.3 STATE_API() [3/6]	358
5.156.2.4 STATE_API() [4/6]	358
5.156.2.5 STATE_API() [5/6]	358
5.156.2.6 STATE_API() [6/6]	359
5.157 ECU Definitions and APIs	359
5.157.1 Detailed Description	360
5.158 Enum representing the result of ID check.	360
5.158.1 Detailed Description	360
5.158.2 Enumeration Type Documentation	360
5.158.2.1 ID_Check_Result	360
5.159 Configuration Macros	360
5.159.1 Detailed Description	361
5.160 ECU APIs	361
5.160.1 Detailed Description	362

5.160.2 Function Documentation	362
5.160.2.1 Admin_Init()	363
5.160.2.2 Check_Flag()	363
5.160.2.3 Check_ID()	364
5.160.2.4 Check_Password()	364
5.160.2.5 combineArrays()	365
5.160.2.6 ECU_Init()	365
5.160.2.7 Enter_Gate_Open()	366
5.160.2.8 Exit_Gate_Open()	366
5.160.2.9 Flag_SET_RESET()	367
5.160.2.10 Trigger_Alarm()	367
5.160.2.11 UserLCD_PrintFreeSlots()	368
5.160.2.12 Wrong_RFID()	369
5.161 Functions_Declarations	369
5.161.1 Detailed Description	370
5.162 Global_Variables_Definitions	370
5.162.1 Detailed Description	370
5.162.2 Macro Definition Documentation	370
5.162.2.1 SIZE1	371
5.162.3 Variable Documentation	371
5.162.3.1 Exit_Flag	371
5.162.3.2 Free_Slots	371
5.162.3.3 Print_Slots_LCD_Flag	371
5.163 API_Definitions	371
5.163.1 Detailed Description	372
5.163.2 Function Documentation	372
5.163.2.1 Admin_Init()	372
5.163.2.2 Check_Flag()	373
5.163.2.3 Check_ID()	373
5.163.2.4 Check_Password()	374
5.163.2.5 combineArrays()	374
5.163.2.6 ECU_Init()	375
5.163.2.7 Enter_Gate_Open()	375
5.163.2.8 Enter_UART_CallBack()	375
5.163.2.9 Exit_Gate_Open()	376
5.163.2.10 Exit_UART_CallBack()	376
5.163.2.11 Flag_SET_RESET()	377
5.163.2.12 Trigger_Alarm()	377
5.163.2.13 UserLCD_PrintFreeSlots()	377
5.163.2.14 Wrong_RFID()	378
5.164 CMSIS	379
5.164.1 Detailed Description	398

5.164.1.1 MISRA-C:2004 Compliance Exceptions	398
5.164.2 Macro Definition Documentation	398
5.164.2.1 __CM4_CMSIS_VERSION	398
5.164.2.2 __CM4_CMSIS_VERSION_MAIN	398
5.164.2.3 __CM4_CMSIS_VERSION_SUB	398
5.164.2.4 __CORE_CM4_H_DEPENDANT	399
5.164.2.5 __CORTEX_M	399
5.164.2.6 __I	399
5.164.2.7 __IO	399
5.164.2.8 __O	399
5.164.2.9 APSR_C_Msk	399
5.164.2.10 APSR_C_Pos	399
5.164.2.11 APSR_GE_Msk	400
5.164.2.12 APSR_GE_Pos	400
5.164.2.13 APSR_N_Msk	400
5.164.2.14 APSR_N_Pos	400
5.164.2.15 APSR_Q_Msk	400
5.164.2.16 APSR_Q_Pos	400
5.164.2.17 APSR_V_Msk	400
5.164.2.18 APSR_V_Pos	400
5.164.2.19 APSR_Z_Msk	401
5.164.2.20 APSR_Z_Pos	401
5.164.2.21 CONTROL_FPCA_Msk	401
5.164.2.22 CONTROL_FPCA_Pos	401
5.164.2.23 CONTROL_nPRIV_Msk	401
5.164.2.24 CONTROL_nPRIV_Pos	401
5.164.2.25 CONTROL_SPSEL_Msk	401
5.164.2.26 CONTROL_SPSEL_Pos	401
5.164.2.27 CoreDebug	402
5.164.2.28 CoreDebug_BASE	402
5.164.2.29 CoreDebug_DCRSR_REGSEL_Msk	402
5.164.2.30 CoreDebug_DCRSR_REGSEL_Pos	402
5.164.2.31 CoreDebug_DCRSR_REGWnR_Msk	402
5.164.2.32 CoreDebug_DCRSR_REGWnR_Pos	402
5.164.2.33 CoreDebug DEMCR_MON_EN_Msk	402
5.164.2.34 CoreDebug DEMCR_MON_EN_Pos	403
5.164.2.35 CoreDebug DEMCR_MON_PEND_Msk	403
5.164.2.36 CoreDebug DEMCR_MON_PEND_Pos	403
5.164.2.37 CoreDebug DEMCR_MON_REQ_Msk	403
5.164.2.38 CoreDebug DEMCR_MON_REQ_Pos	403
5.164.2.39 CoreDebug DEMCR_MON_STEP_Msk	403
5.164.2.40 CoreDebug DEMCR_MON_STEP_Pos	403

5.164.2.41 CoreDebug_DEMCR_TRCENA_Msk	403
5.164.2.42 CoreDebug_DEMCR_TRCENA_Pos	404
5.164.2.43 CoreDebug_DEMCR_VC_BUSERR_Msk	404
5.164.2.44 CoreDebug_DEMCR_VC_BUSERR_Pos	404
5.164.2.45 CoreDebug_DEMCR_VC_CHKERR_Msk	404
5.164.2.46 CoreDebug_DEMCR_VC_CHKERR_Pos	404
5.164.2.47 CoreDebug_DEMCR_VC_CORERESET_Msk	404
5.164.2.48 CoreDebug_DEMCR_VC_CORERESET_Pos	404
5.164.2.49 CoreDebug_DEMCR_VC_HARDERR_Msk	404
5.164.2.50 CoreDebug_DEMCR_VC_HARDERR_Pos	405
5.164.2.51 CoreDebug_DEMCR_VC_INTERR_Msk	405
5.164.2.52 CoreDebug_DEMCR_VC_INTERR_Pos	405
5.164.2.53 CoreDebug_DEMCR_VC_MMERR_Msk	405
5.164.2.54 CoreDebug_DEMCR_VC_MMERR_Pos	405
5.164.2.55 CoreDebug_DEMCR_VC_NOCPERR_Msk	405
5.164.2.56 CoreDebug_DEMCR_VC_NOCPERR_Pos	405
5.164.2.57 CoreDebug_DEMCR_VC_STATERR_Msk	405
5.164.2.58 CoreDebug_DEMCR_VC_STATERR_Pos	406
5.164.2.59 CoreDebug_DHCSR_C_DEBUGEN_Msk	406
5.164.2.60 CoreDebug_DHCSR_C_DEBUGEN_Pos	406
5.164.2.61 CoreDebug_DHCSR_C_HALT_Msk	406
5.164.2.62 CoreDebug_DHCSR_C_HALT_Pos	406
5.164.2.63 CoreDebug_DHCSR_C_MASKINTS_Msk	406
5.164.2.64 CoreDebug_DHCSR_C_MASKINTS_Pos	406
5.164.2.65 CoreDebug_DHCSR_C_SNAPSTALL_Msk	406
5.164.2.66 CoreDebug_DHCSR_C_SNAPSTALL_Pos	407
5.164.2.67 CoreDebug_DHCSR_C_STEP_Msk	407
5.164.2.68 CoreDebug_DHCSR_C_STEP_Pos	407
5.164.2.69 CoreDebug_DHCSR_DBGKEY_Msk	407
5.164.2.70 CoreDebug_DHCSR_DBGKEY_Pos	407
5.164.2.71 CoreDebug_DHCSR_S_HALT_Msk	407
5.164.2.72 CoreDebug_DHCSR_S_HALT_Pos	407
5.164.2.73 CoreDebug_DHCSR_S_LOCKUP_Msk	407
5.164.2.74 CoreDebug_DHCSR_S_LOCKUP_Pos	408
5.164.2.75 CoreDebug_DHCSR_S_REGRDY_Msk	408
5.164.2.76 CoreDebug_DHCSR_S_REGRDY_Pos	408
5.164.2.77 CoreDebug_DHCSR_S_RESET_ST_Msk	408
5.164.2.78 CoreDebug_DHCSR_S_RESET_ST_Pos	408
5.164.2.79 CoreDebug_DHCSR_S_RETIRE_ST_Msk	408
5.164.2.80 CoreDebug_DHCSR_S_RETIRE_ST_Pos	408
5.164.2.81 CoreDebug_DHCSR_S_SLEEP_Msk	408
5.164.2.82 CoreDebug_DHCSR_S_SLEEP_Pos	409

5.164.2.83 DWT	409
5.164.2.84 DWT_BASE	409
5.164.2.85 DWT_CPICNT_CPICNT_Msk	409
5.164.2.86 DWT_CPICNT_CPICNT_Pos	409
5.164.2.87 DWT_CTRL_CPIEVTENA_Msk	409
5.164.2.88 DWT_CTRL_CPIEVTENA_Pos	409
5.164.2.89 DWT_CTRL_CYCCNTENA_Msk	410
5.164.2.90 DWT_CTRL_CYCCNTENA_Pos	410
5.164.2.91 DWT_CTRL_CYCEVTENA_Msk	410
5.164.2.92 DWT_CTRL_CYCEVTENA_Pos	410
5.164.2.93 DWT_CTRL_CYCTAP_Msk	410
5.164.2.94 DWT_CTRL_CYCTAP_Pos	410
5.164.2.95 DWT_CTRL_EXCEVTENA_Msk	410
5.164.2.96 DWT_CTRL_EXCEVTENA_Pos	410
5.164.2.97 DWT_CTRL_EXCTRCENA_Msk	411
5.164.2.98 DWT_CTRL_EXCTRCENA_Pos	411
5.164.2.99 DWT_CTRL_FOLDEVTENA_Msk	411
5.164.2.100 DWT_CTRL_FOLDEVTENA_Pos	411
5.164.2.101 DWT_CTRL_LSUEVTENA_Msk	411
5.164.2.102 DWT_CTRL_LSUEVTENA_Pos	411
5.164.2.103 DWT_CTRL_NOCYCCNT_Msk	411
5.164.2.104 DWT_CTRL_NOCYCCNT_Pos	411
5.164.2.105 DWT_CTRL_NOEXTTRIG_Msk	412
5.164.2.106 DWT_CTRL_NOEXTTRIG_Pos	412
5.164.2.107 DWT_CTRL_NOPRFCNT_Msk	412
5.164.2.108 DWT_CTRL_NOPRFCNT_Pos	412
5.164.2.109 DWT_CTRL_NOTRPCPKT_Msk	412
5.164.2.110 DWT_CTRL_NOTRPCPKT_Pos	412
5.164.2.111 DWT_CTRL_NUMCOMP_Msk	412
5.164.2.112 DWT_CTRL_NUMCOMP_Pos	412
5.164.2.113 DWT_CTRL_PCSAMPLENA_Msk	413
5.164.2.114 DWT_CTRL_PCSAMPLENA_Pos	413
5.164.2.115 DWT_CTRL_POSTINIT_Msk	413
5.164.2.116 DWT_CTRL_POSTINIT_Pos	413
5.164.2.117 DWT_CTRL_POSTPRESET_Msk	413
5.164.2.118 DWT_CTRL_POSTPRESET_Pos	413
5.164.2.119 DWT_CTRL_SLEEPEVTENA_Msk	413
5.164.2.120 DWT_CTRL_SLEEPEVTENA_Pos	413
5.164.2.121 DWT_CTRL_SYNCTAP_Msk	414
5.164.2.122 DWT_CTRL_SYNCTAP_Pos	414
5.164.2.123 DWT_EXCCNT_EXCCNT_Msk	414
5.164.2.124 DWT_EXCCNT_EXCCNT_Pos	414

5.164.2.125 DWT_FOLDCNT_FOLDCNT_Msk	414
5.164.2.126 DWT_FOLDCNT_FOLDCNT_Pos	414
5.164.2.127 DWT_FUNCTION_CYCMATCH_Msk	414
5.164.2.128 DWT_FUNCTION_CYCMATCH_Pos	414
5.164.2.129 DWT_FUNCTION_DATAVADDR0_Msk	415
5.164.2.130 DWT_FUNCTION_DATAVADDR0_Pos	415
5.164.2.131 DWT_FUNCTION_DATAVADDR1_Msk	415
5.164.2.132 DWT_FUNCTION_DATAVADDR1_Pos	415
5.164.2.133 DWT_FUNCTION_DATAVMATCH_Msk	415
5.164.2.134 DWT_FUNCTION_DATAVMATCH_Pos	415
5.164.2.135 DWT_FUNCTION_DATAVSIZE_Msk	415
5.164.2.136 DWT_FUNCTION_DATAVSIZE_Pos	415
5.164.2.137 DWT_FUNCTION_EMITRANGE_Msk	416
5.164.2.138 DWT_FUNCTION_EMITRANGE_Pos	416
5.164.2.139 DWT_FUNCTION_FUNCTION_Msk	416
5.164.2.140 DWT_FUNCTION_FUNCTION_Pos	416
5.164.2.141 DWT_FUNCTION_LNK1ENA_Msk	416
5.164.2.142 DWT_FUNCTION_LNK1ENA_Pos	416
5.164.2.143 DWT_FUNCTION_MATCHED_Msk	416
5.164.2.144 DWT_FUNCTION_MATCHED_Pos	416
5.164.2.145 DWT_LSUCNT_LSUCNT_Msk	417
5.164.2.146 DWT_LSUCNT_LSUCNT_Pos	417
5.164.2.147 DWT_MASK_MASK_Msk	417
5.164.2.148 DWT_MASK_MASK_Pos	417
5.164.2.149 DWT_SLEEPCNT_SLEEPCNT_Msk	417
5.164.2.150 DWT_SLEEPCNT_SLEEPCNT_Pos	417
5.164.2.151 IPSR_ISR_Msk	417
5.164.2.152 IPSR_ISR_Pos	417
5.164.2.153 ITM	418
5.164.2.154 ITM_BASE	418
5.164.2.155 ITM_IMCR_INTEGRATION_Msk	418
5.164.2.156 ITM_IMCR_INTEGRATION_Pos	418
5.164.2.157 ITM_IRR_ATREADYM_Msk	418
5.164.2.158 ITM_IRR_ATREADYM_Pos	418
5.164.2.159 ITM_IWR_ATVALIDM_Msk	418
5.164.2.160 ITM_IWR_ATVALIDM_Pos	419
5.164.2.161 ITM_LSR_Access_Msk	419
5.164.2.162 ITM_LSR_Access_Pos	419
5.164.2.163 ITM_LSR_BytAcc_Msk	419
5.164.2.164 ITM_LSR_BytAcc_Pos	419
5.164.2.165 ITM_LSR_Present_Msk	419
5.164.2.166 ITM_LSR_Present_Pos	419

5.164.2.167 ITM_RXBUFFER_EMPTY	419
5.164.2.168 ITM_TCR_BUSY_Msk	420
5.164.2.169 ITM_TCR_BUSY_Pos	420
5.164.2.170 ITM_TCR_DWTENA_Msk	420
5.164.2.171 ITM_TCR_DWTENA_Pos	420
5.164.2.172 ITM_TCR_GTSFREQ_Msk	420
5.164.2.173 ITM_TCR_GTSFREQ_Pos	420
5.164.2.174 ITM_TCR_ITMENA_Msk	420
5.164.2.175 ITM_TCR_ITMENA_Pos	420
5.164.2.176 ITM_TCR_SWOENA_Msk	421
5.164.2.177 ITM_TCR_SWOENA_Pos	421
5.164.2.178 ITM_TCR_SYNCENA_Msk	421
5.164.2.179 ITM_TCR_SYNCENA_Pos	421
5.164.2.180 ITM_TCR_TraceBusID_Msk	421
5.164.2.181 ITM_TCR_TraceBusID_Pos	421
5.164.2.182 ITM_TCR_TSENA_Msk	421
5.164.2.183 ITM_TCR_TSENA_Pos	421
5.164.2.184 ITM_TCR_TSPrescale_Msk	422
5.164.2.185 ITM_TCR_TSPrescale_Pos	422
5.164.2.186 ITM_TPR_PRIVMASK_Msk	422
5.164.2.187 ITM_TPR_PRIVMASK_Pos	422
5.164.2.188 NVIC	422
5.164.2.189 NVIC_BASE	422
5.164.2.190 NVIC_STIR_INTID_Msk	422
5.164.2.191 NVIC_STIR_INTID_Pos	423
5.164.2.192 SCB	423
5.164.2.193 SCB_AIRCR_ENDIANESS_Msk	423
5.164.2.194 SCB_AIRCR_ENDIANESS_Pos	423
5.164.2.195 SCB_AIRCR_PRIGROUP_Msk	423
5.164.2.196 SCB_AIRCR_PRIGROUP_Pos	423
5.164.2.197 SCB_AIRCR_SYSRESETREQ_Msk	423
5.164.2.198 SCB_AIRCR_SYSRESETREQ_Pos	424
5.164.2.199 SCB_AIRCR_VECTCLRACTIVE_Msk	424
5.164.2.200 SCB_AIRCR_VECTCLRACTIVE_Pos	424
5.164.2.201 SCB_AIRCR_VECTKEY_Msk	424
5.164.2.202 SCB_AIRCR_VECTKEY_Pos	424
5.164.2.203 SCB_AIRCR_VECTKEYSTAT_Msk	424
5.164.2.204 SCB_AIRCR_VECTKEYSTAT_Pos	424
5.164.2.205 SCB_AIRCR_VECTRESET_Msk	424
5.164.2.206 SCB_AIRCR_VECTRESET_Pos	425
5.164.2.207 SCB_BASE	425
5.164.2.208 SCB_CCR_BFHFNIGN_Msk	425

5.164.2.209 SCB_CCR_BFHFMIGN_Pos	425
5.164.2.210 SCB_CCR_DIV_0_TRP_Msk	425
5.164.2.211 SCB_CCR_DIV_0_TRP_Pos	425
5.164.2.212 SCB_CCR_NONBASETHRDENA_Msk	425
5.164.2.213 SCB_CCR_NONBASETHRDENA_Pos	426
5.164.2.214 SCB_CCR_STKALIGN_Msk	426
5.164.2.215 SCB_CCR_STKALIGN_Pos	426
5.164.2.216 SCB_CCR_UNALIGN_TRP_Msk	426
5.164.2.217 SCB_CCR_UNALIGN_TRP_Pos	426
5.164.2.218 SCB_CCR_USERSETPEND_Msk	426
5.164.2.219 SCB_CCR_USERSETPEND_Pos	426
5.164.2.220 SCB_CFSR_BUSFAULTSR_Msk	426
5.164.2.221 SCB_CFSR_BUSFAULTSR_Pos	427
5.164.2.222 SCB_CFSR_MEMFAULTSR_Msk	427
5.164.2.223 SCB_CFSR_MEMFAULTSR_Pos	427
5.164.2.224 SCB_CFSR_USGFAULTSR_Msk	427
5.164.2.225 SCB_CFSR_USGFAULTSR_Pos	427
5.164.2.226 SCB_CPUID_ARCHITECTURE_Msk	427
5.164.2.227 SCB_CPUID_ARCHITECTURE_Pos	427
5.164.2.228 SCB_CPUID_IMPLEMENTER_Msk	427
5.164.2.229 SCB_CPUID_IMPLEMENTER_Pos	428
5.164.2.230 SCB_CPUID_PARTNO_Msk	428
5.164.2.231 SCB_CPUID_PARTNO_Pos	428
5.164.2.232 SCB_CPUID_REVISION_Msk	428
5.164.2.233 SCB_CPUID_REVISION_Pos	428
5.164.2.234 SCB_CPUID_VARIANT_Msk	428
5.164.2.235 SCB_CPUID_VARIANT_Pos	428
5.164.2.236 SCB_DFSR_BKPT_Msk	428
5.164.2.237 SCB_DFSR_BKPT_Pos	429
5.164.2.238 SCB_DFSR_DWTTRAP_Msk	429
5.164.2.239 SCB_DFSR_DWTTRAP_Pos	429
5.164.2.240 SCB_DFSR_EXTERNAL_Msk	429
5.164.2.241 SCB_DFSR_EXTERNAL_Pos	429
5.164.2.242 SCB_DFSR_HALTED_Msk	429
5.164.2.243 SCB_DFSR_HALTED_Pos	429
5.164.2.244 SCB_DFSR_VCATCH_Msk	429
5.164.2.245 SCB_DFSR_VCATCH_Pos	430
5.164.2.246 SCB_HFSR_DEBUGEVT_Msk	430
5.164.2.247 SCB_HFSR_DEBUGEVT_Pos	430
5.164.2.248 SCB_HFSR_FORCED_Msk	430
5.164.2.249 SCB_HFSR_FORCED_Pos	430
5.164.2.250 SCB_HFSR_VECTTBL_Msk	430

5.164.2.251 SCB_HFSR_VECTTBL_Pos	430
5.164.2.252 SCB_ICSR_ISRPENDING_Msk	430
5.164.2.253 SCB_ICSR_ISRPENDING_Pos	431
5.164.2.254 SCB_ICSR_ISRPREEMPT_Msk	431
5.164.2.255 SCB_ICSR_ISRPREEMPT_Pos	431
5.164.2.256 SCB_ICSR_NMIPENDSET_Msk	431
5.164.2.257 SCB_ICSR_NMIPENDSET_Pos	431
5.164.2.258 SCB_ICSR_PENDSTCLR_Msk	431
5.164.2.259 SCB_ICSR_PENDSTCLR_Pos	431
5.164.2.260 SCB_ICSR_PENDSTSET_Msk	431
5.164.2.261 SCB_ICSR_PENDSTSET_Pos	432
5.164.2.262 SCB_ICSR_PENDSVCLR_Msk	432
5.164.2.263 SCB_ICSR_PENDSVCLR_Pos	432
5.164.2.264 SCB_ICSR_PENDSVSET_Msk	432
5.164.2.265 SCB_ICSR_PENDSVSET_Pos	432
5.164.2.266 SCB_ICSR_RETTOBASE_Msk	432
5.164.2.267 SCB_ICSR_RETTOBASE_Pos	432
5.164.2.268 SCB_ICSR_VECTACTIVE_Msk	432
5.164.2.269 SCB_ICSR_VECTACTIVE_Pos	433
5.164.2.270 SCB_ICSR_VECTPENDING_Msk	433
5.164.2.271 SCB_ICSR_VECTPENDING_Pos	433
5.164.2.272 SCB_SCR_SEVONPEND_Msk	433
5.164.2.273 SCB_SCR_SEVONPEND_Pos	433
5.164.2.274 SCB_SCR_SLEEPDEEP_Msk	433
5.164.2.275 SCB_SCR_SLEEPDEEP_Pos	433
5.164.2.276 SCB_SCR_SLEEPONEXIT_Msk	433
5.164.2.277 SCB_SCR_SLEEPONEXIT_Pos	434
5.164.2.278 SCB_SHCSR_BUSFAULTACT_Msk	434
5.164.2.279 SCB_SHCSR_BUSFAULTACT_Pos	434
5.164.2.280 SCB_SHCSR_BUSFAULTENA_Msk	434
5.164.2.281 SCB_SHCSR_BUSFAULTENA_Pos	434
5.164.2.282 SCB_SHCSR_BUSFAULTPENDED_Msk	434
5.164.2.283 SCB_SHCSR_BUSFAULTPENDED_Pos	434
5.164.2.284 SCB_SHCSR_MEMFAULTACT_Msk	434
5.164.2.285 SCB_SHCSR_MEMFAULTACT_Pos	435
5.164.2.286 SCB_SHCSR_MEMFAULTENA_Msk	435
5.164.2.287 SCB_SHCSR_MEMFAULTENA_Pos	435
5.164.2.288 SCB_SHCSR_MEMFAULTPENDED_Msk	435
5.164.2.289 SCB_SHCSR_MEMFAULTPENDED_Pos	435
5.164.2.290 SCB_SHCSR_MONITORACT_Msk	435
5.164.2.291 SCB_SHCSR_MONITORACT_Pos	435
5.164.2.292 SCB_SHCSR_PENDSVACT_Msk	435

5.164.2.293 SCB_SHCSR_PENDSVACT_Pos	436
5.164.2.294 SCB_SHCSR_SVCALLACT_Msk	436
5.164.2.295 SCB_SHCSR_SVCALLACT_Pos	436
5.164.2.296 SCB_SHCSR_SVCALLPENDED_Msk	436
5.164.2.297 SCB_SHCSR_SVCALLPENDED_Pos	436
5.164.2.298 SCB_SHCSR_SYSTICKACT_Msk	436
5.164.2.299 SCB_SHCSR_SYSTICKACT_Pos	436
5.164.2.300 SCB_SHCSR_USGFAULTACT_Msk	436
5.164.2.301 SCB_SHCSR_USGFAULTACT_Pos	437
5.164.2.302 SCB_SHCSR_USGFAULTENA_Msk	437
5.164.2.303 SCB_SHCSR_USGFAULTENA_Pos	437
5.164.2.304 SCB_SHCSR_USGFAULTPENDED_Msk	437
5.164.2.305 SCB_SHCSR_USGFAULTPENDED_Pos	437
5.164.2.306 SCB_VTOR_TBLOFF_Msk	437
5.164.2.307 SCB_VTOR_TBLOFF_Pos	437
5.164.2.308 SCnSCB	437
5.164.2.309 SCnSCB_ACTLR_DISDEFWBUF_Msk	438
5.164.2.310 SCnSCB_ACTLR_DISDEFWBUF_Pos	438
5.164.2.311 SCnSCB_ACTLR_DISFOLD_Msk	438
5.164.2.312 SCnSCB_ACTLR_DISFOLD_Pos	438
5.164.2.313 SCnSCB_ACTLR_DISFPCA_Msk	438
5.164.2.314 SCnSCB_ACTLR_DISFPCA_Pos	438
5.164.2.315 SCnSCB_ACTLR_DISMCYCINT_Msk	438
5.164.2.316 SCnSCB_ACTLR_DISMCYCINT_Pos	438
5.164.2.317 SCnSCB_ACTLR_DISOOPF_Msk	439
5.164.2.318 SCnSCB_ACTLR_DISOOPF_Pos	439
5.164.2.319 SCnSCB_ICTR_INTLINESNUM_Msk	439
5.164.2.320 SCnSCB_ICTR_INTLINESNUM_Pos	439
5.164.2.321 SCS_BASE	439
5.164.2.322 SysTick	439
5.164.2.323 SysTick_BASE	439
5.164.2.324 SysTick_CALIB_NOREF_Msk	440
5.164.2.325 SysTick_CALIB_NOREF_Pos	440
5.164.2.326 SysTick_CALIB_SKEW_Msk	440
5.164.2.327 SysTick_CALIB_SKEW_Pos	440
5.164.2.328 SysTick_CALIB_TENMS_Msk	440
5.164.2.329 SysTick_CALIB_TENMS_Pos	440
5.164.2.330 SysTick_CTRL_CLKSOURCE_Msk	440
5.164.2.331 SysTick_CTRL_CLKSOURCE_Pos	440
5.164.2.332 SysTick_CTRL_COUNTFLAG_Msk	441
5.164.2.333 SysTick_CTRL_COUNTFLAG_Pos	441
5.164.2.334 SysTick_CTRL_ENABLE_Msk	441

5.164.2.335 SysTick_CTRL_ENABLE_Pos	441
5.164.2.336 SysTick_CTRL_TICKINT_Msk	441
5.164.2.337 SysTick_CTRL_TICKINT_Pos	441
5.164.2.338 SysTick_LOAD_RELOAD_Msk	441
5.164.2.339 SysTick_LOAD_RELOAD_Pos	441
5.164.2.340 SysTick_VAL_CURRENT_Msk	442
5.164.2.341 SysTick_VAL_CURRENT_Pos	442
5.164.2.342 TPI	442
5.164.2.343 TPI_ACPR_PRESCALER_Msk	442
5.164.2.344 TPI_ACPR_PRESCALER_Pos	442
5.164.2.345 TPI_BASE	442
5.164.2.346 TPI_DEVID_AsynClkIn_Msk	442
5.164.2.347 TPI_DEVID_AsynClkIn_Pos	443
5.164.2.348 TPI_DEVID_MANCVALID_Msk	443
5.164.2.349 TPI_DEVID_MANCVALID_Pos	443
5.164.2.350 TPI_DEVID_MinBufSz_Msk	443
5.164.2.351 TPI_DEVID_MinBufSz_Pos	443
5.164.2.352 TPI_DEVID_NrTraceInput_Msk	443
5.164.2.353 TPI_DEVID_NrTraceInput_Pos	443
5.164.2.354 TPI_DEVID_NRZVALID_Msk	443
5.164.2.355 TPI_DEVID_NRZVALID_Pos	444
5.164.2.356 TPI_DEVID_PTINVALID_Msk	444
5.164.2.357 TPI_DEVID_PTINVALID_Pos	444
5.164.2.358 TPI_DEVTYPE_MajorType_Msk	444
5.164.2.359 TPI_DEVTYPE_MajorType_Pos	444
5.164.2.360 TPI_DEVTYPE_SubType_Msk	444
5.164.2.361 TPI_DEVTYPE_SubType_Pos	444
5.164.2.362 TPI_FFCR_EnFCont_Msk	444
5.164.2.363 TPI_FFCR_EnFCont_Pos	445
5.164.2.364 TPI_FFCR_TrigIn_Msk	445
5.164.2.365 TPI_FFCR_TrigIn_Pos	445
5.164.2.366 TPI_FFSR_FInProg_Msk	445
5.164.2.367 TPI_FFSR_FInProg_Pos	445
5.164.2.368 TPI_FFSR_FtNonStop_Msk	445
5.164.2.369 TPI_FFSR_FtNonStop_Pos	445
5.164.2.370 TPI_FFSR_FtStopped_Msk	445
5.164.2.371 TPI_FFSR_FtStopped_Pos	446
5.164.2.372 TPI_FFSR_TCPresent_Msk	446
5.164.2.373 TPI_FFSR_TCPresent_Pos	446
5.164.2.374 TPI_FIFO0_ETM0_Msk	446
5.164.2.375 TPI_FIFO0_ETM0_Pos	446
5.164.2.376 TPI_FIFO0_ETM1_Msk	446

5.164.2.377 TPI_FIFO0_ETM1_Pos	446
5.164.2.378 TPI_FIFO0_ETM2_Msk	446
5.164.2.379 TPI_FIFO0_ETM2_Pos	447
5.164.2.380 TPI_FIFO0_ETM_ATVALID_Msk	447
5.164.2.381 TPI_FIFO0_ETM_ATVALID_Pos	447
5.164.2.382 TPI_FIFO0_ETM_bytecount_Msk	447
5.164.2.383 TPI_FIFO0_ETM_bytecount_Pos	447
5.164.2.384 TPI_FIFO0_ITM_ATVALID_Msk	447
5.164.2.385 TPI_FIFO0_ITM_ATVALID_Pos	447
5.164.2.386 TPI_FIFO0_ITM_bytecount_Msk	447
5.164.2.387 TPI_FIFO0_ITM_bytecount_Pos	448
5.164.2.388 TPI_FIFO1_ETM_ATVALID_Msk	448
5.164.2.389 TPI_FIFO1_ETM_ATVALID_Pos	448
5.164.2.390 TPI_FIFO1_ETM_bytecount_Msk	448
5.164.2.391 TPI_FIFO1_ETM_bytecount_Pos	448
5.164.2.392 TPI_FIFO1_ITM0_Msk	448
5.164.2.393 TPI_FIFO1_ITM0_Pos	448
5.164.2.394 TPI_FIFO1_ITM1_Msk	448
5.164.2.395 TPI_FIFO1_ITM1_Pos	449
5.164.2.396 TPI_FIFO1_ITM2_Msk	449
5.164.2.397 TPI_FIFO1_ITM2_Pos	449
5.164.2.398 TPI_FIFO1_ITM_ATVALID_Msk	449
5.164.2.399 TPI_FIFO1_ITM_ATVALID_Pos	449
5.164.2.400 TPI_FIFO1_ITM_bytecount_Msk	449
5.164.2.401 TPI_FIFO1_ITM_bytecount_Pos	449
5.164.2.402 TPI_ITATBCTR0_ATREADY_Msk	449
5.164.2.403 TPI_ITATBCTR0_ATREADY_Pos	450
5.164.2.404 TPI_ITATBCTR2_ATREADY_Msk	450
5.164.2.405 TPI_ITATBCTR2_ATREADY_Pos	450
5.164.2.406 TPI_ITCTRL_Mode_Msk	450
5.164.2.407 TPI_ITCTRL_Mode_Pos	450
5.164.2.408 TPI_SPPR_TXMODE_Msk	450
5.164.2.409 TPI_SPPR_TXMODE_Pos	450
5.164.2.410 TPI_TRIGGER_TRIGGER_Msk	450
5.164.2.411 TPI_TRIGGER_TRIGGER_Pos	451
5.164.2.412 xPSR_C_Msk	451
5.164.2.413 xPSR_C_Pos	451
5.164.2.414 xPSR_GE_Msk	451
5.164.2.415 xPSR_GE_Pos	451
5.164.2.416 xPSR_ISR_Msk	451
5.164.2.417 xPSR_ISR_Pos	451
5.164.2.418 xPSR_IT_Msk	451

5.164.2.419 xPSR_IT_Pos	452
5.164.2.420 xPSR_N_Msk	452
5.164.2.421 xPSR_N_Pos	452
5.164.2.422 xPSR_Q_Msk	452
5.164.2.423 xPSR_Q_Pos	452
5.164.2.424 xPSR_T_Msk	452
5.164.2.425 xPSR_T_Pos	452
5.164.2.426 xPSR_V_Msk	452
5.164.2.427 xPSR_V_Pos	453
5.164.2.428 xPSR_Z_Msk	453
5.164.2.429 xPSR_Z_Pos	453
5.164.3 Function Documentation	453
5.164.3.1 ITM_CheckChar()	453
5.164.3.2 ITM_ReceiveChar()	453
5.164.3.3 ITM_SendChar()	454
5.164.3.4 NVIC_ClearPendingIRQ()	454
5.164.3.5 NVIC_DecodePriority()	454
5.164.3.6 NVIC_DisableIRQ()	455
5.164.3.7 NVIC_EnableIRQ()	455
5.164.3.8 NVIC_EncodePriority()	455
5.164.3.9 NVIC_GetActive()	457
5.164.3.10 NVIC_GetPendingIRQ()	457
5.164.3.11 NVIC_GetPriority()	458
5.164.3.12 NVIC_GetPriorityGrouping()	458
5.164.3.13 NVIC_SetPendingIRQ()	458
5.164.3.14 NVIC_SetPriority()	459
5.164.3.15 NVIC_SetPriorityGrouping()	459
5.164.3.16 NVIC_SystemReset()	459
5.164.3.17 SysTick_Config()	460
5.164.3.18 SysTick_Handler()	460
5.164.4 Variable Documentation	461
5.164.4.1 ACR	461
5.164.4.2 AFR	461
5.164.4.3 AHB1ENR	461
5.164.4.4 AHB1LPENR	461
5.164.4.5 AHB1RSTR	461
5.164.4.6 AHB2ENR	461
5.164.4.7 AHB2LPENR	462
5.164.4.8 AHB2RSTR	462
5.164.4.9 AHB3ENR	462
5.164.4.10 AHB3LPENR	462
5.164.4.11 AHB3RSTR	462

5.164.4.12 ALRMAR	462
5.164.4.13 ALRMASSR	462
5.164.4.14 ALRMBR	462
5.164.4.15 ALRBSSR	463
5.164.4.16 AMTCR	463
5.164.4.17 APB1ENR	463
5.164.4.18 APB1FZ	463
5.164.4.19 APB1LPENR	463
5.164.4.20 APB1RSTR	463
5.164.4.21 APB2ENR	463
5.164.4.22 APB2FZ	463
5.164.4.23 APB2LPENR	464
5.164.4.24 APB2RSTR	464
5.164.4.25 ARG	464
5.164.4.26 ARR	464
5.164.4.27 AWCR	464
5.164.4.28 BCCR	464
5.164.4.29 BDCR	464
5.164.4.30 BDTR	464
5.164.4.31 BFCR	465
5.164.4.32 BGCLUT	465
5.164.4.33 BGCMAR	465
5.164.4.34 BGCOLR	465
5.164.4.35 BGMAR	465
5.164.4.36 BGOR	465
5.164.4.37 BGPFCCR	465
5.164.4.38 BKP0R	465
5.164.4.39 BKP10R	466
5.164.4.40 BKP11R	466
5.164.4.41 BKP12R	466
5.164.4.42 BKP13R	466
5.164.4.43 BKP14R	466
5.164.4.44 BKP15R	466
5.164.4.45 BKP16R	466
5.164.4.46 BKP17R	466
5.164.4.47 BKP18R	467
5.164.4.48 BKP19R	467
5.164.4.49 BKP1R	467
5.164.4.50 BKP2R	467
5.164.4.51 BKP3R	467
5.164.4.52 BKP4R	467
5.164.4.53 BKP5R	467

5.164.4.54 BKP6R	467
5.164.4.55 BKP7R	468
5.164.4.56 BKP8R	468
5.164.4.57 BKP9R	468
5.164.4.58 BPCR	468
5.164.4.59 BRR	468
5.164.4.60 BSRRH	468
5.164.4.61 BSRRL	468
5.164.4.62 BTR	469
5.164.4.63 CACR	469
5.164.4.64 CALIBR	469
5.164.4.65 CALR	469
5.164.4.66 CCER	469
5.164.4.67 CCMR1	469
5.164.4.68 CCMR2	469
5.164.4.69 CCR [1/2]	470
5.164.4.70 CCR [2/2]	470
5.164.4.71 CCR1	470
5.164.4.72 CCR2	470
5.164.4.73 CCR3	470
5.164.4.74 CCR4	470
5.164.4.75 CDR	470
5.164.4.76 CDSR	470
5.164.4.77 CFBAR	471
5.164.4.78 CFBLNR	471
5.164.4.79 CFBLR	471
5.164.4.80 CFGR	471
5.164.4.81 CFR	471
5.164.4.82 CIR	471
5.164.4.83 CKCR	471
5.164.4.84 CKGATENR	471
5.164.4.85 CLKCR	472
5.164.4.86 CLRFR	472
5.164.4.87 CLUTWR	472
5.164.4.88 CMD	472
5.164.4.89 CMPCR	472
5.164.4.90 CNT	472
5.164.4.91 CPSR	472
5.164.4.92 CR [1/15]	473
5.164.4.93 CR [2/15]	473
5.164.4.94 CR [3/15]	473
5.164.4.95 CR [4/15]	473

5.164.4.96 CR [5/15]	473
5.164.4.97 CR [6/15]	473
5.164.4.98 CR [7/15]	473
5.164.4.99 CR [8/15]	474
5.164.4.100 CR [9/15]	474
5.164.4.101 CR [10/15]	474
5.164.4.102 CR [11/15]	474
5.164.4.103 CR [12/15]	474
5.164.4.104 CR [13/15]	474
5.164.4.105 CR [14/15]	474
5.164.4.106 CR [15/15]	475
5.164.4.107 CR1 [1/6]	475
5.164.4.108 CR1 [2/6]	475
5.164.4.109 CR1 [3/6]	475
5.164.4.110 CR1 [4/6]	475
5.164.4.111 CR1 [5/6]	475
5.164.4.112 CR1 [6/6]	475
5.164.4.113 CR2 [1/6]	475
5.164.4.114 CR2 [2/6]	476
5.164.4.115 CR2 [3/6]	476
5.164.4.116 CR2 [4/6]	476
5.164.4.117 CR2 [5/6]	476
5.164.4.118 CR2 [6/6]	476
5.164.4.119 CR3	476
5.164.4.120 CRCPR	476
5.164.4.121 CSGCM0R	476
5.164.4.122 CSGCM1R	477
5.164.4.123 CSGCM2R	477
5.164.4.124 CSGCM3R	477
5.164.4.125 CSGCM4R	477
5.164.4.126 CSGCM5R	477
5.164.4.127 CSGCM6R	477
5.164.4.128 CSGCM7R	477
5.164.4.129 CSGCMCCM0R	477
5.164.4.130 CSGCMCCM1R	478
5.164.4.131 CSGCMCCM2R	478
5.164.4.132 CSGCMCCM3R	478
5.164.4.133 CSGCMCCM4R	478
5.164.4.134 CSGCMCCM5R	478
5.164.4.135 CSGCMCCM6R	478
5.164.4.136 CSGCMCCM7R	478
5.164.4.137 CSR [1/4]	478

5.164.4.138 CSR [2/4]	479
5.164.4.139 CSR [3/4]	479
5.164.4.140 CSR [4/4]	479
5.164.4.141 CWSIZER	479
5.164.4.142 CWSTRTR	479
5.164.4.143 DCCR	479
5.164.4.144 DCKCFGR	479
5.164.4.145 DCKCFGR2	479
5.164.4.146 DCOUNT	480
5.164.4.147 DCR	480
5.164.4.148 DCTRL	480
5.164.4.149 DHR12L1	480
5.164.4.150 DHR12L2	480
5.164.4.151 DHR12LD	480
5.164.4.152 DHR12R1	480
5.164.4.153 DHR12R2	480
5.164.4.154 DHR12RD	481
5.164.4.155 DHR8R1	481
5.164.4.156 DHR8R2	481
5.164.4.157 DHR8RD	481
5.164.4.158 DIER	481
5.164.4.159 DIN	481
5.164.4.160 DLEN	481
5.164.4.161 DMACR	482
5.164.4.162 DMAR	482
5.164.4.163 DOR1	482
5.164.4.164 DOR2	482
5.164.4.165 DOUT	482
5.164.4.166 DR [1/10]	482
5.164.4.167 DR [2/10]	482
5.164.4.168 DR [3/10]	482
5.164.4.169 DR [4/10]	483
5.164.4.170 DR [5/10]	483
5.164.4.171 DR [6/10]	483
5.164.4.172 DR [7/10]	483
5.164.4.173 DR [8/10]	483
5.164.4.174 DR [9/10]	483
5.164.4.175 DR [10/10]	483
5.164.4.176 DTIMER	483
5.164.4.177 EGR	484
5.164.4.178 EMR	484
5.164.4.179 ESCR	484

5.164.4.180 ESR	484
5.164.4.181 ESUR	484
5.164.4.182 EXTICR	484
5.164.4.183 FA1R	484
5.164.4.184 FCR	485
5.164.4.185 FFA1R	485
5.164.4.186 FGCLUT	485
5.164.4.187 FGCMAR	485
5.164.4.188 FGCOLR	485
5.164.4.189 FGMAR	485
5.164.4.190 FGOR	485
5.164.4.191 FGPFCCR	486
5.164.4.192 FIFO	486
5.164.4.193 FIFOCNT	486
5.164.4.194 FLTR	486
5.164.4.195 FM1R	486
5.164.4.196 FMR	486
5.164.4.197 FR1	486
5.164.4.198 FR2	487
5.164.4.199 FRCR	487
5.164.4.200 FS1R	487
5.164.4.201 FTSR	487
5.164.4.202 GCR [1/2]	487
5.164.4.203 GCR [2/2]	487
5.164.4.204 GTPR	487
5.164.4.205 HIFCR	488
5.164.4.206 HISR	488
5.164.4.207 HR [1/2]	488
5.164.4.208 HR [2/2]	488
5.164.4.209 HTR	488
5.164.4.210 I2SCFGR	488
5.164.4.211 I2SPR	488
5.164.4.212 ICR [1/3]	489
5.164.4.213 ICR [2/3]	489
5.164.4.214 ICR [3/3]	489
5.164.4.215 IDCODE	489
5.164.4.216 IDR [1/2]	489
5.164.4.217 IDR [2/2]	489
5.164.4.218 IER [1/3]	489
5.164.4.219 IER [2/3]	490
5.164.4.220 IER [3/3]	490
5.164.4.221 IFCR	490

5.164.4.222 IMR [1/3]	490
5.164.4.223 IMR [2/3]	490
5.164.4.224 IMR [3/3]	490
5.164.4.225 IMSCR	490
5.164.4.226 ISR [1/3]	491
5.164.4.227 ISR [2/3]	491
5.164.4.228 ISR [3/3]	491
5.164.4.229 ITM_RxBuffer	491
5.164.4.230 IV0LR	491
5.164.4.231 IV0RR	491
5.164.4.232 IV1LR	491
5.164.4.233 IV1RR	492
5.164.4.234 JDR1	492
5.164.4.235 JDR2	492
5.164.4.236 JDR3	492
5.164.4.237 JDR4	492
5.164.4.238 JOFR1	492
5.164.4.239 JOFR2	492
5.164.4.240 JOFR3	492
5.164.4.241 JOFR4	493
5.164.4.242 JSQR	493
5.164.4.243 K0LR	493
5.164.4.244 K0RR	493
5.164.4.245 K1LR	493
5.164.4.246 K1RR	493
5.164.4.247 K2LR	493
5.164.4.248 K2RR	493
5.164.4.249 K3LR	494
5.164.4.250 K3RR	494
5.164.4.251 KEYR	494
5.164.4.252 KR	494
5.164.4.253 LCKR	494
5.164.4.254 LIFCR	494
5.164.4.255 LIPCR	494
5.164.4.256 LISR	495
5.164.4.257 LTR	495
5.164.4.258 LWR	495
5.164.4.259 M0AR	495
5.164.4.260 M1AR	495
5.164.4.261 MASK	495
5.164.4.262 MCR	495
5.164.4.263 MEMRMP	496

5.164.4.264 MISR [1/2]	496
5.164.4.265 MISR [2/2]	496
5.164.4.266 MODER	496
5.164.4.267 MSR	496
5.164.4.268 NDTR	496
5.164.4.269 NLR	496
5.164.4.270 OAR1	497
5.164.4.271 OAR2	497
5.164.4.272 OCOLR	497
5.164.4.273 ODR	497
5.164.4.274 OMAR	497
5.164.4.275 OOR	497
5.164.4.276 OPFCCR	497
5.164.4.277 OPTCR	498
5.164.4.278 OPTCR1	498
5.164.4.279 OPTKEYR	498
5.164.4.280 OR	498
5.164.4.281 OSPEEDR	498
5.164.4.282 OTYPER	498
5.164.4.283 PAR	498
5.164.4.284 PFCR	499
5.164.4.285 PLLCFGR	499
5.164.4.286 PLLI2SCFGR	499
5.164.4.287 PLLSAICFGR	499
5.164.4.288 PMC	499
5.164.4.289 POWER	499
5.164.4.290 PR [1/2]	499
5.164.4.291 PR [2/2]	500
5.164.4.292 PRER	500
5.164.4.293 PSC	500
5.164.4.294 PUPDR	500
5.164.4.295 RCR	500
5.164.4.296 RDHR	500
5.164.4.297 RDLR	500
5.164.4.298 RDTR	501
5.164.4.299 RESERVED [1/3]	501
5.164.4.300 RESERVED [2/3]	501
5.164.4.301 RESERVED [3/3]	501
5.164.4.302 RESERVED0 [1/10]	501
5.164.4.303 RESERVED0 [2/10]	501
5.164.4.304 RESERVED0 [3/10]	501
5.164.4.305 RESERVED0 [4/10]	502

5.164.4.306 RESERVED0 [5/10]	502
5.164.4.307 RESERVED0 [6/10]	502
5.164.4.308 RESERVED0 [7/10]	502
5.164.4.309 RESERVED0 [8/10]	502
5.164.4.310 RESERVED0 [9/10]	502
5.164.4.311 RESERVED0 [10/10]	502
5.164.4.312 RESERVED1 [1/10]	503
5.164.4.313 RESERVED1 [2/10]	503
5.164.4.314 RESERVED1 [3/10]	503
5.164.4.315 RESERVED1 [4/10]	503
5.164.4.316 RESERVED1 [5/10]	503
5.164.4.317 RESERVED1 [6/10]	503
5.164.4.318 RESERVED1 [7/10]	503
5.164.4.319 RESERVED1 [8/10]	504
5.164.4.320 RESERVED1 [9/10]	504
5.164.4.321 RESERVED1 [10/10]	504
5.164.4.322 RESERVED10	504
5.164.4.323 RESERVED11	504
5.164.4.324 RESERVED12	504
5.164.4.325 RESERVED13	504
5.164.4.326 RESERVED14	505
5.164.4.327 RESERVED2 [1/7]	505
5.164.4.328 RESERVED2 [2/7]	505
5.164.4.329 RESERVED2 [3/7]	505
5.164.4.330 RESERVED2 [4/7]	505
5.164.4.331 RESERVED2 [5/7]	505
5.164.4.332 RESERVED2 [6/7]	505
5.164.4.333 RESERVED2 [7/7]	506
5.164.4.334 RESERVED3 [1/7]	506
5.164.4.335 RESERVED3 [2/7]	506
5.164.4.336 RESERVED3 [3/7]	506
5.164.4.337 RESERVED3 [4/7]	506
5.164.4.338 RESERVED3 [5/7]	506
5.164.4.339 RESERVED3 [6/7]	506
5.164.4.340 RESERVED3 [7/7]	507
5.164.4.341 RESERVED4 [1/6]	507
5.164.4.342 RESERVED4 [2/6]	507
5.164.4.343 RESERVED4 [3/6]	507
5.164.4.344 RESERVED4 [4/6]	507
5.164.4.345 RESERVED4 [5/6]	507
5.164.4.346 RESERVED4 [6/6]	507
5.164.4.347 RESERVED5 [1/6]	508

5.164.4.348 RESERVED5 [2/6]	508
5.164.4.349 RESERVED5 [3/6]	508
5.164.4.350 RESERVED5 [4/6]	508
5.164.4.351 RESERVED5 [5/6]	508
5.164.4.352 RESERVED5 [6/6]	508
5.164.4.353 RESERVED6 [1/5]	508
5.164.4.354 RESERVED6 [2/5]	509
5.164.4.355 RESERVED6 [3/5]	509
5.164.4.356 RESERVED6 [4/5]	509
5.164.4.357 RESERVED6 [5/5]	509
5.164.4.358 RESERVED7 [1/4]	509
5.164.4.359 RESERVED7 [2/4]	509
5.164.4.360 RESERVED7 [3/4]	509
5.164.4.361 RESERVED7 [4/4]	510
5.164.4.362 RESERVED8 [1/3]	510
5.164.4.363 RESERVED8 [2/3]	510
5.164.4.364 RESERVED8 [3/3]	510
5.164.4.365 RESERVED9 [1/2]	510
5.164.4.366 RESERVED9 [2/2]	510
5.164.4.367 RESP1	510
5.164.4.368 RESP2	511
5.164.4.369 RESP3	511
5.164.4.370 RESP4	511
5.164.4.371 RESPCMD	511
5.164.4.372 RF0R	511
5.164.4.373 RF1R	511
5.164.4.374 RIR	511
5.164.4.375 RISR [1/2]	512
5.164.4.376 RISR [2/2]	512
5.164.4.377 RLR	512
5.164.4.378 RTSR	512
5.164.4.379 RXCRCR	512
5.164.4.380 sFIFOMailBox	512
5.164.4.381 sFilterRegister	512
5.164.4.382 SHIFTR	513
5.164.4.383 SLOTR	513
5.164.4.384 SMCR	513
5.164.4.385 SMPR1	513
5.164.4.386 SMPR2	513
5.164.4.387 SQR1	513
5.164.4.388 SQR2	513
5.164.4.389 SQR3	513

5.164.4.390 SR [1/13]	514
5.164.4.391 SR [2/13]	514
5.164.4.392 SR [3/13]	514
5.164.4.393 SR [4/13]	514
5.164.4.394 SR [5/13]	514
5.164.4.395 SR [6/13]	514
5.164.4.396 SR [7/13]	514
5.164.4.397 SR [8/13]	514
5.164.4.398 SR [9/13]	515
5.164.4.399 SR [10/13]	515
5.164.4.400 SR [11/13]	515
5.164.4.401 SR [12/13]	515
5.164.4.402 SR [13/13]	515
5.164.4.403 SR1	515
5.164.4.404 SR2	515
5.164.4.405 SRCR	516
5.164.4.406 SSCGR	516
5.164.4.407 SSCR	516
5.164.4.408 SSR	516
5.164.4.409 STA	516
5.164.4.410 STR	516
5.164.4.411 sTxMailBox	516
5.164.4.412 SWIER	517
5.164.4.413 SWTRIGR	517
5.164.4.414 TAFCR	517
5.164.4.415 TDHR	517
5.164.4.416 TDLR	517
5.164.4.417 TDTR	517
5.164.4.418 TIR	517
5.164.4.419 TR	517
5.164.4.420 TRISE	518
5.164.4.421 TSDR	518
5.164.4.422 TSR	518
5.164.4.423 TSSSR	518
5.164.4.424 TSTR	518
5.164.4.425 TWCR	518
5.164.4.426 TXCRCR	518
5.164.4.427 WHPCR	519
5.164.4.428 WPR	519
5.164.4.429 WUTR	519
5.164.4.430 WVPCR	519
5.165 Stm32f4xx	519

5.165.1 Detailed Description	520
5.166 Library_configuration_section	520
5.166.1 Detailed Description	520
5.166.2 Macro Definition Documentation	520
5.166.2.1 __STM32F4XX_STDPERIPH_VERSION	521
5.166.2.2 __STM32F4XX_STDPERIPH_VERSION_MAIN	521
5.166.2.3 __STM32F4XX_STDPERIPH_VERSION_RC	521
5.166.2.4 __STM32F4XX_STDPERIPH_VERSION_SUB1	521
5.166.2.5 __STM32F4XX_STDPERIPH_VERSION_SUB2	521
5.166.2.6 HSE_STARTUP_TIMEOUT	521
5.166.2.7 HSE_VALUE	522
5.166.2.8 HSI_VALUE	522
5.167 Configuration_section_for_CMSIS	522
5.167.1 Detailed Description	523
5.167.2 Macro Definition Documentation	523
5.167.2.1 __CM4_REV	523
5.167.2.2 __FPU_PRESENT	523
5.167.2.3 __MPU_PRESENT	524
5.167.2.4 __NVIC_PRIO_BITS	524
5.167.2.5 __Vendor_SysTickConfig	524
5.167.3 Enumeration Type Documentation	524
5.167.3.1 IRQn	524
5.168 Exported_types	527
5.168.1 Detailed Description	528
5.168.2 Typedef Documentation	528
5.168.2.1 s32	528
5.168.2.2 sc16	528
5.168.2.3 sc32	528
5.168.2.4 sc8	528
5.168.2.5 uc16	528
5.168.2.6 uc32	528
5.168.2.7 uc8	529
5.168.2.8 vsc16	529
5.168.2.9 vsc32	529
5.168.2.10 vsc8	529
5.168.2.11 vuc16	529
5.168.2.12 vuc32	529
5.168.2.13 vuc8	529
5.168.3 Enumeration Type Documentation	529
5.168.3.1 FlagStatus	529
5.169 Peripheral_registers_structures	530
5.169.1 Detailed Description	532

5.170 Peripheral_memory_map	532
5.170.1 Detailed Description	534
5.170.2 Macro Definition Documentation	534
5.170.2.1 AHB2PERIPH_BASE	534
5.170.2.2 BKPSRAM_BASE	534
5.170.2.3 BKPSRAM_BB_BASE	534
5.170.2.4 CCMDATARAM_BASE	535
5.170.2.5 CCMDATARAM_BB_BASE	535
5.170.2.6 DMA2D_BASE	535
5.170.2.7 FLASH_BASE	535
5.170.2.8 GPIOA_BASE	535
5.170.2.9 PERIPH_BASE	535
5.170.2.10 PERIPH_BB_BASE	535
5.170.2.11 SRAM1_BASE	536
5.170.2.12 SRAM1_BB_BASE	536
5.170.2.13 SRAM_BB_BASE	536
5.170.2.14 UART8_BASE	536
5.171 Peripheral_declaration	536
5.171.1 Detailed Description	539
5.172 Exported_constants	539
5.172.1 Detailed Description	539
5.173 Peripheral_Registers_Bits_Definition	539
5.173.1 Detailed Description	623
5.173.2 Macro Definition Documentation	623
5.173.2.1 ADC_CCR_ADCPRE	623
5.173.2.2 ADC_CCR_ADCPRE_0	623
5.173.2.3 ADC_CCR_ADCPRE_1	624
5.173.2.4 ADC_CCR_DDS	624
5.173.2.5 ADC_CCR_DELAY	624
5.173.2.6 ADC_CCR_DELAY_0	624
5.173.2.7 ADC_CCR_DELAY_1	624
5.173.2.8 ADC_CCR_DELAY_2	624
5.173.2.9 ADC_CCR_DELAY_3	624
5.173.2.10 ADC_CCR_DMA	624
5.173.2.11 ADC_CCR_DMA_0	625
5.173.2.12 ADC_CCR_DMA_1	625
5.173.2.13 ADC_CCR_MULTI	625
5.173.2.14 ADC_CCR_MULTI_0	625
5.173.2.15 ADC_CCR_MULTI_1	625
5.173.2.16 ADC_CCR_MULTI_2	625
5.173.2.17 ADC_CCR_MULTI_3	625
5.173.2.18 ADC_CCR_MULTI_4	625

5.173.2.19 ADC_CCR_TSVREFE	626
5.173.2.20 ADC_CCR_VBATE	626
5.173.2.21 ADC_CDR_DATA1	626
5.173.2.22 ADC_CDR_DATA2	626
5.173.2.23 ADC_CR1_AWDCH	626
5.173.2.24 ADC_CR1_AWDCH_0	626
5.173.2.25 ADC_CR1_AWDCH_1	626
5.173.2.26 ADC_CR1_AWDCH_2	626
5.173.2.27 ADC_CR1_AWDCH_3	627
5.173.2.28 ADC_CR1_AWDCH_4	627
5.173.2.29 ADC_CR1_AWDEN	627
5.173.2.30 ADC_CR1_AWDIE	627
5.173.2.31 ADC_CR1_AWDSGL	627
5.173.2.32 ADC_CR1_DISCEN	627
5.173.2.33 ADC_CR1_DISCNUM	627
5.173.2.34 ADC_CR1_DISCNUM_0	628
5.173.2.35 ADC_CR1_DISCNUM_1	628
5.173.2.36 ADC_CR1_DISCNUM_2	628
5.173.2.37 ADC_CR1_EOCIE	628
5.173.2.38 ADC_CR1_JAUTO	628
5.173.2.39 ADC_CR1_JAWDEN	628
5.173.2.40 ADC_CR1_JDISCEN	628
5.173.2.41 ADC_CR1_JEOCIE	629
5.173.2.42 ADC_CR1_OVRIE	629
5.173.2.43 ADC_CR1_RES	629
5.173.2.44 ADC_CR1_RES_0	629
5.173.2.45 ADC_CR1_RES_1	629
5.173.2.46 ADC_CR1_SCAN	629
5.173.2.47 ADC_CR2_ADON	629
5.173.2.48 ADC_CR2_ALIGN	630
5.173.2.49 ADC_CR2_CONT	630
5.173.2.50 ADC_CR2_DDS	630
5.173.2.51 ADC_CR2_DMA	630
5.173.2.52 ADC_CR2_EOCS	630
5.173.2.53 ADC_CR2_EXTEN	630
5.173.2.54 ADC_CR2_EXTEN_0	630
5.173.2.55 ADC_CR2_EXTEN_1	631
5.173.2.56 ADC_CR2_EXTSEL	631
5.173.2.57 ADC_CR2_EXTSEL_0	631
5.173.2.58 ADC_CR2_EXTSEL_1	631
5.173.2.59 ADC_CR2_EXTSEL_2	631
5.173.2.60 ADC_CR2_EXTSEL_3	631

5.173.2.61 ADC_CR2_JEXTEN	631
5.173.2.62 ADC_CR2_JEXTEN_0	631
5.173.2.63 ADC_CR2_JEXTEN_1	632
5.173.2.64 ADC_CR2_JEXTSEL	632
5.173.2.65 ADC_CR2_JEXTSEL_0	632
5.173.2.66 ADC_CR2_JEXTSEL_1	632
5.173.2.67 ADC_CR2_JEXTSEL_2	632
5.173.2.68 ADC_CR2_JEXTSEL_3	632
5.173.2.69 ADC_CR2_JSWSTART	632
5.173.2.70 ADC_CR2_SWSTART	632
5.173.2.71 ADC_CSR_AWD1	633
5.173.2.72 ADC_CSR_AWD2	633
5.173.2.73 ADC_CSR_AWD3	633
5.173.2.74 ADC_CSR_EOC1	633
5.173.2.75 ADC_CSR_EOC2	633
5.173.2.76 ADC_CSR_EOC3	633
5.173.2.77 ADC_CSR_JEOC1	633
5.173.2.78 ADC_CSR_JEOC2	633
5.173.2.79 ADC_CSR_JEOC3	634
5.173.2.80 ADC_CSR_JSTRT1	634
5.173.2.81 ADC_CSR_JSTRT2	634
5.173.2.82 ADC_CSR_JSTRT3	634
5.173.2.83 ADC_CSR_OVR1	634
5.173.2.84 ADC_CSR_OVR2	634
5.173.2.85 ADC_CSR_OVR3	634
5.173.2.86 ADC_CSR_STRT1	634
5.173.2.87 ADC_CSR_STRT2	635
5.173.2.88 ADC_CSR_STRT3	635
5.173.2.89 ADC_DR_ADC2DATA	635
5.173.2.90 ADC_DR_DATA	635
5.173.2.91 ADC_HTR_HT	635
5.173.2.92 ADC_JDR1_JDATA	635
5.173.2.93 ADC_JDR2_JDATA	635
5.173.2.94 ADC_JDR3_JDATA	635
5.173.2.95 ADC_JDR4_JDATA	636
5.173.2.96 ADC_JOFR1_JOFFSET1	636
5.173.2.97 ADC_JOFR2_JOFFSET2	636
5.173.2.98 ADC_JOFR3_JOFFSET3	636
5.173.2.99 ADC_JOFR4_JOFFSET4	636
5.173.2.100 ADC_JSQR_JL	636
5.173.2.101 ADC_JSQR_JL_0	636
5.173.2.102 ADC_JSQR_JL_1	636

5.173.2.103 ADC_JSQR_JSQ1	637
5.173.2.104 ADC_JSQR_JSQ1_0	637
5.173.2.105 ADC_JSQR_JSQ1_1	637
5.173.2.106 ADC_JSQR_JSQ1_2	637
5.173.2.107 ADC_JSQR_JSQ1_3	637
5.173.2.108 ADC_JSQR_JSQ1_4	637
5.173.2.109 ADC_JSQR_JSQ2	637
5.173.2.110 ADC_JSQR_JSQ2_0	637
5.173.2.111 ADC_JSQR_JSQ2_1	638
5.173.2.112 ADC_JSQR_JSQ2_2	638
5.173.2.113 ADC_JSQR_JSQ2_3	638
5.173.2.114 ADC_JSQR_JSQ2_4	638
5.173.2.115 ADC_JSQR_JSQ3	638
5.173.2.116 ADC_JSQR_JSQ3_0	638
5.173.2.117 ADC_JSQR_JSQ3_1	638
5.173.2.118 ADC_JSQR_JSQ3_2	638
5.173.2.119 ADC_JSQR_JSQ3_3	639
5.173.2.120 ADC_JSQR_JSQ3_4	639
5.173.2.121 ADC_JSQR_JSQ4	639
5.173.2.122 ADC_JSQR_JSQ4_0	639
5.173.2.123 ADC_JSQR_JSQ4_1	639
5.173.2.124 ADC_JSQR_JSQ4_2	639
5.173.2.125 ADC_JSQR_JSQ4_3	639
5.173.2.126 ADC_JSQR_JSQ4_4	639
5.173.2.127 ADC_LTR_LT	640
5.173.2.128 ADC_SMPR1_SMP10	640
5.173.2.129 ADC_SMPR1_SMP10_0	640
5.173.2.130 ADC_SMPR1_SMP10_1	640
5.173.2.131 ADC_SMPR1_SMP10_2	640
5.173.2.132 ADC_SMPR1_SMP11	640
5.173.2.133 ADC_SMPR1_SMP11_0	640
5.173.2.134 ADC_SMPR1_SMP11_1	640
5.173.2.135 ADC_SMPR1_SMP11_2	641
5.173.2.136 ADC_SMPR1_SMP12	641
5.173.2.137 ADC_SMPR1_SMP12_0	641
5.173.2.138 ADC_SMPR1_SMP12_1	641
5.173.2.139 ADC_SMPR1_SMP12_2	641
5.173.2.140 ADC_SMPR1_SMP13	641
5.173.2.141 ADC_SMPR1_SMP13_0	641
5.173.2.142 ADC_SMPR1_SMP13_1	641
5.173.2.143 ADC_SMPR1_SMP13_2	642
5.173.2.144 ADC_SMPR1_SMP14	642

5.173.2.145 ADC_SMPR1_SMP14_0	642
5.173.2.146 ADC_SMPR1_SMP14_1	642
5.173.2.147 ADC_SMPR1_SMP14_2	642
5.173.2.148 ADC_SMPR1_SMP15	642
5.173.2.149 ADC_SMPR1_SMP15_0	642
5.173.2.150 ADC_SMPR1_SMP15_1	642
5.173.2.151 ADC_SMPR1_SMP15_2	643
5.173.2.152 ADC_SMPR1_SMP16	643
5.173.2.153 ADC_SMPR1_SMP16_0	643
5.173.2.154 ADC_SMPR1_SMP16_1	643
5.173.2.155 ADC_SMPR1_SMP16_2	643
5.173.2.156 ADC_SMPR1_SMP17	643
5.173.2.157 ADC_SMPR1_SMP17_0	643
5.173.2.158 ADC_SMPR1_SMP17_1	643
5.173.2.159 ADC_SMPR1_SMP17_2	644
5.173.2.160 ADC_SMPR1_SMP18	644
5.173.2.161 ADC_SMPR1_SMP18_0	644
5.173.2.162 ADC_SMPR1_SMP18_1	644
5.173.2.163 ADC_SMPR1_SMP18_2	644
5.173.2.164 ADC_SMPR2_SMP0	644
5.173.2.165 ADC_SMPR2_SMP0_0	644
5.173.2.166 ADC_SMPR2_SMP0_1	644
5.173.2.167 ADC_SMPR2_SMP0_2	645
5.173.2.168 ADC_SMPR2_SMP1	645
5.173.2.169 ADC_SMPR2_SMP1_0	645
5.173.2.170 ADC_SMPR2_SMP1_1	645
5.173.2.171 ADC_SMPR2_SMP1_2	645
5.173.2.172 ADC_SMPR2_SMP2	645
5.173.2.173 ADC_SMPR2_SMP2_0	645
5.173.2.174 ADC_SMPR2_SMP2_1	645
5.173.2.175 ADC_SMPR2_SMP2_2	646
5.173.2.176 ADC_SMPR2_SMP3	646
5.173.2.177 ADC_SMPR2_SMP3_0	646
5.173.2.178 ADC_SMPR2_SMP3_1	646
5.173.2.179 ADC_SMPR2_SMP3_2	646
5.173.2.180 ADC_SMPR2_SMP4	646
5.173.2.181 ADC_SMPR2_SMP4_0	646
5.173.2.182 ADC_SMPR2_SMP4_1	646
5.173.2.183 ADC_SMPR2_SMP4_2	647
5.173.2.184 ADC_SMPR2_SMP5	647
5.173.2.185 ADC_SMPR2_SMP5_0	647
5.173.2.186 ADC_SMPR2_SMP5_1	647

5.173.2.187 ADC_SMPR2_SMP5_2	647
5.173.2.188 ADC_SMPR2_SMP6	647
5.173.2.189 ADC_SMPR2_SMP6_0	647
5.173.2.190 ADC_SMPR2_SMP6_1	647
5.173.2.191 ADC_SMPR2_SMP6_2	648
5.173.2.192 ADC_SMPR2_SMP7	648
5.173.2.193 ADC_SMPR2_SMP7_0	648
5.173.2.194 ADC_SMPR2_SMP7_1	648
5.173.2.195 ADC_SMPR2_SMP7_2	648
5.173.2.196 ADC_SMPR2_SMP8	648
5.173.2.197 ADC_SMPR2_SMP8_0	648
5.173.2.198 ADC_SMPR2_SMP8_1	648
5.173.2.199 ADC_SMPR2_SMP8_2	649
5.173.2.200 ADC_SMPR2_SMP9	649
5.173.2.201 ADC_SMPR2_SMP9_0	649
5.173.2.202 ADC_SMPR2_SMP9_1	649
5.173.2.203 ADC_SMPR2_SMP9_2	649
5.173.2.204 ADC_SQR1_L	649
5.173.2.205 ADC_SQR1_L_0	649
5.173.2.206 ADC_SQR1_L_1	649
5.173.2.207 ADC_SQR1_L_2	650
5.173.2.208 ADC_SQR1_L_3	650
5.173.2.209 ADC_SQR1_SQ13	650
5.173.2.210 ADC_SQR1_SQ13_0	650
5.173.2.211 ADC_SQR1_SQ13_1	650
5.173.2.212 ADC_SQR1_SQ13_2	650
5.173.2.213 ADC_SQR1_SQ13_3	650
5.173.2.214 ADC_SQR1_SQ13_4	650
5.173.2.215 ADC_SQR1_SQ14	651
5.173.2.216 ADC_SQR1_SQ14_0	651
5.173.2.217 ADC_SQR1_SQ14_1	651
5.173.2.218 ADC_SQR1_SQ14_2	651
5.173.2.219 ADC_SQR1_SQ14_3	651
5.173.2.220 ADC_SQR1_SQ14_4	651
5.173.2.221 ADC_SQR1_SQ15	651
5.173.2.222 ADC_SQR1_SQ15_0	651
5.173.2.223 ADC_SQR1_SQ15_1	652
5.173.2.224 ADC_SQR1_SQ15_2	652
5.173.2.225 ADC_SQR1_SQ15_3	652
5.173.2.226 ADC_SQR1_SQ15_4	652
5.173.2.227 ADC_SQR1_SQ16	652
5.173.2.228 ADC_SQR1_SQ16_0	652

5.173.2.229 ADC_SQR1_SQ16_1	652
5.173.2.230 ADC_SQR1_SQ16_2	652
5.173.2.231 ADC_SQR1_SQ16_3	653
5.173.2.232 ADC_SQR1_SQ16_4	653
5.173.2.233 ADC_SQR2_SQ10	653
5.173.2.234 ADC_SQR2_SQ10_0	653
5.173.2.235 ADC_SQR2_SQ10_1	653
5.173.2.236 ADC_SQR2_SQ10_2	653
5.173.2.237 ADC_SQR2_SQ10_3	653
5.173.2.238 ADC_SQR2_SQ10_4	653
5.173.2.239 ADC_SQR2_SQ11	654
5.173.2.240 ADC_SQR2_SQ11_0	654
5.173.2.241 ADC_SQR2_SQ11_1	654
5.173.2.242 ADC_SQR2_SQ11_2	654
5.173.2.243 ADC_SQR2_SQ11_3	654
5.173.2.244 ADC_SQR2_SQ11_4	654
5.173.2.245 ADC_SQR2_SQ12	654
5.173.2.246 ADC_SQR2_SQ12_0	654
5.173.2.247 ADC_SQR2_SQ12_1	655
5.173.2.248 ADC_SQR2_SQ12_2	655
5.173.2.249 ADC_SQR2_SQ12_3	655
5.173.2.250 ADC_SQR2_SQ12_4	655
5.173.2.251 ADC_SQR2_SQ7	655
5.173.2.252 ADC_SQR2_SQ7_0	655
5.173.2.253 ADC_SQR2_SQ7_1	655
5.173.2.254 ADC_SQR2_SQ7_2	655
5.173.2.255 ADC_SQR2_SQ7_3	656
5.173.2.256 ADC_SQR2_SQ7_4	656
5.173.2.257 ADC_SQR2_SQ8	656
5.173.2.258 ADC_SQR2_SQ8_0	656
5.173.2.259 ADC_SQR2_SQ8_1	656
5.173.2.260 ADC_SQR2_SQ8_2	656
5.173.2.261 ADC_SQR2_SQ8_3	656
5.173.2.262 ADC_SQR2_SQ8_4	656
5.173.2.263 ADC_SQR2_SQ9	657
5.173.2.264 ADC_SQR2_SQ9_0	657
5.173.2.265 ADC_SQR2_SQ9_1	657
5.173.2.266 ADC_SQR2_SQ9_2	657
5.173.2.267 ADC_SQR2_SQ9_3	657
5.173.2.268 ADC_SQR2_SQ9_4	657
5.173.2.269 ADC_SQR3_SQ1	657
5.173.2.270 ADC_SQR3_SQ1_0	657

5.173.2.271 ADC_SQR3_SQ1_1	658
5.173.2.272 ADC_SQR3_SQ1_2	658
5.173.2.273 ADC_SQR3_SQ1_3	658
5.173.2.274 ADC_SQR3_SQ1_4	658
5.173.2.275 ADC_SQR3_SQ2	658
5.173.2.276 ADC_SQR3_SQ2_0	658
5.173.2.277 ADC_SQR3_SQ2_1	658
5.173.2.278 ADC_SQR3_SQ2_2	658
5.173.2.279 ADC_SQR3_SQ2_3	659
5.173.2.280 ADC_SQR3_SQ2_4	659
5.173.2.281 ADC_SQR3_SQ3	659
5.173.2.282 ADC_SQR3_SQ3_0	659
5.173.2.283 ADC_SQR3_SQ3_1	659
5.173.2.284 ADC_SQR3_SQ3_2	659
5.173.2.285 ADC_SQR3_SQ3_3	659
5.173.2.286 ADC_SQR3_SQ3_4	659
5.173.2.287 ADC_SQR3_SQ4	660
5.173.2.288 ADC_SQR3_SQ4_0	660
5.173.2.289 ADC_SQR3_SQ4_1	660
5.173.2.290 ADC_SQR3_SQ4_2	660
5.173.2.291 ADC_SQR3_SQ4_3	660
5.173.2.292 ADC_SQR3_SQ4_4	660
5.173.2.293 ADC_SQR3_SQ5	660
5.173.2.294 ADC_SQR3_SQ5_0	660
5.173.2.295 ADC_SQR3_SQ5_1	661
5.173.2.296 ADC_SQR3_SQ5_2	661
5.173.2.297 ADC_SQR3_SQ5_3	661
5.173.2.298 ADC_SQR3_SQ5_4	661
5.173.2.299 ADC_SQR3_SQ6	661
5.173.2.300 ADC_SQR3_SQ6_0	661
5.173.2.301 ADC_SQR3_SQ6_1	661
5.173.2.302 ADC_SQR3_SQ6_2	661
5.173.2.303 ADC_SQR3_SQ6_3	662
5.173.2.304 ADC_SQR3_SQ6_4	662
5.173.2.305 ADC_SR_AWD	662
5.173.2.306 ADC_SR_EOC	662
5.173.2.307 ADC_SR_JEOC	662
5.173.2.308 ADC_SR_JSTRT	662
5.173.2.309 ADC_SR_OVR	662
5.173.2.310 ADC_SR_STRT	663
5.173.2.311 CAN_BTR_BRP	663
5.173.2.312 CAN_BTR_LBKM	663

5.173.2.313 CAN_BTR_SILM	663
5.173.2.314 CAN_BTR_SJW	663
5.173.2.315 CAN_BTR_TS1	663
5.173.2.316 CAN_BTR_TS2	663
5.173.2.317 CAN_ESR_BOFF	664
5.173.2.318 CAN_ESR_EPVF	664
5.173.2.319 CAN_ESR_EWGF	664
5.173.2.320 CAN_ESR_LEC	664
5.173.2.321 CAN_ESR_LEC_0	664
5.173.2.322 CAN_ESR_LEC_1	664
5.173.2.323 CAN_ESR_LEC_2	664
5.173.2.324 CAN_ESR_REC	664
5.173.2.325 CAN_ESR_TEC	665
5.173.2.326 CAN_F0R1_FB0	665
5.173.2.327 CAN_F0R1_FB1	665
5.173.2.328 CAN_F0R1_FB10	665
5.173.2.329 CAN_F0R1_FB11	665
5.173.2.330 CAN_F0R1_FB12	665
5.173.2.331 CAN_F0R1_FB13	665
5.173.2.332 CAN_F0R1_FB14	665
5.173.2.333 CAN_F0R1_FB15	666
5.173.2.334 CAN_F0R1_FB16	666
5.173.2.335 CAN_F0R1_FB17	666
5.173.2.336 CAN_F0R1_FB18	666
5.173.2.337 CAN_F0R1_FB19	666
5.173.2.338 CAN_F0R1_FB2	666
5.173.2.339 CAN_F0R1_FB20	666
5.173.2.340 CAN_F0R1_FB21	666
5.173.2.341 CAN_F0R1_FB22	667
5.173.2.342 CAN_F0R1_FB23	667
5.173.2.343 CAN_F0R1_FB24	667
5.173.2.344 CAN_F0R1_FB25	667
5.173.2.345 CAN_F0R1_FB26	667
5.173.2.346 CAN_F0R1_FB27	667
5.173.2.347 CAN_F0R1_FB28	667
5.173.2.348 CAN_F0R1_FB29	667
5.173.2.349 CAN_F0R1_FB3	668
5.173.2.350 CAN_F0R1_FB30	668
5.173.2.351 CAN_F0R1_FB31	668
5.173.2.352 CAN_F0R1_FB4	668
5.173.2.353 CAN_F0R1_FB5	668
5.173.2.354 CAN_F0R1_FB6	668

5.173.2.355 CAN_F0R1_FB7	668
5.173.2.356 CAN_F0R1_FB8	668
5.173.2.357 CAN_F0R1_FB9	669
5.173.2.358 CAN_F0R2_FB0	669
5.173.2.359 CAN_F0R2_FB1	669
5.173.2.360 CAN_F0R2_FB10	669
5.173.2.361 CAN_F0R2_FB11	669
5.173.2.362 CAN_F0R2_FB12	669
5.173.2.363 CAN_F0R2_FB13	669
5.173.2.364 CAN_F0R2_FB14	669
5.173.2.365 CAN_F0R2_FB15	670
5.173.2.366 CAN_F0R2_FB16	670
5.173.2.367 CAN_F0R2_FB17	670
5.173.2.368 CAN_F0R2_FB18	670
5.173.2.369 CAN_F0R2_FB19	670
5.173.2.370 CAN_F0R2_FB2	670
5.173.2.371 CAN_F0R2_FB20	670
5.173.2.372 CAN_F0R2_FB21	670
5.173.2.373 CAN_F0R2_FB22	671
5.173.2.374 CAN_F0R2_FB23	671
5.173.2.375 CAN_F0R2_FB24	671
5.173.2.376 CAN_F0R2_FB25	671
5.173.2.377 CAN_F0R2_FB26	671
5.173.2.378 CAN_F0R2_FB27	671
5.173.2.379 CAN_F0R2_FB28	671
5.173.2.380 CAN_F0R2_FB29	671
5.173.2.381 CAN_F0R2_FB3	672
5.173.2.382 CAN_F0R2_FB30	672
5.173.2.383 CAN_F0R2_FB31	672
5.173.2.384 CAN_F0R2_FB4	672
5.173.2.385 CAN_F0R2_FB5	672
5.173.2.386 CAN_F0R2_FB6	672
5.173.2.387 CAN_F0R2_FB7	672
5.173.2.388 CAN_F0R2_FB8	672
5.173.2.389 CAN_F0R2_FB9	673
5.173.2.390 CAN_F10R1_FB0	673
5.173.2.391 CAN_F10R1_FB1	673
5.173.2.392 CAN_F10R1_FB10	673
5.173.2.393 CAN_F10R1_FB11	673
5.173.2.394 CAN_F10R1_FB12	673
5.173.2.395 CAN_F10R1_FB13	673
5.173.2.396 CAN_F10R1_FB14	673

5.173.2.397 CAN_F10R1_FB15	674
5.173.2.398 CAN_F10R1_FB16	674
5.173.2.399 CAN_F10R1_FB17	674
5.173.2.400 CAN_F10R1_FB18	674
5.173.2.401 CAN_F10R1_FB19	674
5.173.2.402 CAN_F10R1_FB2	674
5.173.2.403 CAN_F10R1_FB20	674
5.173.2.404 CAN_F10R1_FB21	674
5.173.2.405 CAN_F10R1_FB22	675
5.173.2.406 CAN_F10R1_FB23	675
5.173.2.407 CAN_F10R1_FB24	675
5.173.2.408 CAN_F10R1_FB25	675
5.173.2.409 CAN_F10R1_FB26	675
5.173.2.410 CAN_F10R1_FB27	675
5.173.2.411 CAN_F10R1_FB28	675
5.173.2.412 CAN_F10R1_FB29	675
5.173.2.413 CAN_F10R1_FB3	676
5.173.2.414 CAN_F10R1_FB30	676
5.173.2.415 CAN_F10R1_FB31	676
5.173.2.416 CAN_F10R1_FB4	676
5.173.2.417 CAN_F10R1_FB5	676
5.173.2.418 CAN_F10R1_FB6	676
5.173.2.419 CAN_F10R1_FB7	676
5.173.2.420 CAN_F10R1_FB8	676
5.173.2.421 CAN_F10R1_FB9	677
5.173.2.422 CAN_F10R2_FB0	677
5.173.2.423 CAN_F10R2_FB1	677
5.173.2.424 CAN_F10R2_FB10	677
5.173.2.425 CAN_F10R2_FB11	677
5.173.2.426 CAN_F10R2_FB12	677
5.173.2.427 CAN_F10R2_FB13	677
5.173.2.428 CAN_F10R2_FB14	677
5.173.2.429 CAN_F10R2_FB15	678
5.173.2.430 CAN_F10R2_FB16	678
5.173.2.431 CAN_F10R2_FB17	678
5.173.2.432 CAN_F10R2_FB18	678
5.173.2.433 CAN_F10R2_FB19	678
5.173.2.434 CAN_F10R2_FB2	678
5.173.2.435 CAN_F10R2_FB20	678
5.173.2.436 CAN_F10R2_FB21	678
5.173.2.437 CAN_F10R2_FB22	679
5.173.2.438 CAN_F10R2_FB23	679

5.173.2.439 CAN_F10R2_FB24	679
5.173.2.440 CAN_F10R2_FB25	679
5.173.2.441 CAN_F10R2_FB26	679
5.173.2.442 CAN_F10R2_FB27	679
5.173.2.443 CAN_F10R2_FB28	679
5.173.2.444 CAN_F10R2_FB29	679
5.173.2.445 CAN_F10R2_FB3	680
5.173.2.446 CAN_F10R2_FB30	680
5.173.2.447 CAN_F10R2_FB31	680
5.173.2.448 CAN_F10R2_FB4	680
5.173.2.449 CAN_F10R2_FB5	680
5.173.2.450 CAN_F10R2_FB6	680
5.173.2.451 CAN_F10R2_FB7	680
5.173.2.452 CAN_F10R2_FB8	680
5.173.2.453 CAN_F10R2_FB9	681
5.173.2.454 CAN_F11R1_FB0	681
5.173.2.455 CAN_F11R1_FB1	681
5.173.2.456 CAN_F11R1_FB10	681
5.173.2.457 CAN_F11R1_FB11	681
5.173.2.458 CAN_F11R1_FB12	681
5.173.2.459 CAN_F11R1_FB13	681
5.173.2.460 CAN_F11R1_FB14	681
5.173.2.461 CAN_F11R1_FB15	682
5.173.2.462 CAN_F11R1_FB16	682
5.173.2.463 CAN_F11R1_FB17	682
5.173.2.464 CAN_F11R1_FB18	682
5.173.2.465 CAN_F11R1_FB19	682
5.173.2.466 CAN_F11R1_FB2	682
5.173.2.467 CAN_F11R1_FB20	682
5.173.2.468 CAN_F11R1_FB21	682
5.173.2.469 CAN_F11R1_FB22	683
5.173.2.470 CAN_F11R1_FB23	683
5.173.2.471 CAN_F11R1_FB24	683
5.173.2.472 CAN_F11R1_FB25	683
5.173.2.473 CAN_F11R1_FB26	683
5.173.2.474 CAN_F11R1_FB27	683
5.173.2.475 CAN_F11R1_FB28	683
5.173.2.476 CAN_F11R1_FB29	683
5.173.2.477 CAN_F11R1_FB3	684
5.173.2.478 CAN_F11R1_FB30	684
5.173.2.479 CAN_F11R1_FB31	684
5.173.2.480 CAN_F11R1_FB4	684

5.173.2.481 CAN_F11R1_FB5	684
5.173.2.482 CAN_F11R1_FB6	684
5.173.2.483 CAN_F11R1_FB7	684
5.173.2.484 CAN_F11R1_FB8	684
5.173.2.485 CAN_F11R1_FB9	685
5.173.2.486 CAN_F11R2_FB0	685
5.173.2.487 CAN_F11R2_FB1	685
5.173.2.488 CAN_F11R2_FB10	685
5.173.2.489 CAN_F11R2_FB11	685
5.173.2.490 CAN_F11R2_FB12	685
5.173.2.491 CAN_F11R2_FB13	685
5.173.2.492 CAN_F11R2_FB14	685
5.173.2.493 CAN_F11R2_FB15	686
5.173.2.494 CAN_F11R2_FB16	686
5.173.2.495 CAN_F11R2_FB17	686
5.173.2.496 CAN_F11R2_FB18	686
5.173.2.497 CAN_F11R2_FB19	686
5.173.2.498 CAN_F11R2_FB2	686
5.173.2.499 CAN_F11R2_FB20	686
5.173.2.500 CAN_F11R2_FB21	686
5.173.2.501 CAN_F11R2_FB22	687
5.173.2.502 CAN_F11R2_FB23	687
5.173.2.503 CAN_F11R2_FB24	687
5.173.2.504 CAN_F11R2_FB25	687
5.173.2.505 CAN_F11R2_FB26	687
5.173.2.506 CAN_F11R2_FB27	687
5.173.2.507 CAN_F11R2_FB28	687
5.173.2.508 CAN_F11R2_FB29	687
5.173.2.509 CAN_F11R2_FB3	688
5.173.2.510 CAN_F11R2_FB30	688
5.173.2.511 CAN_F11R2_FB31	688
5.173.2.512 CAN_F11R2_FB4	688
5.173.2.513 CAN_F11R2_FB5	688
5.173.2.514 CAN_F11R2_FB6	688
5.173.2.515 CAN_F11R2_FB7	688
5.173.2.516 CAN_F11R2_FB8	688
5.173.2.517 CAN_F11R2_FB9	689
5.173.2.518 CAN_F12R1_FB0	689
5.173.2.519 CAN_F12R1_FB1	689
5.173.2.520 CAN_F12R1_FB10	689
5.173.2.521 CAN_F12R1_FB11	689
5.173.2.522 CAN_F12R1_FB12	689

5.173.2.523 CAN_F12R1_FB13	689
5.173.2.524 CAN_F12R1_FB14	689
5.173.2.525 CAN_F12R1_FB15	690
5.173.2.526 CAN_F12R1_FB16	690
5.173.2.527 CAN_F12R1_FB17	690
5.173.2.528 CAN_F12R1_FB18	690
5.173.2.529 CAN_F12R1_FB19	690
5.173.2.530 CAN_F12R1_FB2	690
5.173.2.531 CAN_F12R1_FB20	690
5.173.2.532 CAN_F12R1_FB21	690
5.173.2.533 CAN_F12R1_FB22	691
5.173.2.534 CAN_F12R1_FB23	691
5.173.2.535 CAN_F12R1_FB24	691
5.173.2.536 CAN_F12R1_FB25	691
5.173.2.537 CAN_F12R1_FB26	691
5.173.2.538 CAN_F12R1_FB27	691
5.173.2.539 CAN_F12R1_FB28	691
5.173.2.540 CAN_F12R1_FB29	691
5.173.2.541 CAN_F12R1_FB3	692
5.173.2.542 CAN_F12R1_FB30	692
5.173.2.543 CAN_F12R1_FB31	692
5.173.2.544 CAN_F12R1_FB4	692
5.173.2.545 CAN_F12R1_FB5	692
5.173.2.546 CAN_F12R1_FB6	692
5.173.2.547 CAN_F12R1_FB7	692
5.173.2.548 CAN_F12R1_FB8	692
5.173.2.549 CAN_F12R1_FB9	693
5.173.2.550 CAN_F12R2_FB0	693
5.173.2.551 CAN_F12R2_FB1	693
5.173.2.552 CAN_F12R2_FB10	693
5.173.2.553 CAN_F12R2_FB11	693
5.173.2.554 CAN_F12R2_FB12	693
5.173.2.555 CAN_F12R2_FB13	693
5.173.2.556 CAN_F12R2_FB14	693
5.173.2.557 CAN_F12R2_FB15	694
5.173.2.558 CAN_F12R2_FB16	694
5.173.2.559 CAN_F12R2_FB17	694
5.173.2.560 CAN_F12R2_FB18	694
5.173.2.561 CAN_F12R2_FB19	694
5.173.2.562 CAN_F12R2_FB2	694
5.173.2.563 CAN_F12R2_FB20	694
5.173.2.564 CAN_F12R2_FB21	694

5.173.2.565 CAN_F12R2_FB22	695
5.173.2.566 CAN_F12R2_FB23	695
5.173.2.567 CAN_F12R2_FB24	695
5.173.2.568 CAN_F12R2_FB25	695
5.173.2.569 CAN_F12R2_FB26	695
5.173.2.570 CAN_F12R2_FB27	695
5.173.2.571 CAN_F12R2_FB28	695
5.173.2.572 CAN_F12R2_FB29	695
5.173.2.573 CAN_F12R2_FB3	696
5.173.2.574 CAN_F12R2_FB30	696
5.173.2.575 CAN_F12R2_FB31	696
5.173.2.576 CAN_F12R2_FB4	696
5.173.2.577 CAN_F12R2_FB5	696
5.173.2.578 CAN_F12R2_FB6	696
5.173.2.579 CAN_F12R2_FB7	696
5.173.2.580 CAN_F12R2_FB8	696
5.173.2.581 CAN_F12R2_FB9	697
5.173.2.582 CAN_F13R1_FB0	697
5.173.2.583 CAN_F13R1_FB1	697
5.173.2.584 CAN_F13R1_FB10	697
5.173.2.585 CAN_F13R1_FB11	697
5.173.2.586 CAN_F13R1_FB12	697
5.173.2.587 CAN_F13R1_FB13	697
5.173.2.588 CAN_F13R1_FB14	697
5.173.2.589 CAN_F13R1_FB15	698
5.173.2.590 CAN_F13R1_FB16	698
5.173.2.591 CAN_F13R1_FB17	698
5.173.2.592 CAN_F13R1_FB18	698
5.173.2.593 CAN_F13R1_FB19	698
5.173.2.594 CAN_F13R1_FB2	698
5.173.2.595 CAN_F13R1_FB20	698
5.173.2.596 CAN_F13R1_FB21	698
5.173.2.597 CAN_F13R1_FB22	699
5.173.2.598 CAN_F13R1_FB23	699
5.173.2.599 CAN_F13R1_FB24	699
5.173.2.600 CAN_F13R1_FB25	699
5.173.2.601 CAN_F13R1_FB26	699
5.173.2.602 CAN_F13R1_FB27	699
5.173.2.603 CAN_F13R1_FB28	699
5.173.2.604 CAN_F13R1_FB29	699
5.173.2.605 CAN_F13R1_FB3	700
5.173.2.606 CAN_F13R1_FB30	700

5.173.2.607 CAN_F13R1_FB31	700
5.173.2.608 CAN_F13R1_FB4	700
5.173.2.609 CAN_F13R1_FB5	700
5.173.2.610 CAN_F13R1_FB6	700
5.173.2.611 CAN_F13R1_FB7	700
5.173.2.612 CAN_F13R1_FB8	700
5.173.2.613 CAN_F13R1_FB9	701
5.173.2.614 CAN_F13R2_FB0	701
5.173.2.615 CAN_F13R2_FB1	701
5.173.2.616 CAN_F13R2_FB10	701
5.173.2.617 CAN_F13R2_FB11	701
5.173.2.618 CAN_F13R2_FB12	701
5.173.2.619 CAN_F13R2_FB13	701
5.173.2.620 CAN_F13R2_FB14	701
5.173.2.621 CAN_F13R2_FB15	702
5.173.2.622 CAN_F13R2_FB16	702
5.173.2.623 CAN_F13R2_FB17	702
5.173.2.624 CAN_F13R2_FB18	702
5.173.2.625 CAN_F13R2_FB19	702
5.173.2.626 CAN_F13R2_FB2	702
5.173.2.627 CAN_F13R2_FB20	702
5.173.2.628 CAN_F13R2_FB21	702
5.173.2.629 CAN_F13R2_FB22	703
5.173.2.630 CAN_F13R2_FB23	703
5.173.2.631 CAN_F13R2_FB24	703
5.173.2.632 CAN_F13R2_FB25	703
5.173.2.633 CAN_F13R2_FB26	703
5.173.2.634 CAN_F13R2_FB27	703
5.173.2.635 CAN_F13R2_FB28	703
5.173.2.636 CAN_F13R2_FB29	703
5.173.2.637 CAN_F13R2_FB3	704
5.173.2.638 CAN_F13R2_FB30	704
5.173.2.639 CAN_F13R2_FB31	704
5.173.2.640 CAN_F13R2_FB4	704
5.173.2.641 CAN_F13R2_FB5	704
5.173.2.642 CAN_F13R2_FB6	704
5.173.2.643 CAN_F13R2_FB7	704
5.173.2.644 CAN_F13R2_FB8	704
5.173.2.645 CAN_F13R2_FB9	705
5.173.2.646 CAN_F1R1_FB0	705
5.173.2.647 CAN_F1R1_FB1	705
5.173.2.648 CAN_F1R1_FB10	705

5.173.2.649 CAN_F1R1_FB11	705
5.173.2.650 CAN_F1R1_FB12	705
5.173.2.651 CAN_F1R1_FB13	705
5.173.2.652 CAN_F1R1_FB14	705
5.173.2.653 CAN_F1R1_FB15	706
5.173.2.654 CAN_F1R1_FB16	706
5.173.2.655 CAN_F1R1_FB17	706
5.173.2.656 CAN_F1R1_FB18	706
5.173.2.657 CAN_F1R1_FB19	706
5.173.2.658 CAN_F1R1_FB2	706
5.173.2.659 CAN_F1R1_FB20	706
5.173.2.660 CAN_F1R1_FB21	706
5.173.2.661 CAN_F1R1_FB22	707
5.173.2.662 CAN_F1R1_FB23	707
5.173.2.663 CAN_F1R1_FB24	707
5.173.2.664 CAN_F1R1_FB25	707
5.173.2.665 CAN_F1R1_FB26	707
5.173.2.666 CAN_F1R1_FB27	707
5.173.2.667 CAN_F1R1_FB28	707
5.173.2.668 CAN_F1R1_FB29	707
5.173.2.669 CAN_F1R1_FB3	708
5.173.2.670 CAN_F1R1_FB30	708
5.173.2.671 CAN_F1R1_FB31	708
5.173.2.672 CAN_F1R1_FB4	708
5.173.2.673 CAN_F1R1_FB5	708
5.173.2.674 CAN_F1R1_FB6	708
5.173.2.675 CAN_F1R1_FB7	708
5.173.2.676 CAN_F1R1_FB8	708
5.173.2.677 CAN_F1R1_FB9	709
5.173.2.678 CAN_F1R2_FB0	709
5.173.2.679 CAN_F1R2_FB1	709
5.173.2.680 CAN_F1R2_FB10	709
5.173.2.681 CAN_F1R2_FB11	709
5.173.2.682 CAN_F1R2_FB12	709
5.173.2.683 CAN_F1R2_FB13	709
5.173.2.684 CAN_F1R2_FB14	709
5.173.2.685 CAN_F1R2_FB15	710
5.173.2.686 CAN_F1R2_FB16	710
5.173.2.687 CAN_F1R2_FB17	710
5.173.2.688 CAN_F1R2_FB18	710
5.173.2.689 CAN_F1R2_FB19	710
5.173.2.690 CAN_F1R2_FB2	710

5.173.2.691 CAN_F1R2_FB20	710
5.173.2.692 CAN_F1R2_FB21	710
5.173.2.693 CAN_F1R2_FB22	711
5.173.2.694 CAN_F1R2_FB23	711
5.173.2.695 CAN_F1R2_FB24	711
5.173.2.696 CAN_F1R2_FB25	711
5.173.2.697 CAN_F1R2_FB26	711
5.173.2.698 CAN_F1R2_FB27	711
5.173.2.699 CAN_F1R2_FB28	711
5.173.2.700 CAN_F1R2_FB29	711
5.173.2.701 CAN_F1R2_FB3	712
5.173.2.702 CAN_F1R2_FB30	712
5.173.2.703 CAN_F1R2_FB31	712
5.173.2.704 CAN_F1R2_FB4	712
5.173.2.705 CAN_F1R2_FB5	712
5.173.2.706 CAN_F1R2_FB6	712
5.173.2.707 CAN_F1R2_FB7	712
5.173.2.708 CAN_F1R2_FB8	712
5.173.2.709 CAN_F1R2_FB9	713
5.173.2.710 CAN_F2R1_FB0	713
5.173.2.711 CAN_F2R1_FB1	713
5.173.2.712 CAN_F2R1_FB10	713
5.173.2.713 CAN_F2R1_FB11	713
5.173.2.714 CAN_F2R1_FB12	713
5.173.2.715 CAN_F2R1_FB13	713
5.173.2.716 CAN_F2R1_FB14	713
5.173.2.717 CAN_F2R1_FB15	714
5.173.2.718 CAN_F2R1_FB16	714
5.173.2.719 CAN_F2R1_FB17	714
5.173.2.720 CAN_F2R1_FB18	714
5.173.2.721 CAN_F2R1_FB19	714
5.173.2.722 CAN_F2R1_FB2	714
5.173.2.723 CAN_F2R1_FB20	714
5.173.2.724 CAN_F2R1_FB21	714
5.173.2.725 CAN_F2R1_FB22	715
5.173.2.726 CAN_F2R1_FB23	715
5.173.2.727 CAN_F2R1_FB24	715
5.173.2.728 CAN_F2R1_FB25	715
5.173.2.729 CAN_F2R1_FB26	715
5.173.2.730 CAN_F2R1_FB27	715
5.173.2.731 CAN_F2R1_FB28	715
5.173.2.732 CAN_F2R1_FB29	715

5.173.2.733 CAN_F2R1_FB3	716
5.173.2.734 CAN_F2R1_FB30	716
5.173.2.735 CAN_F2R1_FB31	716
5.173.2.736 CAN_F2R1_FB4	716
5.173.2.737 CAN_F2R1_FB5	716
5.173.2.738 CAN_F2R1_FB6	716
5.173.2.739 CAN_F2R1_FB7	716
5.173.2.740 CAN_F2R1_FB8	716
5.173.2.741 CAN_F2R1_FB9	717
5.173.2.742 CAN_F2R2_FB0	717
5.173.2.743 CAN_F2R2_FB1	717
5.173.2.744 CAN_F2R2_FB10	717
5.173.2.745 CAN_F2R2_FB11	717
5.173.2.746 CAN_F2R2_FB12	717
5.173.2.747 CAN_F2R2_FB13	717
5.173.2.748 CAN_F2R2_FB14	717
5.173.2.749 CAN_F2R2_FB15	718
5.173.2.750 CAN_F2R2_FB16	718
5.173.2.751 CAN_F2R2_FB17	718
5.173.2.752 CAN_F2R2_FB18	718
5.173.2.753 CAN_F2R2_FB19	718
5.173.2.754 CAN_F2R2_FB2	718
5.173.2.755 CAN_F2R2_FB20	718
5.173.2.756 CAN_F2R2_FB21	718
5.173.2.757 CAN_F2R2_FB22	719
5.173.2.758 CAN_F2R2_FB23	719
5.173.2.759 CAN_F2R2_FB24	719
5.173.2.760 CAN_F2R2_FB25	719
5.173.2.761 CAN_F2R2_FB26	719
5.173.2.762 CAN_F2R2_FB27	719
5.173.2.763 CAN_F2R2_FB28	719
5.173.2.764 CAN_F2R2_FB29	719
5.173.2.765 CAN_F2R2_FB3	720
5.173.2.766 CAN_F2R2_FB30	720
5.173.2.767 CAN_F2R2_FB31	720
5.173.2.768 CAN_F2R2_FB4	720
5.173.2.769 CAN_F2R2_FB5	720
5.173.2.770 CAN_F2R2_FB6	720
5.173.2.771 CAN_F2R2_FB7	720
5.173.2.772 CAN_F2R2_FB8	720
5.173.2.773 CAN_F2R2_FB9	721
5.173.2.774 CAN_F3R1_FB0	721

5.173.2.775 CAN_F3R1_FB1	721
5.173.2.776 CAN_F3R1_FB10	721
5.173.2.777 CAN_F3R1_FB11	721
5.173.2.778 CAN_F3R1_FB12	721
5.173.2.779 CAN_F3R1_FB13	721
5.173.2.780 CAN_F3R1_FB14	721
5.173.2.781 CAN_F3R1_FB15	722
5.173.2.782 CAN_F3R1_FB16	722
5.173.2.783 CAN_F3R1_FB17	722
5.173.2.784 CAN_F3R1_FB18	722
5.173.2.785 CAN_F3R1_FB19	722
5.173.2.786 CAN_F3R1_FB2	722
5.173.2.787 CAN_F3R1_FB20	722
5.173.2.788 CAN_F3R1_FB21	722
5.173.2.789 CAN_F3R1_FB22	723
5.173.2.790 CAN_F3R1_FB23	723
5.173.2.791 CAN_F3R1_FB24	723
5.173.2.792 CAN_F3R1_FB25	723
5.173.2.793 CAN_F3R1_FB26	723
5.173.2.794 CAN_F3R1_FB27	723
5.173.2.795 CAN_F3R1_FB28	723
5.173.2.796 CAN_F3R1_FB29	723
5.173.2.797 CAN_F3R1_FB3	724
5.173.2.798 CAN_F3R1_FB30	724
5.173.2.799 CAN_F3R1_FB31	724
5.173.2.800 CAN_F3R1_FB4	724
5.173.2.801 CAN_F3R1_FB5	724
5.173.2.802 CAN_F3R1_FB6	724
5.173.2.803 CAN_F3R1_FB7	724
5.173.2.804 CAN_F3R1_FB8	724
5.173.2.805 CAN_F3R1_FB9	725
5.173.2.806 CAN_F3R2_FB0	725
5.173.2.807 CAN_F3R2_FB1	725
5.173.2.808 CAN_F3R2_FB10	725
5.173.2.809 CAN_F3R2_FB11	725
5.173.2.810 CAN_F3R2_FB12	725
5.173.2.811 CAN_F3R2_FB13	725
5.173.2.812 CAN_F3R2_FB14	725
5.173.2.813 CAN_F3R2_FB15	726
5.173.2.814 CAN_F3R2_FB16	726
5.173.2.815 CAN_F3R2_FB17	726
5.173.2.816 CAN_F3R2_FB18	726

5.173.2.817 CAN_F3R2_FB19	726
5.173.2.818 CAN_F3R2_FB2	726
5.173.2.819 CAN_F3R2_FB20	726
5.173.2.820 CAN_F3R2_FB21	726
5.173.2.821 CAN_F3R2_FB22	727
5.173.2.822 CAN_F3R2_FB23	727
5.173.2.823 CAN_F3R2_FB24	727
5.173.2.824 CAN_F3R2_FB25	727
5.173.2.825 CAN_F3R2_FB26	727
5.173.2.826 CAN_F3R2_FB27	727
5.173.2.827 CAN_F3R2_FB28	727
5.173.2.828 CAN_F3R2_FB29	727
5.173.2.829 CAN_F3R2_FB3	728
5.173.2.830 CAN_F3R2_FB30	728
5.173.2.831 CAN_F3R2_FB31	728
5.173.2.832 CAN_F3R2_FB4	728
5.173.2.833 CAN_F3R2_FB5	728
5.173.2.834 CAN_F3R2_FB6	728
5.173.2.835 CAN_F3R2_FB7	728
5.173.2.836 CAN_F3R2_FB8	728
5.173.2.837 CAN_F3R2_FB9	729
5.173.2.838 CAN_F4R1_FB0	729
5.173.2.839 CAN_F4R1_FB1	729
5.173.2.840 CAN_F4R1_FB10	729
5.173.2.841 CAN_F4R1_FB11	729
5.173.2.842 CAN_F4R1_FB12	729
5.173.2.843 CAN_F4R1_FB13	729
5.173.2.844 CAN_F4R1_FB14	729
5.173.2.845 CAN_F4R1_FB15	730
5.173.2.846 CAN_F4R1_FB16	730
5.173.2.847 CAN_F4R1_FB17	730
5.173.2.848 CAN_F4R1_FB18	730
5.173.2.849 CAN_F4R1_FB19	730
5.173.2.850 CAN_F4R1_FB2	730
5.173.2.851 CAN_F4R1_FB20	730
5.173.2.852 CAN_F4R1_FB21	730
5.173.2.853 CAN_F4R1_FB22	731
5.173.2.854 CAN_F4R1_FB23	731
5.173.2.855 CAN_F4R1_FB24	731
5.173.2.856 CAN_F4R1_FB25	731
5.173.2.857 CAN_F4R1_FB26	731
5.173.2.858 CAN_F4R1_FB27	731

5.173.2.859 CAN_F4R1_FB28	731
5.173.2.860 CAN_F4R1_FB29	731
5.173.2.861 CAN_F4R1_FB3	732
5.173.2.862 CAN_F4R1_FB30	732
5.173.2.863 CAN_F4R1_FB31	732
5.173.2.864 CAN_F4R1_FB4	732
5.173.2.865 CAN_F4R1_FB5	732
5.173.2.866 CAN_F4R1_FB6	732
5.173.2.867 CAN_F4R1_FB7	732
5.173.2.868 CAN_F4R1_FB8	732
5.173.2.869 CAN_F4R1_FB9	733
5.173.2.870 CAN_F4R2_FB0	733
5.173.2.871 CAN_F4R2_FB1	733
5.173.2.872 CAN_F4R2_FB10	733
5.173.2.873 CAN_F4R2_FB11	733
5.173.2.874 CAN_F4R2_FB12	733
5.173.2.875 CAN_F4R2_FB13	733
5.173.2.876 CAN_F4R2_FB14	733
5.173.2.877 CAN_F4R2_FB15	734
5.173.2.878 CAN_F4R2_FB16	734
5.173.2.879 CAN_F4R2_FB17	734
5.173.2.880 CAN_F4R2_FB18	734
5.173.2.881 CAN_F4R2_FB19	734
5.173.2.882 CAN_F4R2_FB2	734
5.173.2.883 CAN_F4R2_FB20	734
5.173.2.884 CAN_F4R2_FB21	734
5.173.2.885 CAN_F4R2_FB22	735
5.173.2.886 CAN_F4R2_FB23	735
5.173.2.887 CAN_F4R2_FB24	735
5.173.2.888 CAN_F4R2_FB25	735
5.173.2.889 CAN_F4R2_FB26	735
5.173.2.890 CAN_F4R2_FB27	735
5.173.2.891 CAN_F4R2_FB28	735
5.173.2.892 CAN_F4R2_FB29	735
5.173.2.893 CAN_F4R2_FB3	736
5.173.2.894 CAN_F4R2_FB30	736
5.173.2.895 CAN_F4R2_FB31	736
5.173.2.896 CAN_F4R2_FB4	736
5.173.2.897 CAN_F4R2_FB5	736
5.173.2.898 CAN_F4R2_FB6	736
5.173.2.899 CAN_F4R2_FB7	736
5.173.2.900 CAN_F4R2_FB8	736

5.173.2.901 CAN_F4R2_FB9	737
5.173.2.902 CAN_F5R1_FB0	737
5.173.2.903 CAN_F5R1_FB1	737
5.173.2.904 CAN_F5R1_FB10	737
5.173.2.905 CAN_F5R1_FB11	737
5.173.2.906 CAN_F5R1_FB12	737
5.173.2.907 CAN_F5R1_FB13	737
5.173.2.908 CAN_F5R1_FB14	737
5.173.2.909 CAN_F5R1_FB15	738
5.173.2.910 CAN_F5R1_FB16	738
5.173.2.911 CAN_F5R1_FB17	738
5.173.2.912 CAN_F5R1_FB18	738
5.173.2.913 CAN_F5R1_FB19	738
5.173.2.914 CAN_F5R1_FB2	738
5.173.2.915 CAN_F5R1_FB20	738
5.173.2.916 CAN_F5R1_FB21	738
5.173.2.917 CAN_F5R1_FB22	739
5.173.2.918 CAN_F5R1_FB23	739
5.173.2.919 CAN_F5R1_FB24	739
5.173.2.920 CAN_F5R1_FB25	739
5.173.2.921 CAN_F5R1_FB26	739
5.173.2.922 CAN_F5R1_FB27	739
5.173.2.923 CAN_F5R1_FB28	739
5.173.2.924 CAN_F5R1_FB29	739
5.173.2.925 CAN_F5R1_FB3	740
5.173.2.926 CAN_F5R1_FB30	740
5.173.2.927 CAN_F5R1_FB31	740
5.173.2.928 CAN_F5R1_FB4	740
5.173.2.929 CAN_F5R1_FB5	740
5.173.2.930 CAN_F5R1_FB6	740
5.173.2.931 CAN_F5R1_FB7	740
5.173.2.932 CAN_F5R1_FB8	740
5.173.2.933 CAN_F5R1_FB9	741
5.173.2.934 CAN_F5R2_FB0	741
5.173.2.935 CAN_F5R2_FB1	741
5.173.2.936 CAN_F5R2_FB10	741
5.173.2.937 CAN_F5R2_FB11	741
5.173.2.938 CAN_F5R2_FB12	741
5.173.2.939 CAN_F5R2_FB13	741
5.173.2.940 CAN_F5R2_FB14	741
5.173.2.941 CAN_F5R2_FB15	742
5.173.2.942 CAN_F5R2_FB16	742

5.173.2.943 CAN_F5R2_FB17	742
5.173.2.944 CAN_F5R2_FB18	742
5.173.2.945 CAN_F5R2_FB19	742
5.173.2.946 CAN_F5R2_FB2	742
5.173.2.947 CAN_F5R2_FB20	742
5.173.2.948 CAN_F5R2_FB21	742
5.173.2.949 CAN_F5R2_FB22	743
5.173.2.950 CAN_F5R2_FB23	743
5.173.2.951 CAN_F5R2_FB24	743
5.173.2.952 CAN_F5R2_FB25	743
5.173.2.953 CAN_F5R2_FB26	743
5.173.2.954 CAN_F5R2_FB27	743
5.173.2.955 CAN_F5R2_FB28	743
5.173.2.956 CAN_F5R2_FB29	743
5.173.2.957 CAN_F5R2_FB3	744
5.173.2.958 CAN_F5R2_FB30	744
5.173.2.959 CAN_F5R2_FB31	744
5.173.2.960 CAN_F5R2_FB4	744
5.173.2.961 CAN_F5R2_FB5	744
5.173.2.962 CAN_F5R2_FB6	744
5.173.2.963 CAN_F5R2_FB7	744
5.173.2.964 CAN_F5R2_FB8	744
5.173.2.965 CAN_F5R2_FB9	745
5.173.2.966 CAN_F6R1_FB0	745
5.173.2.967 CAN_F6R1_FB1	745
5.173.2.968 CAN_F6R1_FB10	745
5.173.2.969 CAN_F6R1_FB11	745
5.173.2.970 CAN_F6R1_FB12	745
5.173.2.971 CAN_F6R1_FB13	745
5.173.2.972 CAN_F6R1_FB14	745
5.173.2.973 CAN_F6R1_FB15	746
5.173.2.974 CAN_F6R1_FB16	746
5.173.2.975 CAN_F6R1_FB17	746
5.173.2.976 CAN_F6R1_FB18	746
5.173.2.977 CAN_F6R1_FB19	746
5.173.2.978 CAN_F6R1_FB2	746
5.173.2.979 CAN_F6R1_FB20	746
5.173.2.980 CAN_F6R1_FB21	746
5.173.2.981 CAN_F6R1_FB22	747
5.173.2.982 CAN_F6R1_FB23	747
5.173.2.983 CAN_F6R1_FB24	747
5.173.2.984 CAN_F6R1_FB25	747

5.173.2.985 CAN_F6R1_FB26	747
5.173.2.986 CAN_F6R1_FB27	747
5.173.2.987 CAN_F6R1_FB28	747
5.173.2.988 CAN_F6R1_FB29	747
5.173.2.989 CAN_F6R1_FB3	748
5.173.2.990 CAN_F6R1_FB30	748
5.173.2.991 CAN_F6R1_FB31	748
5.173.2.992 CAN_F6R1_FB4	748
5.173.2.993 CAN_F6R1_FB5	748
5.173.2.994 CAN_F6R1_FB6	748
5.173.2.995 CAN_F6R1_FB7	748
5.173.2.996 CAN_F6R1_FB8	748
5.173.2.997 CAN_F6R1_FB9	749
5.173.2.998 CAN_F6R2_FB0	749
5.173.2.999 CAN_F6R2_FB1	749
5.173.2.1000 CAN_F6R2_FB10	749
5.173.2.1001 CAN_F6R2_FB11	749
5.173.2.1002 CAN_F6R2_FB12	749
5.173.2.1003 CAN_F6R2_FB13	749
5.173.2.1004 CAN_F6R2_FB14	749
5.173.2.1005 CAN_F6R2_FB15	750
5.173.2.1006 CAN_F6R2_FB16	750
5.173.2.1007 CAN_F6R2_FB17	750
5.173.2.1008 CAN_F6R2_FB18	750
5.173.2.1009 CAN_F6R2_FB19	750
5.173.2.1010 CAN_F6R2_FB2	750
5.173.2.1011 CAN_F6R2_FB20	750
5.173.2.1012 CAN_F6R2_FB21	750
5.173.2.1013 CAN_F6R2_FB22	751
5.173.2.1014 CAN_F6R2_FB23	751
5.173.2.1015 CAN_F6R2_FB24	751
5.173.2.1016 CAN_F6R2_FB25	751
5.173.2.1017 CAN_F6R2_FB26	751
5.173.2.1018 CAN_F6R2_FB27	751
5.173.2.1019 CAN_F6R2_FB28	751
5.173.2.1020 CAN_F6R2_FB29	751
5.173.2.1021 CAN_F6R2_FB3	752
5.173.2.1022 CAN_F6R2_FB30	752
5.173.2.1023 CAN_F6R2_FB31	752
5.173.2.1024 CAN_F6R2_FB4	752
5.173.2.1025 CAN_F6R2_FB5	752
5.173.2.1026 CAN_F6R2_FB6	752

5.173.2.1027 CAN_F6R2_FB7	752
5.173.2.1028 CAN_F6R2_FB8	752
5.173.2.1029 CAN_F6R2_FB9	753
5.173.2.1030 CAN_F7R1_FB0	753
5.173.2.1031 CAN_F7R1_FB1	753
5.173.2.1032 CAN_F7R1_FB10	753
5.173.2.1033 CAN_F7R1_FB11	753
5.173.2.1034 CAN_F7R1_FB12	753
5.173.2.1035 CAN_F7R1_FB13	753
5.173.2.1036 CAN_F7R1_FB14	753
5.173.2.1037 CAN_F7R1_FB15	754
5.173.2.1038 CAN_F7R1_FB16	754
5.173.2.1039 CAN_F7R1_FB17	754
5.173.2.1040 CAN_F7R1_FB18	754
5.173.2.1041 CAN_F7R1_FB19	754
5.173.2.1042 CAN_F7R1_FB2	754
5.173.2.1043 CAN_F7R1_FB20	754
5.173.2.1044 CAN_F7R1_FB21	754
5.173.2.1045 CAN_F7R1_FB22	755
5.173.2.1046 CAN_F7R1_FB23	755
5.173.2.1047 CAN_F7R1_FB24	755
5.173.2.1048 CAN_F7R1_FB25	755
5.173.2.1049 CAN_F7R1_FB26	755
5.173.2.1050 CAN_F7R1_FB27	755
5.173.2.1051 CAN_F7R1_FB28	755
5.173.2.1052 CAN_F7R1_FB29	755
5.173.2.1053 CAN_F7R1_FB3	756
5.173.2.1054 CAN_F7R1_FB30	756
5.173.2.1055 CAN_F7R1_FB31	756
5.173.2.1056 CAN_F7R1_FB4	756
5.173.2.1057 CAN_F7R1_FB5	756
5.173.2.1058 CAN_F7R1_FB6	756
5.173.2.1059 CAN_F7R1_FB7	756
5.173.2.1060 CAN_F7R1_FB8	756
5.173.2.1061 CAN_F7R1_FB9	757
5.173.2.1062 CAN_F7R2_FB0	757
5.173.2.1063 CAN_F7R2_FB1	757
5.173.2.1064 CAN_F7R2_FB10	757
5.173.2.1065 CAN_F7R2_FB11	757
5.173.2.1066 CAN_F7R2_FB12	757
5.173.2.1067 CAN_F7R2_FB13	757
5.173.2.1068 CAN_F7R2_FB14	757

5.173.2.1069 CAN_F7R2_FB15	758
5.173.2.1070 CAN_F7R2_FB16	758
5.173.2.1071 CAN_F7R2_FB17	758
5.173.2.1072 CAN_F7R2_FB18	758
5.173.2.1073 CAN_F7R2_FB19	758
5.173.2.1074 CAN_F7R2_FB2	758
5.173.2.1075 CAN_F7R2_FB20	758
5.173.2.1076 CAN_F7R2_FB21	758
5.173.2.1077 CAN_F7R2_FB22	759
5.173.2.1078 CAN_F7R2_FB23	759
5.173.2.1079 CAN_F7R2_FB24	759
5.173.2.1080 CAN_F7R2_FB25	759
5.173.2.1081 CAN_F7R2_FB26	759
5.173.2.1082 CAN_F7R2_FB27	759
5.173.2.1083 CAN_F7R2_FB28	759
5.173.2.1084 CAN_F7R2_FB29	759
5.173.2.1085 CAN_F7R2_FB3	760
5.173.2.1086 CAN_F7R2_FB30	760
5.173.2.1087 CAN_F7R2_FB31	760
5.173.2.1088 CAN_F7R2_FB4	760
5.173.2.1089 CAN_F7R2_FB5	760
5.173.2.1090 CAN_F7R2_FB6	760
5.173.2.1091 CAN_F7R2_FB7	760
5.173.2.1092 CAN_F7R2_FB8	760
5.173.2.1093 CAN_F7R2_FB9	761
5.173.2.1094 CAN_F8R1_FB0	761
5.173.2.1095 CAN_F8R1_FB1	761
5.173.2.1096 CAN_F8R1_FB10	761
5.173.2.1097 CAN_F8R1_FB11	761
5.173.2.1098 CAN_F8R1_FB12	761
5.173.2.1099 CAN_F8R1_FB13	761
5.173.2.1100 CAN_F8R1_FB14	761
5.173.2.1101 CAN_F8R1_FB15	762
5.173.2.1102 CAN_F8R1_FB16	762
5.173.2.1103 CAN_F8R1_FB17	762
5.173.2.1104 CAN_F8R1_FB18	762
5.173.2.1105 CAN_F8R1_FB19	762
5.173.2.1106 CAN_F8R1_FB2	762
5.173.2.1107 CAN_F8R1_FB20	762
5.173.2.1108 CAN_F8R1_FB21	762
5.173.2.1109 CAN_F8R1_FB22	763
5.173.2.1110 CAN_F8R1_FB23	763

5.173.2.1111 CAN_F8R1_FB24	763
5.173.2.1112 CAN_F8R1_FB25	763
5.173.2.1113 CAN_F8R1_FB26	763
5.173.2.1114 CAN_F8R1_FB27	763
5.173.2.1115 CAN_F8R1_FB28	763
5.173.2.1116 CAN_F8R1_FB29	763
5.173.2.1117 CAN_F8R1_FB3	764
5.173.2.1118 CAN_F8R1_FB30	764
5.173.2.1119 CAN_F8R1_FB31	764
5.173.2.1120 CAN_F8R1_FB4	764
5.173.2.1121 CAN_F8R1_FB5	764
5.173.2.1122 CAN_F8R1_FB6	764
5.173.2.1123 CAN_F8R1_FB7	764
5.173.2.1124 CAN_F8R1_FB8	764
5.173.2.1125 CAN_F8R1_FB9	765
5.173.2.1126 CAN_F8R2_FB0	765
5.173.2.1127 CAN_F8R2_FB1	765
5.173.2.1128 CAN_F8R2_FB10	765
5.173.2.1129 CAN_F8R2_FB11	765
5.173.2.1130 CAN_F8R2_FB12	765
5.173.2.1131 CAN_F8R2_FB13	765
5.173.2.1132 CAN_F8R2_FB14	765
5.173.2.1133 CAN_F8R2_FB15	766
5.173.2.1134 CAN_F8R2_FB16	766
5.173.2.1135 CAN_F8R2_FB17	766
5.173.2.1136 CAN_F8R2_FB18	766
5.173.2.1137 CAN_F8R2_FB19	766
5.173.2.1138 CAN_F8R2_FB2	766
5.173.2.1139 CAN_F8R2_FB20	766
5.173.2.1140 CAN_F8R2_FB21	766
5.173.2.1141 CAN_F8R2_FB22	767
5.173.2.1142 CAN_F8R2_FB23	767
5.173.2.1143 CAN_F8R2_FB24	767
5.173.2.1144 CAN_F8R2_FB25	767
5.173.2.1145 CAN_F8R2_FB26	767
5.173.2.1146 CAN_F8R2_FB27	767
5.173.2.1147 CAN_F8R2_FB28	767
5.173.2.1148 CAN_F8R2_FB29	767
5.173.2.1149 CAN_F8R2_FB3	768
5.173.2.1150 CAN_F8R2_FB30	768
5.173.2.1151 CAN_F8R2_FB31	768
5.173.2.1152 CAN_F8R2_FB4	768

5.173.2.1153 CAN_F8R2_FB5	768
5.173.2.1154 CAN_F8R2_FB6	768
5.173.2.1155 CAN_F8R2_FB7	768
5.173.2.1156 CAN_F8R2_FB8	768
5.173.2.1157 CAN_F8R2_FB9	769
5.173.2.1158 CAN_F9R1_FB0	769
5.173.2.1159 CAN_F9R1_FB1	769
5.173.2.1160 CAN_F9R1_FB10	769
5.173.2.1161 CAN_F9R1_FB11	769
5.173.2.1162 CAN_F9R1_FB12	769
5.173.2.1163 CAN_F9R1_FB13	769
5.173.2.1164 CAN_F9R1_FB14	769
5.173.2.1165 CAN_F9R1_FB15	770
5.173.2.1166 CAN_F9R1_FB16	770
5.173.2.1167 CAN_F9R1_FB17	770
5.173.2.1168 CAN_F9R1_FB18	770
5.173.2.1169 CAN_F9R1_FB19	770
5.173.2.1170 CAN_F9R1_FB2	770
5.173.2.1171 CAN_F9R1_FB20	770
5.173.2.1172 CAN_F9R1_FB21	770
5.173.2.1173 CAN_F9R1_FB22	771
5.173.2.1174 CAN_F9R1_FB23	771
5.173.2.1175 CAN_F9R1_FB24	771
5.173.2.1176 CAN_F9R1_FB25	771
5.173.2.1177 CAN_F9R1_FB26	771
5.173.2.1178 CAN_F9R1_FB27	771
5.173.2.1179 CAN_F9R1_FB28	771
5.173.2.1180 CAN_F9R1_FB29	771
5.173.2.1181 CAN_F9R1_FB3	772
5.173.2.1182 CAN_F9R1_FB30	772
5.173.2.1183 CAN_F9R1_FB31	772
5.173.2.1184 CAN_F9R1_FB4	772
5.173.2.1185 CAN_F9R1_FB5	772
5.173.2.1186 CAN_F9R1_FB6	772
5.173.2.1187 CAN_F9R1_FB7	772
5.173.2.1188 CAN_F9R1_FB8	772
5.173.2.1189 CAN_F9R1_FB9	773
5.173.2.1190 CAN_F9R2_FB0	773
5.173.2.1191 CAN_F9R2_FB1	773
5.173.2.1192 CAN_F9R2_FB10	773
5.173.2.1193 CAN_F9R2_FB11	773
5.173.2.1194 CAN_F9R2_FB12	773

5.173.2.1195 CAN_F9R2_FB13	773
5.173.2.1196 CAN_F9R2_FB14	773
5.173.2.1197 CAN_F9R2_FB15	774
5.173.2.1198 CAN_F9R2_FB16	774
5.173.2.1199 CAN_F9R2_FB17	774
5.173.2.1200 CAN_F9R2_FB18	774
5.173.2.1201 CAN_F9R2_FB19	774
5.173.2.1202 CAN_F9R2_FB2	774
5.173.2.1203 CAN_F9R2_FB20	774
5.173.2.1204 CAN_F9R2_FB21	774
5.173.2.1205 CAN_F9R2_FB22	775
5.173.2.1206 CAN_F9R2_FB23	775
5.173.2.1207 CAN_F9R2_FB24	775
5.173.2.1208 CAN_F9R2_FB25	775
5.173.2.1209 CAN_F9R2_FB26	775
5.173.2.1210 CAN_F9R2_FB27	775
5.173.2.1211 CAN_F9R2_FB28	775
5.173.2.1212 CAN_F9R2_FB29	775
5.173.2.1213 CAN_F9R2_FB3	776
5.173.2.1214 CAN_F9R2_FB30	776
5.173.2.1215 CAN_F9R2_FB31	776
5.173.2.1216 CAN_F9R2_FB4	776
5.173.2.1217 CAN_F9R2_FB5	776
5.173.2.1218 CAN_F9R2_FB6	776
5.173.2.1219 CAN_F9R2_FB7	776
5.173.2.1220 CAN_F9R2_FB8	776
5.173.2.1221 CAN_F9R2_FB9	777
5.173.2.1222 CAN_FA1R_FACT	777
5.173.2.1223 CAN_FA1R_FACT0	777
5.173.2.1224 CAN_FA1R_FACT1	777
5.173.2.1225 CAN_FA1R_FACT10	777
5.173.2.1226 CAN_FA1R_FACT11	777
5.173.2.1227 CAN_FA1R_FACT12	777
5.173.2.1228 CAN_FA1R_FACT13	777
5.173.2.1229 CAN_FA1R_FACT2	778
5.173.2.1230 CAN_FA1R_FACT3	778
5.173.2.1231 CAN_FA1R_FACT4	778
5.173.2.1232 CAN_FA1R_FACT5	778
5.173.2.1233 CAN_FA1R_FACT6	778
5.173.2.1234 CAN_FA1R_FACT7	778
5.173.2.1235 CAN_FA1R_FACT8	778
5.173.2.1236 CAN_FA1R_FACT9	778

5.173.2.1237 CAN_FFA1R_FFA	779
5.173.2.1238 CAN_FFA1R_FFA0	779
5.173.2.1239 CAN_FFA1R_FFA1	779
5.173.2.1240 CAN_FFA1R_FFA10	779
5.173.2.1241 CAN_FFA1R_FFA11	779
5.173.2.1242 CAN_FFA1R_FFA12	779
5.173.2.1243 CAN_FFA1R_FFA13	779
5.173.2.1244 CAN_FFA1R_FFA2	779
5.173.2.1245 CAN_FFA1R_FFA3	780
5.173.2.1246 CAN_FFA1R_FFA4	780
5.173.2.1247 CAN_FFA1R_FFA5	780
5.173.2.1248 CAN_FFA1R_FFA6	780
5.173.2.1249 CAN_FFA1R_FFA7	780
5.173.2.1250 CAN_FFA1R_FFA8	780
5.173.2.1251 CAN_FFA1R_FFA9	780
5.173.2.1252 CAN_FM1R_FBM	780
5.173.2.1253 CAN_FM1R_FBM0	781
5.173.2.1254 CAN_FM1R_FBM1	781
5.173.2.1255 CAN_FM1R_FBM10	781
5.173.2.1256 CAN_FM1R_FBM11	781
5.173.2.1257 CAN_FM1R_FBM12	781
5.173.2.1258 CAN_FM1R_FBM13	781
5.173.2.1259 CAN_FM1R_FBM2	781
5.173.2.1260 CAN_FM1R_FBM3	781
5.173.2.1261 CAN_FM1R_FBM4	782
5.173.2.1262 CAN_FM1R_FBM5	782
5.173.2.1263 CAN_FM1R_FBM6	782
5.173.2.1264 CAN_FM1R_FBM7	782
5.173.2.1265 CAN_FM1R_FBM8	782
5.173.2.1266 CAN_FM1R_FBM9	782
5.173.2.1267 CAN_FMR_FINIT	782
5.173.2.1268 CAN_FS1R_FSC	782
5.173.2.1269 CAN_FS1R_FSC0	783
5.173.2.1270 CAN_FS1R_FSC1	783
5.173.2.1271 CAN_FS1R_FSC10	783
5.173.2.1272 CAN_FS1R_FSC11	783
5.173.2.1273 CAN_FS1R_FSC12	783
5.173.2.1274 CAN_FS1R_FSC13	783
5.173.2.1275 CAN_FS1R_FSC2	783
5.173.2.1276 CAN_FS1R_FSC3	783
5.173.2.1277 CAN_FS1R_FSC4	784
5.173.2.1278 CAN_FS1R_FSC5	784

5.173.2.1279 CAN_FS1R_FSC6	784
5.173.2.1280 CAN_FS1R_FSC7	784
5.173.2.1281 CAN_FS1R_FSC8	784
5.173.2.1282 CAN_FS1R_FSC9	784
5.173.2.1283 CAN_IER_BOFIE	784
5.173.2.1284 CAN_IER_EPVIE	784
5.173.2.1285 CAN_IER_ERRIE	785
5.173.2.1286 CAN_IER_EWGLIE	785
5.173.2.1287 CAN_IER_FFIIE0	785
5.173.2.1288 CAN_IER_FFIIE1	785
5.173.2.1289 CAN_IER_FMPIE0	785
5.173.2.1290 CAN_IER_FMPIE1	785
5.173.2.1291 CAN_IER_FOVIE0	785
5.173.2.1292 CAN_IER_FOVIE1	785
5.173.2.1293 CAN_IER_LECIE	786
5.173.2.1294 CAN_IER_SLKIE	786
5.173.2.1295 CAN_IER_TMEIE	786
5.173.2.1296 CAN_IER_WKUIE	786
5.173.2.1297 CAN_MCR_ABOM	786
5.173.2.1298 CAN_MCR_AWUM	786
5.173.2.1299 CAN_MCR_INRQ	786
5.173.2.1300 CAN_MCR_NART	786
5.173.2.1301 CAN_MCR_RESET	787
5.173.2.1302 CAN_MCR_RFLM	787
5.173.2.1303 CAN_MCR_SLEEP	787
5.173.2.1304 CAN_MCR_TTCM	787
5.173.2.1305 CAN_MCR_TXFP	787
5.173.2.1306 CAN_MSR_ERRI	787
5.173.2.1307 CAN_MSR_INAK	787
5.173.2.1308 CAN_MSR_RX	787
5.173.2.1309 CAN_MSR_RXM	788
5.173.2.1310 CAN_MSR_SAMP	788
5.173.2.1311 CAN_MSR_SLAK	788
5.173.2.1312 CAN_MSR_SLAKI	788
5.173.2.1313 CAN_MSR_TXM	788
5.173.2.1314 CAN_MSR_WKUI	788
5.173.2.1315 CAN_RDH0R_DATA4	788
5.173.2.1316 CAN_RDH0R_DATA5	788
5.173.2.1317 CAN_RDH0R_DATA6	789
5.173.2.1318 CAN_RDH0R_DATA7	789
5.173.2.1319 CAN_RDH1R_DATA4	789
5.173.2.1320 CAN_RDH1R_DATA5	789

5.173.2.1321 CAN_RDH1R_DATA6	789
5.173.2.1322 CAN_RDH1R_DATA7	789
5.173.2.1323 CAN_RDL0R_DATA0	789
5.173.2.1324 CAN_RDL0R_DATA1	789
5.173.2.1325 CAN_RDL0R_DATA2	790
5.173.2.1326 CAN_RDL0R_DATA3	790
5.173.2.1327 CAN_RDL1R_DATA0	790
5.173.2.1328 CAN_RDL1R_DATA1	790
5.173.2.1329 CAN_RDL1R_DATA2	790
5.173.2.1330 CAN_RDL1R_DATA3	790
5.173.2.1331 CAN_RDT0R_DLC	790
5.173.2.1332 CAN_RDT0R_FMI	790
5.173.2.1333 CAN_RDT0R_TIME	791
5.173.2.1334 CAN_RDT1R_DLC	791
5.173.2.1335 CAN_RDT1R_FMI	791
5.173.2.1336 CAN_RDT1R_TIME	791
5.173.2.1337 CAN_RF0R_FMP0	791
5.173.2.1338 CAN_RF0R_FOVR0	791
5.173.2.1339 CAN_RF0R_FULL0	791
5.173.2.1340 CAN_RF0R_RFOM0	791
5.173.2.1341 CAN_RF1R_FMP1	792
5.173.2.1342 CAN_RF1R_FOVR1	792
5.173.2.1343 CAN_RF1R_FULL1	792
5.173.2.1344 CAN_RF1R_RFOM1	792
5.173.2.1345 CAN_RI0R_EXID	792
5.173.2.1346 CAN_RI0R_IDE	792
5.173.2.1347 CAN_RI0R_RTR	792
5.173.2.1348 CAN_RI0R_STID	792
5.173.2.1349 CAN_RI1R_EXID	793
5.173.2.1350 CAN_RI1R_IDE	793
5.173.2.1351 CAN_RI1R_RTR	793
5.173.2.1352 CAN_RI1R_STID	793
5.173.2.1353 CAN_TDH0R_DATA4	793
5.173.2.1354 CAN_TDH0R_DATA5	793
5.173.2.1355 CAN_TDH0R_DATA6	793
5.173.2.1356 CAN_TDH0R_DATA7	793
5.173.2.1357 CAN_TDH1R_DATA4	794
5.173.2.1358 CAN_TDH1R_DATA5	794
5.173.2.1359 CAN_TDH1R_DATA6	794
5.173.2.1360 CAN_TDH1R_DATA7	794
5.173.2.1361 CAN_TDH2R_DATA4	794
5.173.2.1362 CAN_TDH2R_DATA5	794

5.173.2.1363 CAN_TDH2R_DATA6	794
5.173.2.1364 CAN_TDH2R_DATA7	794
5.173.2.1365 CAN_TDL0R_DATA0	795
5.173.2.1366 CAN_TDL0R_DATA1	795
5.173.2.1367 CAN_TDL0R_DATA2	795
5.173.2.1368 CAN_TDL0R_DATA3	795
5.173.2.1369 CAN_TDL1R_DATA0	795
5.173.2.1370 CAN_TDL1R_DATA1	795
5.173.2.1371 CAN_TDL1R_DATA2	795
5.173.2.1372 CAN_TDL1R_DATA3	795
5.173.2.1373 CAN_TDL2R_DATA0	796
5.173.2.1374 CAN_TDL2R_DATA1	796
5.173.2.1375 CAN_TDL2R_DATA2	796
5.173.2.1376 CAN_TDL2R_DATA3	796
5.173.2.1377 CAN_TDT0R_DLC	796
5.173.2.1378 CAN_TDT0R_TGT	796
5.173.2.1379 CAN_TDT0R_TIME	796
5.173.2.1380 CAN_TDT1R_DLC	796
5.173.2.1381 CAN_TDT1R_TGT	797
5.173.2.1382 CAN_TDT1R_TIME	797
5.173.2.1383 CAN_TDT2R_DLC	797
5.173.2.1384 CAN_TDT2R_TGT	797
5.173.2.1385 CAN_TDT2R_TIME	797
5.173.2.1386 CAN_TI0R_EXID	797
5.173.2.1387 CAN_TI0R_IDE	797
5.173.2.1388 CAN_TI0R_RTR	797
5.173.2.1389 CAN_TI0R_STID	798
5.173.2.1390 CAN_TI0R_TXRQ	798
5.173.2.1391 CAN_TI1R_EXID	798
5.173.2.1392 CAN_TI1R_IDE	798
5.173.2.1393 CAN_TI1R_RTR	798
5.173.2.1394 CAN_TI1R_STID	798
5.173.2.1395 CAN_TI1R_TXRQ	798
5.173.2.1396 CAN_TI2R_EXID	798
5.173.2.1397 CAN_TI2R_IDE	799
5.173.2.1398 CAN_TI2R_RTR	799
5.173.2.1399 CAN_TI2R_STID	799
5.173.2.1400 CAN_TI2R_TXRQ	799
5.173.2.1401 CAN_TSR_ABRQ0	799
5.173.2.1402 CAN_TSR_ABRQ1	799
5.173.2.1403 CAN_TSR_ABRQ2	799
5.173.2.1404 CAN_TSR_ALST0	799

5.173.2.1405 CAN_TSR_ALST1	800
5.173.2.1406 CAN_TSR_ALST2	800
5.173.2.1407 CAN_TSR_CODE	800
5.173.2.1408 CAN_TSR_LOW	800
5.173.2.1409 CAN_TSR_LOW0	800
5.173.2.1410 CAN_TSR_LOW1	800
5.173.2.1411 CAN_TSR_LOW2	800
5.173.2.1412 CAN_TSR_RQCP0	800
5.173.2.1413 CAN_TSR_RQCP1	801
5.173.2.1414 CAN_TSR_RQCP2	801
5.173.2.1415 CAN_TSR_TERR0	801
5.173.2.1416 CAN_TSR_TERR1	801
5.173.2.1417 CAN_TSR_TERR2	801
5.173.2.1418 CAN_TSR_TME	801
5.173.2.1419 CAN_TSR_TME0	801
5.173.2.1420 CAN_TSR_TME1	801
5.173.2.1421 CAN_TSR_TME2	802
5.173.2.1422 CAN_TSR_TXOK0	802
5.173.2.1423 CAN_TSR_TXOK1	802
5.173.2.1424 CAN_TSR_TXOK2	802
5.173.2.1425 CRC_CR_RESET	802
5.173.2.1426 CRC_DR_DR	802
5.173.2.1427 CRC_IDR_IDR	802
5.173.2.1428 DAC_CR_BOFF1	802
5.173.2.1429 DAC_CR_BOFF2	803
5.173.2.1430 DAC_CR_DMAEN1	803
5.173.2.1431 DAC_CR_DMAEN2	803
5.173.2.1432 DAC_CR_DMAUDRIE1	803
5.173.2.1433 DAC_CR_DMAUDRIE2	803
5.173.2.1434 DAC_CR_EN1	803
5.173.2.1435 DAC_CR_EN2	803
5.173.2.1436 DAC_CR_MAMP1	803
5.173.2.1437 DAC_CR_MAMP1_0	804
5.173.2.1438 DAC_CR_MAMP1_1	804
5.173.2.1439 DAC_CR_MAMP1_2	804
5.173.2.1440 DAC_CR_MAMP1_3	804
5.173.2.1441 DAC_CR_MAMP2	804
5.173.2.1442 DAC_CR_MAMP2_0	804
5.173.2.1443 DAC_CR_MAMP2_1	804
5.173.2.1444 DAC_CR_MAMP2_2	804
5.173.2.1445 DAC_CR_MAMP2_3	805
5.173.2.1446 DAC_CR_TEN1	805

5.173.2.1447 DAC_CR_TEN2	805
5.173.2.1448 DAC_CR_TSEL1	805
5.173.2.1449 DAC_CR_TSEL1_0	805
5.173.2.1450 DAC_CR_TSEL1_1	805
5.173.2.1451 DAC_CR_TSEL1_2	805
5.173.2.1452 DAC_CR_TSEL2	805
5.173.2.1453 DAC_CR_TSEL2_0	806
5.173.2.1454 DAC_CR_TSEL2_1	806
5.173.2.1455 DAC_CR_TSEL2_2	806
5.173.2.1456 DAC_CR_WAVE1	806
5.173.2.1457 DAC_CR_WAVE1_0	806
5.173.2.1458 DAC_CR_WAVE1_1	806
5.173.2.1459 DAC_CR_WAVE2	806
5.173.2.1460 DAC_CR_WAVE2_0	806
5.173.2.1461 DAC_CR_WAVE2_1	807
5.173.2.1462 DAC_DHR12L1_DACC1DHR	807
5.173.2.1463 DAC_DHR12L2_DACC2DHR	807
5.173.2.1464 DAC_DHR12LD_DACC1DHR	807
5.173.2.1465 DAC_DHR12LD_DACC2DHR	807
5.173.2.1466 DAC_DHR12R1_DACC1DHR	807
5.173.2.1467 DAC_DHR12R2_DACC2DHR	807
5.173.2.1468 DAC_DHR12RD_DACC1DHR	807
5.173.2.1469 DAC_DHR12RD_DACC2DHR	808
5.173.2.1470 DAC_DHR8R1_DACC1DHR	808
5.173.2.1471 DAC_DHR8R2_DACC2DHR	808
5.173.2.1472 DAC_DHR8RD_DACC1DHR	808
5.173.2.1473 DAC_DHR8RD_DACC2DHR	808
5.173.2.1474 DAC_DOR1_DACC1DOR	808
5.173.2.1475 DAC_DOR2_DACC2DOR	808
5.173.2.1476 DAC_SR_DMAUDR1	808
5.173.2.1477 DAC_SR_DMAUDR2	809
5.173.2.1478 DAC_SWTRIGR_SWTRIG1	809
5.173.2.1479 DAC_SWTRIGR_SWTRIG2	809
5.173.2.1480 DBGMCU_CR_TRACE_MODE_0	809
5.173.2.1481 DBGMCU_CR_TRACE_MODE_1	809
5.173.2.1482 DFSDM_CHAWSCDR_AWFORD	809
5.173.2.1483 DFSDM_CHAWSCDR_AWFORD_0	809
5.173.2.1484 DFSDM_CHAWSCDR_AWFORD_1	809
5.173.2.1485 DFSDM_CHAWSCDR_AWFOSR	810
5.173.2.1486 DFSDM_CHAWSCDR_BKSCD	810
5.173.2.1487 DFSDM_CHAWSCDR_SCDT	810
5.173.2.1488 DFSDM_CHCFG1_CHEN	810

5.173.2.1489 DFSDM_CHCFG1_CHINSEL	810
5.173.2.1490 DFSDM_CHCFG1_CKABEN	810
5.173.2.1491 DFSDM_CHCFG1_CKOUTDIV	810
5.173.2.1492 DFSDM_CHCFG1_CKOUTSRC	810
5.173.2.1493 DFSDM_CHCFG1_DATMPX	811
5.173.2.1494 DFSDM_CHCFG1_DATMPX_0	811
5.173.2.1495 DFSDM_CHCFG1_DATMPX_1	811
5.173.2.1496 DFSDM_CHCFG1_DATPACK	811
5.173.2.1497 DFSDM_CHCFG1_DATPACK_0	811
5.173.2.1498 DFSDM_CHCFG1_DATPACK_1	811
5.173.2.1499 DFSDM_CHCFG1_DFSDMEN	811
5.173.2.1500 DFSDM_CHCFG1_SCDEN	811
5.173.2.1501 DFSDM_CHCFG1_SITP	812
5.173.2.1502 DFSDM_CHCFG1_SITP_0	812
5.173.2.1503 DFSDM_CHCFG1_SITP_1	812
5.173.2.1504 DFSDM_CHCFG1_SPICKSEL	812
5.173.2.1505 DFSDM_CHCFG1_SPICKSEL_0	812
5.173.2.1506 DFSDM_CHCFG1_SPICKSEL_1	812
5.173.2.1507 DFSDM_CHCFG2_DTRBS	812
5.173.2.1508 DFSDM_CHCFG2_OFFSET	812
5.173.2.1509 DFSDM_CHDATINR_INDATA0	813
5.173.2.1510 DFSDM_CHDATINR_INDATA1	813
5.173.2.1511 DFSDM_CHWDATR_WDATA	813
5.173.2.1512 DFSDM_FLTAWCFR_CLRAWHTF	813
5.173.2.1513 DFSDM_FLTAWCFR_CLRAWLTF	813
5.173.2.1514 DFSDM_FLTAWHTR_AWHT	813
5.173.2.1515 DFSDM_FLTAWHTR_BKAWH	813
5.173.2.1516 DFSDM_FLTAWLTR_AWLT	813
5.173.2.1517 DFSDM_FLTAWLTR_BKAWL	814
5.173.2.1518 DFSDM_FLTAWSR_AWHTF	814
5.173.2.1519 DFSDM_FLTAWSR_AWLTF	814
5.173.2.1520 DFSDM_FLTCNVTIMR_CNVCONT	814
5.173.2.1521 DFSDM_FLTCR1_AWFSEL	814
5.173.2.1522 DFSDM_FLTCR1_DFEN	814
5.173.2.1523 DFSDM_FLTCR1_FAST	814
5.173.2.1524 DFSDM_FLTCR1_JDMAEN	814
5.173.2.1525 DFSDM_FLTCR1_JEXTEN	815
5.173.2.1526 DFSDM_FLTCR1_JEXTEN_0	815
5.173.2.1527 DFSDM_FLTCR1_JEXTEN_1	815
5.173.2.1528 DFSDM_FLTCR1_JEXTSEL	815
5.173.2.1529 DFSDM_FLTCR1_JEXTSEL_0	815
5.173.2.1530 DFSDM_FLTCR1_JEXTSEL_1	815

5.173.2.1531 DFSDM_FLTCR1_JEXTSEL_2	815
5.173.2.1532 DFSDM_FLTCR1_JSCAN	815
5.173.2.1533 DFSDM_FLTCR1_JSWSTART	816
5.173.2.1534 DFSDM_FLTCR1_JSYNC	816
5.173.2.1535 DFSDM_FLTCR1_RCH	816
5.173.2.1536 DFSDM_FLTCR1_RCONT	816
5.173.2.1537 DFSDM_FLTCR1_RDMAEN	816
5.173.2.1538 DFSDM_FLTCR1_RSWSTART	816
5.173.2.1539 DFSDM_FLTCR1_RSYNC	816
5.173.2.1540 DFSDM_FLTCR2_AWDCH	817
5.173.2.1541 DFSDM_FLTCR2_AWDIE	817
5.173.2.1542 DFSDM_FLTCR2_CKABIE	817
5.173.2.1543 DFSDM_FLTCR2_EXCH	817
5.173.2.1544 DFSDM_FLTCR2_JEOCIE	817
5.173.2.1545 DFSDM_FLTCR2_JOVRIE	817
5.173.2.1546 DFSDM_FLTCR2_REOCIE	817
5.173.2.1547 DFSDM_FLTCR2_ROVRIE	817
5.173.2.1548 DFSDM_FLTCR2_SCDIE	818
5.173.2.1549 DFSDM_FLTEXMAX_EXMAX	818
5.173.2.1550 DFSDM_FLTEXMAX_EXMAXCH	818
5.173.2.1551 DFSDM_FLTEXMIN_EXMIN	818
5.173.2.1552 DFSDM_FLTEXMIN_EXMINCH	818
5.173.2.1553 DFSDM_FLTFCR_FORD	818
5.173.2.1554 DFSDM_FLTFCR_FORD_0	818
5.173.2.1555 DFSDM_FLTFCR_FORD_1	818
5.173.2.1556 DFSDM_FLTFCR_FORD_2	819
5.173.2.1557 DFSDM_FLTFCR_FOSR	819
5.173.2.1558 DFSDM_FLTFCR_IOSR	819
5.173.2.1559 DFSDM_FLTICR_CLRCKABF	819
5.173.2.1560 DFSDM_FLTICR_CLRJOVRF	819
5.173.2.1561 DFSDM_FLTICR_CLRRROVRF	819
5.173.2.1562 DFSDM_FLTICR_CLRSCSDF	819
5.173.2.1563 DFSDM_FLTISR_AWDF	819
5.173.2.1564 DFSDM_FLTISR_CKABF	820
5.173.2.1565 DFSDM_FLTISR_JCIP	820
5.173.2.1566 DFSDM_FLTISR_JEOCF	820
5.173.2.1567 DFSDM_FLTISR_JOVRF	820
5.173.2.1568 DFSDM_FLTISR_RCIP	820
5.173.2.1569 DFSDM_FLTISR_REOCF	820
5.173.2.1570 DFSDM_FLTISR_ROVRF	820
5.173.2.1571 DFSDM_FLTISR_SCDF	820
5.173.2.1572 DFSDM_FLTJCHGR_JCHG	821

5.173.2.1573 DFSDM_FLTJDATAR_JDATA	821
5.173.2.1574 DFSDM_FLTJDATAR_JDATACH	821
5.173.2.1575 DFSDM_FLTRDATAR_RDATA	821
5.173.2.1576 DFSDM_FLTRDATAR_RDATAACH	821
5.173.2.1577 DFSDM_FLTRDATAR_RPEND	821
5.173.2.1578 DMA2D_AMTCR_DT	821
5.173.2.1579 DMA2D_AMTCR_EN	821
5.173.2.1580 DMA2D_BGCMAR_MA	822
5.173.2.1581 DMA2D_BGCOLR_BLUE	822
5.173.2.1582 DMA2D_BGCOLR_GREEN	822
5.173.2.1583 DMA2D_BGCOLR_RED	822
5.173.2.1584 DMA2D_BGMAR_MA	822
5.173.2.1585 DMA2D_BGOR_LO	822
5.173.2.1586 DMA2D_BGPFCCR_ALPHA	822
5.173.2.1587 DMA2D_BGPFCCR_AM	822
5.173.2.1588 DMA2D_BGPFCCR_AM_0	823
5.173.2.1589 DMA2D_BGPFCCR_AM_1	823
5.173.2.1590 DMA2D_BGPFCCR_CCM	823
5.173.2.1591 DMA2D_BGPFCCR_CM	823
5.173.2.1592 DMA2D_BGPFCCR_CM_0	823
5.173.2.1593 DMA2D_BGPFCCR_CM_1	823
5.173.2.1594 DMA2D_BGPFCCR_CM_2	823
5.173.2.1595 DMA2D_BGPFCCR_CS	823
5.173.2.1596 DMA2D_BGPFCCR_START	824
5.173.2.1597 DMA2D_CR_ABORT	824
5.173.2.1598 DMA2D_CR_CAEIE	824
5.173.2.1599 DMA2D_CR_CEIE	824
5.173.2.1600 DMA2D_CR_CTCIE	824
5.173.2.1601 DMA2D_CR_MODE	824
5.173.2.1602 DMA2D_CR_START	824
5.173.2.1603 DMA2D_CR_SUSP	824
5.173.2.1604 DMA2D_CR_TCIE	825
5.173.2.1605 DMA2D_CR_TEIE	825
5.173.2.1606 DMA2D_CR_TWIE	825
5.173.2.1607 DMA2D_FGCMAR_MA	825
5.173.2.1608 DMA2D_FGCOLR_BLUE	825
5.173.2.1609 DMA2D_FGCOLR_GREEN	825
5.173.2.1610 DMA2D_FGCOLR_RED	825
5.173.2.1611 DMA2D_FGMAR_MA	825
5.173.2.1612 DMA2D_FGOR_LO	826
5.173.2.1613 DMA2D_FGPFCCR_ALPHA	826
5.173.2.1614 DMA2D_FGPFCCR_AM	826

5.173.2.1615 DMA2D_FGPFCCR_AM_0	826
5.173.2.1616 DMA2D_FGPFCCR_AM_1	826
5.173.2.1617 DMA2D_FGPFCCR_CCM	826
5.173.2.1618 DMA2D_FGPFCCR_CM	826
5.173.2.1619 DMA2D_FGPFCCR_CM_0	826
5.173.2.1620 DMA2D_FGPFCCR_CM_1	827
5.173.2.1621 DMA2D_FGPFCCR_CM_2	827
5.173.2.1622 DMA2D_FGPFCCR_CM_3 [1/2]	827
5.173.2.1623 DMA2D_FGPFCCR_CM_3 [2/2]	827
5.173.2.1624 DMA2D_FGPFCCR_CS	827
5.173.2.1625 DMA2D_FGPFCCR_START	827
5.173.2.1626 DMA2D_IFCR_CAEIF	827
5.173.2.1627 DMA2D_IFCR_CCEIF	828
5.173.2.1628 DMA2D_IFCR_CCTCIF	828
5.173.2.1629 DMA2D_IFCR_CTCIF	828
5.173.2.1630 DMA2D_IFCR_CTEIF	828
5.173.2.1631 DMA2D_IFCR_CTWIF	828
5.173.2.1632 DMA2D_IFSR_CCAEIF	828
5.173.2.1633 DMA2D_IFSR_CCEIF	828
5.173.2.1634 DMA2D_IFSR_CCTCIF	829
5.173.2.1635 DMA2D_IFSR_CTCIF	829
5.173.2.1636 DMA2D_IFSR_CTEIF	829
5.173.2.1637 DMA2D_IFSR_CTWIF	829
5.173.2.1638 DMA2D_ISR_CAEIF	829
5.173.2.1639 DMA2D_ISR_CEIF	829
5.173.2.1640 DMA2D_ISR_CTCIF	829
5.173.2.1641 DMA2D_ISR_TCIF	830
5.173.2.1642 DMA2D_ISR_TEIF	830
5.173.2.1643 DMA2D_ISR_TWIF	830
5.173.2.1644 DMA2D_LWR_LW	830
5.173.2.1645 DMA2D_NLR_NL	830
5.173.2.1646 DMA2D_NLR_PL	830
5.173.2.1647 DMA2D_OCOLR_ALPHA_1	830
5.173.2.1648 DMA2D_OCOLR_ALPHA_3	830
5.173.2.1649 DMA2D_OCOLR_ALPHA_4	831
5.173.2.1650 DMA2D_OCOLR_BLUE_1	831
5.173.2.1651 DMA2D_OCOLR_BLUE_2	831
5.173.2.1652 DMA2D_OCOLR_BLUE_3	831
5.173.2.1653 DMA2D_OCOLR_BLUE_4	831
5.173.2.1654 DMA2D_OCOLR_GREEN_1	831
5.173.2.1655 DMA2D_OCOLR_GREEN_2	831
5.173.2.1656 DMA2D_OCOLR_GREEN_3	832

5.173.2.1657 DMA2D_OCOLR_GREEN_4	832
5.173.2.1658 DMA2D_OCOLR_RED_1	832
5.173.2.1659 DMA2D_OCOLR_RED_2	832
5.173.2.1660 DMA2D_OCOLR_RED_3	832
5.173.2.1661 DMA2D_OCOLR_RED_4	832
5.173.2.1662 DMA2D_OMAR_MA	832
5.173.2.1663 DMA2D_OOR_LO	833
5.173.2.1664 DMA2D_OPFCCR_CM	833
5.173.2.1665 DMA2D_OPFCCR_CM_0	833
5.173.2.1666 DMA2D_OPFCCR_CM_1	833
5.173.2.1667 DMA2D_OPFCCR_CM_2	833
5.173.2.1668 ETH_MACCR_BL	833
5.173.2.1669 EXTI_EMR_MR0	833
5.173.2.1670 EXTI_EMR_MR1	834
5.173.2.1671 EXTI_EMR_MR10	834
5.173.2.1672 EXTI_EMR_MR11	834
5.173.2.1673 EXTI_EMR_MR12	834
5.173.2.1674 EXTI_EMR_MR13	834
5.173.2.1675 EXTI_EMR_MR14	834
5.173.2.1676 EXTI_EMR_MR15	834
5.173.2.1677 EXTI_EMR_MR16	834
5.173.2.1678 EXTI_EMR_MR17	835
5.173.2.1679 EXTI_EMR_MR18	835
5.173.2.1680 EXTI_EMR_MR19	835
5.173.2.1681 EXTI_EMR_MR2	835
5.173.2.1682 EXTI_EMR_MR23	835
5.173.2.1683 EXTI_EMR_MR3	835
5.173.2.1684 EXTI_EMR_MR4	835
5.173.2.1685 EXTI_EMR_MR5	835
5.173.2.1686 EXTI_EMR_MR6	836
5.173.2.1687 EXTI_EMR_MR7	836
5.173.2.1688 EXTI_EMR_MR8	836
5.173.2.1689 EXTI_EMR_MR9	836
5.173.2.1690 EXTI_FTSR_TR0	836
5.173.2.1691 EXTI_FTSR_TR1	836
5.173.2.1692 EXTI_FTSR_TR10	836
5.173.2.1693 EXTI_FTSR_TR11	836
5.173.2.1694 EXTI_FTSR_TR12	837
5.173.2.1695 EXTI_FTSR_TR13	837
5.173.2.1696 EXTI_FTSR_TR14	837
5.173.2.1697 EXTI_FTSR_TR15	837
5.173.2.1698 EXTI_FTSR_TR16	837

5.173.2.1699 EXTI_FTSR_TR17	837
5.173.2.1700 EXTI_FTSR_TR18	837
5.173.2.1701 EXTI_FTSR_TR19	837
5.173.2.1702 EXTI_FTSR_TR2	838
5.173.2.1703 EXTI_FTSR_TR23	838
5.173.2.1704 EXTI_FTSR_TR3	838
5.173.2.1705 EXTI_FTSR_TR4	838
5.173.2.1706 EXTI_FTSR_TR5	838
5.173.2.1707 EXTI_FTSR_TR6	838
5.173.2.1708 EXTI_FTSR_TR7	838
5.173.2.1709 EXTI_FTSR_TR8	838
5.173.2.1710 EXTI_FTSR_TR9	839
5.173.2.1711 EXTI_IMR_MR0	839
5.173.2.1712 EXTI_IMR_MR1	839
5.173.2.1713 EXTI_IMR_MR10	839
5.173.2.1714 EXTI_IMR_MR11	839
5.173.2.1715 EXTI_IMR_MR12	839
5.173.2.1716 EXTI_IMR_MR13	839
5.173.2.1717 EXTI_IMR_MR14	839
5.173.2.1718 EXTI_IMR_MR15	840
5.173.2.1719 EXTI_IMR_MR16	840
5.173.2.1720 EXTI_IMR_MR17	840
5.173.2.1721 EXTI_IMR_MR18	840
5.173.2.1722 EXTI_IMR_MR19	840
5.173.2.1723 EXTI_IMR_MR2	840
5.173.2.1724 EXTI_IMR_MR23	840
5.173.2.1725 EXTI_IMR_MR3	840
5.173.2.1726 EXTI_IMR_MR4	841
5.173.2.1727 EXTI_IMR_MR5	841
5.173.2.1728 EXTI_IMR_MR6	841
5.173.2.1729 EXTI_IMR_MR7	841
5.173.2.1730 EXTI_IMR_MR8	841
5.173.2.1731 EXTI_IMR_MR9	841
5.173.2.1732 EXTI_PR_PR0	841
5.173.2.1733 EXTI_PR_PR1	841
5.173.2.1734 EXTI_PR_PR10	842
5.173.2.1735 EXTI_PR_PR11	842
5.173.2.1736 EXTI_PR_PR12	842
5.173.2.1737 EXTI_PR_PR13	842
5.173.2.1738 EXTI_PR_PR14	842
5.173.2.1739 EXTI_PR_PR15	842
5.173.2.1740 EXTI_PR_PR16	842

5.173.2.1741 EXTI_PR_PR17	842
5.173.2.1742 EXTI_PR_PR18	843
5.173.2.1743 EXTI_PR_PR19	843
5.173.2.1744 EXTI_PR_PR2	843
5.173.2.1745 EXTI_PR_PR23	843
5.173.2.1746 EXTI_PR_PR3	843
5.173.2.1747 EXTI_PR_PR4	843
5.173.2.1748 EXTI_PR_PR5	843
5.173.2.1749 EXTI_PR_PR6	843
5.173.2.1750 EXTI_PR_PR7	844
5.173.2.1751 EXTI_PR_PR8	844
5.173.2.1752 EXTI_PR_PR9	844
5.173.2.1753 EXTI_RTSR_TR0	844
5.173.2.1754 EXTI_RTSR_TR1	844
5.173.2.1755 EXTI_RTSR_TR10	844
5.173.2.1756 EXTI_RTSR_TR11	844
5.173.2.1757 EXTI_RTSR_TR12	844
5.173.2.1758 EXTI_RTSR_TR13	845
5.173.2.1759 EXTI_RTSR_TR14	845
5.173.2.1760 EXTI_RTSR_TR15	845
5.173.2.1761 EXTI_RTSR_TR16	845
5.173.2.1762 EXTI_RTSR_TR17	845
5.173.2.1763 EXTI_RTSR_TR18	845
5.173.2.1764 EXTI_RTSR_TR19	845
5.173.2.1765 EXTI_RTSR_TR2	845
5.173.2.1766 EXTI_RTSR_TR23	846
5.173.2.1767 EXTI_RTSR_TR3	846
5.173.2.1768 EXTI_RTSR_TR4	846
5.173.2.1769 EXTI_RTSR_TR5	846
5.173.2.1770 EXTI_RTSR_TR6	846
5.173.2.1771 EXTI_RTSR_TR7	846
5.173.2.1772 EXTI_RTSR_TR8	846
5.173.2.1773 EXTI_RTSR_TR9	846
5.173.2.1774 EXTI_SWIER_SWIER0	847
5.173.2.1775 EXTI_SWIER_SWIER1	847
5.173.2.1776 EXTI_SWIER_SWIER10	847
5.173.2.1777 EXTI_SWIER_SWIER11	847
5.173.2.1778 EXTI_SWIER_SWIER12	847
5.173.2.1779 EXTI_SWIER_SWIER13	847
5.173.2.1780 EXTI_SWIER_SWIER14	847
5.173.2.1781 EXTI_SWIER_SWIER15	847
5.173.2.1782 EXTI_SWIER_SWIER16	848

5.173.2.1783 EXTI_SWIER_SWIER17	848
5.173.2.1784 EXTI_SWIER_SWIER18	848
5.173.2.1785 EXTI_SWIER_SWIER19	848
5.173.2.1786 EXTI_SWIER_SWIER2	848
5.173.2.1787 EXTI_SWIER_SWIER23	848
5.173.2.1788 EXTI_SWIER_SWIER3	848
5.173.2.1789 EXTI_SWIER_SWIER4	848
5.173.2.1790 EXTI_SWIER_SWIER5	849
5.173.2.1791 EXTI_SWIER_SWIER6	849
5.173.2.1792 EXTI_SWIER_SWIER7	849
5.173.2.1793 EXTI_SWIER_SWIER8	849
5.173.2.1794 EXTI_SWIER_SWIER9	849
5.173.2.1795 I2C_CCR_CCR	849
5.173.2.1796 I2C_CCR_DUTY	849
5.173.2.1797 I2C_CCR_FS	850
5.173.2.1798 I2C_CR1_ACK	850
5.173.2.1799 I2C_CR1_ALERT	850
5.173.2.1800 I2C_CR1_ENARP	850
5.173.2.1801 I2C_CR1_ENGC	850
5.173.2.1802 I2C_CR1_ENPEC	850
5.173.2.1803 I2C_CR1_NOSTRETCH	850
5.173.2.1804 I2C_CR1_PE	851
5.173.2.1805 I2C_CR1_PEC	851
5.173.2.1806 I2C_CR1_POS	851
5.173.2.1807 I2C_CR1_SMBTYPE	851
5.173.2.1808 I2C_CR1_SMBUS	851
5.173.2.1809 I2C_CR1_START	851
5.173.2.1810 I2C_CR1_STOP	851
5.173.2.1811 I2C_CR1_SWRST	852
5.173.2.1812 I2C_CR2_DMAEN	852
5.173.2.1813 I2C_CR2_FREQ	852
5.173.2.1814 I2C_CR2_FREQ_0	852
5.173.2.1815 I2C_CR2_FREQ_1	852
5.173.2.1816 I2C_CR2_FREQ_2	852
5.173.2.1817 I2C_CR2_FREQ_3	852
5.173.2.1818 I2C_CR2_FREQ_4	853
5.173.2.1819 I2C_CR2_FREQ_5	853
5.173.2.1820 I2C_CR2_ITBUFEN	853
5.173.2.1821 I2C_CR2_ITERREN	853
5.173.2.1822 I2C_CR2 ITEVTEN	853
5.173.2.1823 I2C_CR2_LAST	853
5.173.2.1824 I2C_DR_DR	853

5.173.2.1825 I2C_FLTR_ANOFF	854
5.173.2.1826 I2C_FLTR_DNF	854
5.173.2.1827 I2C_OAR1_ADD0	854
5.173.2.1828 I2C_OAR1_ADD1	854
5.173.2.1829 I2C_OAR1_ADD1_7	854
5.173.2.1830 I2C_OAR1_ADD2	854
5.173.2.1831 I2C_OAR1_ADD3	854
5.173.2.1832 I2C_OAR1_ADD4	854
5.173.2.1833 I2C_OAR1_ADD5	855
5.173.2.1834 I2C_OAR1_ADD6	855
5.173.2.1835 I2C_OAR1_ADD7	855
5.173.2.1836 I2C_OAR1_ADD8	855
5.173.2.1837 I2C_OAR1_ADD8_9	855
5.173.2.1838 I2C_OAR1_ADD9	855
5.173.2.1839 I2C_OAR1_ADDMODE	855
5.173.2.1840 I2C_OAR2_ADD2	855
5.173.2.1841 I2C_OAR2_ENDUAL	856
5.173.2.1842 I2C_SR1_ADD10	856
5.173.2.1843 I2C_SR1_ADDR	856
5.173.2.1844 I2C_SR1_AF	856
5.173.2.1845 I2C_SR1_ARLO	856
5.173.2.1846 I2C_SR1_BERR	856
5.173.2.1847 I2C_SR1_BTF	856
5.173.2.1848 I2C_SR1_OVR	857
5.173.2.1849 I2C_SR1_PECERR	857
5.173.2.1850 I2C_SR1_RXNE	857
5.173.2.1851 I2C_SR1_SB	857
5.173.2.1852 I2C_SR1_SMBALERT	857
5.173.2.1853 I2C_SR1_STOPF	857
5.173.2.1854 I2C_SR1_TIMEOUT	857
5.173.2.1855 I2C_SR1_TXE	858
5.173.2.1856 I2C_SR2_BUSY	858
5.173.2.1857 I2C_SR2_DUALF	858
5.173.2.1858 I2C_SR2_GENCALL	858
5.173.2.1859 I2C_SR2_MSL	858
5.173.2.1860 I2C_SR2_PEC	858
5.173.2.1861 I2C_SR2_SMBDEFAULT	858
5.173.2.1862 I2C_SR2_SMBHOST	859
5.173.2.1863 I2C_SR2_TRA	859
5.173.2.1864 I2C_TRISE_TRISE	859
5.173.2.1865 IWDG_KR_KEY	859
5.173.2.1866 IWDG_PR_PR	859

5.173.2.1867 IWDG_PR_PR_0	859
5.173.2.1868 IWDG_PR_PR_1	859
5.173.2.1869 IWDG_PR_PR_2	860
5.173.2.1870 IWDG_RLR_RL	860
5.173.2.1871 IWDG_SR_PVU	860
5.173.2.1872 IWDG_SR_RVU	860
5.173.2.1873 LTDC_AWCR_AAH	860
5.173.2.1874 LTDC_AWCR_AAW	860
5.173.2.1875 LTDC_BCCR_BCBLUE	860
5.173.2.1876 LTDC_BCCR_BCGREEN	861
5.173.2.1877 LTDC_BCCR_BCRED	861
5.173.2.1878 LTDC_BPCR_AHBP	861
5.173.2.1879 LTDC_BPCR_AVBP	861
5.173.2.1880 LTDC_CDSR_HDES	861
5.173.2.1881 LTDC_CDSR_HSYNCS	861
5.173.2.1882 LTDC_CDSR_VDES	861
5.173.2.1883 LTDC_CDSR_VSYNCS	861
5.173.2.1884 LTDC_CPSR_CXPOS	862
5.173.2.1885 LTDC_CPSR_CYPOS	862
5.173.2.1886 LTDC_GCR_DBW	862
5.173.2.1887 LTDC_GCR_DEN	862
5.173.2.1888 LTDC_GCR_DEPOL	862
5.173.2.1889 LTDC_GCR_DGW	862
5.173.2.1890 LTDC_GCR_DRW	862
5.173.2.1891 LTDC_GCR_HSPOL	862
5.173.2.1892 LTDC_GCR_LTDCEN	863
5.173.2.1893 LTDC_GCR_PCPOL	863
5.173.2.1894 LTDC_GCR_VSPOL	863
5.173.2.1895 LTDC_ICR_CFUIF	863
5.173.2.1896 LTDC_ICR_CLIF	863
5.173.2.1897 LTDC_ICR_CRRIF	863
5.173.2.1898 LTDC_ICR_CTERRIF	863
5.173.2.1899 LTDC_IER_FUIE	863
5.173.2.1900 LTDC_IER_LIE	864
5.173.2.1901 LTDC_IER_RRIE	864
5.173.2.1902 LTDC_IER_TERRIE	864
5.173.2.1903 LTDC_ISR_FUIF	864
5.173.2.1904 LTDC_ISR_LIF	864
5.173.2.1905 LTDC_ISR_RRIF	864
5.173.2.1906 LTDC_ISR_TERRIF	864
5.173.2.1907 LTDC_LIPCR_LIPOS	864
5.173.2.1908 LTDC_LxBFCR_BF1	865

5.173.2.1909 LTDC_LxBFCR_BF2	865
5.173.2.1910 LTDC_LxCACR_CONSTA	865
5.173.2.1911 LTDC_LxCFBAR_CFBADD	865
5.173.2.1912 LTDC_LxCFBLNR_CFBLNBR	865
5.173.2.1913 LTDC_LxCFBLR_CFBLL	865
5.173.2.1914 LTDC_LxCFBLR_CFBP	865
5.173.2.1915 LTDC_LxCKCR_CKBLUE	865
5.173.2.1916 LTDC_LxCKCR_CKGREEN	866
5.173.2.1917 LTDC_LxCKCR_CKRED	866
5.173.2.1918 LTDC_LxCLUTWR_BLUE	866
5.173.2.1919 LTDC_LxCLUTWR_CLUTADD	866
5.173.2.1920 LTDC_LxCLUTWR_GREEN	866
5.173.2.1921 LTDC_LxCLUTWR_RED	866
5.173.2.1922 LTDC_LxCR_CLUTEN	866
5.173.2.1923 LTDC_LxCR_COLKEN	866
5.173.2.1924 LTDC_LxCR_LEN	867
5.173.2.1925 LTDC_LxDCCR_DCALPHA	867
5.173.2.1926 LTDC_LxDCCR_DCBLUE	867
5.173.2.1927 LTDC_LxDCCR_DCGREEN	867
5.173.2.1928 LTDC_LxDCCR_DCRED	867
5.173.2.1929 LTDC_LxPFCR_PF	867
5.173.2.1930 LTDC_LxWHPCR_WHSPPOS	867
5.173.2.1931 LTDC_LxWHPCR_WHSTPOS	867
5.173.2.1932 LTDC_LxWVPCR_WVSPPOS	868
5.173.2.1933 LTDC_LxWVPCR_WVSTPOS	868
5.173.2.1934 LTDC_SRCR_IMR	868
5.173.2.1935 LTDC_SRCR_VBR	868
5.173.2.1936 LTDC_SSCR_HSW	868
5.173.2.1937 LTDC_SSCR_VSH	868
5.173.2.1938 LTDC_TWCR_TOTALH	868
5.173.2.1939 LTDC_TWCR_TOTALW	868
5.173.2.1940 PWR_CR_ADCDC1	869
5.173.2.1941 PWR_CR_CSBF	869
5.173.2.1942 PWR_CR_CWUF	869
5.173.2.1943 PWR_CR_DBP	869
5.173.2.1944 PWR_CR_FISSR	869
5.173.2.1945 PWR_CR_FMSSR	869
5.173.2.1946 PWR_CR_FPDS	869
5.173.2.1947 PWR_CR_LPDS	870
5.173.2.1948 PWR_CR_LPLVDS	870
5.173.2.1949 PWR_CR_LPUDS	870
5.173.2.1950 PWR_CR_MRLVDS	870

5.173.2.1951 PWR_CR_MRUDS	870
5.173.2.1952 PWR_CR_ODEN	870
5.173.2.1953 PWR_CR_ODSWEN	870
5.173.2.1954 PWR_CR_PDDS	871
5.173.2.1955 PWR_CR_PLS	871
5.173.2.1956 PWR_CR_PLS_0	871
5.173.2.1957 PWR_CR_PLS_1	871
5.173.2.1958 PWR_CR_PLS_2	871
5.173.2.1959 PWR_CR_PLS_LEV0	871
5.173.2.1960 PWR_CR_PLS_LEV1	871
5.173.2.1961 PWR_CR_PLS_LEV2	872
5.173.2.1962 PWR_CR_PLS_LEV3	872
5.173.2.1963 PWR_CR_PLS_LEV4	872
5.173.2.1964 PWR_CR_PLS_LEV5	872
5.173.2.1965 PWR_CR_PLS_LEV6	872
5.173.2.1966 PWR_CR_PLS_LEV7	872
5.173.2.1967 PWR_CR_PVDE	872
5.173.2.1968 PWR_CR_UDEN	873
5.173.2.1969 PWR_CR_UDEN_0	873
5.173.2.1970 PWR_CR_UDEN_1	873
5.173.2.1971 PWR_CR_VOS	873
5.173.2.1972 PWR_CR_VOS_0	873
5.173.2.1973 PWR_CR_VOS_1	873
5.173.2.1974 PWR_CSR_BRE	873
5.173.2.1975 PWR_CSR_BRR	874
5.173.2.1976 PWR_CSR_EWUP	874
5.173.2.1977 PWR_CSR_ODRDY	874
5.173.2.1978 PWR_CSR_ODSWRDY	874
5.173.2.1979 PWR_CSR_PVDO	874
5.173.2.1980 PWR_CSR_SBF	874
5.173.2.1981 PWR_CSR_UDSWRDY	874
5.173.2.1982 PWR_CSR_VOSRDY	875
5.173.2.1983 PWR_CSR_WUF	875
5.173.2.1984 PWR_CSR_WUPP	875
5.173.2.1985 RCC_CFGR_HPRE	875
5.173.2.1986 RCC_CFGR_HPRE_0	875
5.173.2.1987 RCC_CFGR_HPRE_1	875
5.173.2.1988 RCC_CFGR_HPRE_2	875
5.173.2.1989 RCC_CFGR_HPRE_3	876
5.173.2.1990 RCC_CFGR_HPRE_DIV1	876
5.173.2.1991 RCC_CFGR_HPRE_DIV128	876
5.173.2.1992 RCC_CFGR_HPRE_DIV16	876

5.173.2.1993 RCC_CFGR_HPRE_DIV2	876
5.173.2.1994 RCC_CFGR_HPRE_DIV256	876
5.173.2.1995 RCC_CFGR_HPRE_DIV4	876
5.173.2.1996 RCC_CFGR_HPRE_DIV512	876
5.173.2.1997 RCC_CFGR_HPRE_DIV64	877
5.173.2.1998 RCC_CFGR_HPRE_DIV8	877
5.173.2.1999 RCC_CFGR_PPRE1	877
5.173.2.2000 RCC_CFGR_PPRE1_0	877
5.173.2.2001 RCC_CFGR_PPRE1_1	877
5.173.2.2002 RCC_CFGR_PPRE1_2	877
5.173.2.2003 RCC_CFGR_PPRE1_DIV1	877
5.173.2.2004 RCC_CFGR_PPRE1_DIV16	877
5.173.2.2005 RCC_CFGR_PPRE1_DIV2	878
5.173.2.2006 RCC_CFGR_PPRE1_DIV4	878
5.173.2.2007 RCC_CFGR_PPRE1_DIV8	878
5.173.2.2008 RCC_CFGR_PPRE2	878
5.173.2.2009 RCC_CFGR_PPRE2_0	878
5.173.2.2010 RCC_CFGR_PPRE2_1	878
5.173.2.2011 RCC_CFGR_PPRE2_2	878
5.173.2.2012 RCC_CFGR_PPRE2_DIV1	878
5.173.2.2013 RCC_CFGR_PPRE2_DIV16	879
5.173.2.2014 RCC_CFGR_PPRE2_DIV2	879
5.173.2.2015 RCC_CFGR_PPRE2_DIV4	879
5.173.2.2016 RCC_CFGR_PPRE2_DIV8	879
5.173.2.2017 RCC_CFGR_RTCPRE_4	879
5.173.2.2018 RCC_CFGR_SW	879
5.173.2.2019 RCC_CFGR_SW_0	879
5.173.2.2020 RCC_CFGR_SW_1	879
5.173.2.2021 RCC_CFGR_SW_HSE	880
5.173.2.2022 RCC_CFGR_SW_HSI	880
5.173.2.2023 RCC_CFGR_SW_PLL	880
5.173.2.2024 RCC_CFGR_SWS	880
5.173.2.2025 RCC_CFGR_SWS_0	880
5.173.2.2026 RCC_CFGR_SWS_1	880
5.173.2.2027 RCC_CFGR_SWS_HSE	880
5.173.2.2028 RCC_CFGR_SWS_HSI	880
5.173.2.2029 RCC_CFGR_SWS_PLL	881
5.173.2.2030 RCC_CR_HSICAL_0	881
5.173.2.2031 RCC_CR_HSICAL_1	881
5.173.2.2032 RCC_CR_HSICAL_2	881
5.173.2.2033 RCC_CR_HSICAL_3	881
5.173.2.2034 RCC_CR_HSICAL_4	881

5.173.2.2035 RCC_CR_HSICAL_5	881
5.173.2.2036 RCC_CR_HSICAL_6	882
5.173.2.2037 RCC_CR_HSICAL_7	882
5.173.2.2038 RCC_CR_HSITRIM_0	882
5.173.2.2039 RCC_CR_HSITRIM_1	882
5.173.2.2040 RCC_CR_HSITRIM_2	882
5.173.2.2041 RCC_CR_HSITRIM_3	882
5.173.2.2042 RCC_CR_HSITRIM_4	882
5.173.2.2043 SAI_GCR_SYNCIN	882
5.173.2.2044 SAI_GCR_SYNCIN_0	883
5.173.2.2045 SAI_GCR_SYNCIN_1	883
5.173.2.2046 SAI_GCR_SYNCOUT	883
5.173.2.2047 SAI_GCR_SYNCOUT_0	883
5.173.2.2048 SAI_GCR_SYNCOUT_1	883
5.173.2.2049 SAI_xCLRFR_CAFSDET	883
5.173.2.2050 SAI_xCLRFR_CCNRDY	883
5.173.2.2051 SAI_xCLRFR_CFREQ	884
5.173.2.2052 SAI_xCLRFR_CLFSDET	884
5.173.2.2053 SAI_xCLRFR_CMUTEDET	884
5.173.2.2054 SAI_xCLRFR_COVRUDR	884
5.173.2.2055 SAI_xCLRFR_CWCKCFG	884
5.173.2.2056 SAI_xCR1_CKSTR	884
5.173.2.2057 SAI_xCR1_DMAEN	884
5.173.2.2058 SAI_xCR1_DS	885
5.173.2.2059 SAI_xCR1_DS_0	885
5.173.2.2060 SAI_xCR1_DS_1	885
5.173.2.2061 SAI_xCR1_DS_2	885
5.173.2.2062 SAI_xCR1 LSBFIRST	885
5.173.2.2063 SAI_xCR1_MCKDIV	885
5.173.2.2064 SAI_xCR1_MCKDIV_0	885
5.173.2.2065 SAI_xCR1_MCKDIV_1	886
5.173.2.2066 SAI_xCR1_MCKDIV_2	886
5.173.2.2067 SAI_xCR1_MCKDIV_3	886
5.173.2.2068 SAI_xCR1_MODE	886
5.173.2.2069 SAI_xCR1_MODE_0	886
5.173.2.2070 SAI_xCR1_MODE_1	886
5.173.2.2071 SAI_xCR1_MONO	886
5.173.2.2072 SAI_xCR1_NODIV	887
5.173.2.2073 SAI_xCR1_OUTDRV	887
5.173.2.2074 SAI_xCR1_PRTCFG	887
5.173.2.2075 SAI_xCR1_PRTCFG_0	887
5.173.2.2076 SAI_xCR1_PRTCFG_1	887

5.173.2.2077 SAI_xCR1_SAIEN	887
5.173.2.2078 SAI_xCR1_SYNCEN	887
5.173.2.2079 SAI_xCR1_SYNCEN_0	888
5.173.2.2080 SAI_xCR1_SYNCEN_1	888
5.173.2.2081 SAI_xCR2_COMP	888
5.173.2.2082 SAI_xCR2_COMP_0	888
5.173.2.2083 SAI_xCR2_COMP_1	888
5.173.2.2084 SAI_xCR2_CPL	888
5.173.2.2085 SAI_xCR2_FFLUSH	888
5.173.2.2086 SAI_xCR2_FTH	889
5.173.2.2087 SAI_xCR2_FTH_0	889
5.173.2.2088 SAI_xCR2_FTH_1	889
5.173.2.2089 SAI_xCR2_MUTE	889
5.173.2.2090 SAI_xCR2_MUTECNT	889
5.173.2.2091 SAI_xCR2_MUTECNT_0	889
5.173.2.2092 SAI_xCR2_MUTECNT_1	889
5.173.2.2093 SAI_xCR2_MUTECNT_2	890
5.173.2.2094 SAI_xCR2_MUTECNT_3	890
5.173.2.2095 SAI_xCR2_MUTECNT_4	890
5.173.2.2096 SAI_xCR2_MUTECNT_5	890
5.173.2.2097 SAI_xCR2_MUTEVAL	890
5.173.2.2098 SAI_xCR2_TRIS	890
5.173.2.2099 SAI_xFRCR_FRL	890
5.173.2.2100 SAI_xFRCR_FRL_0	891
5.173.2.2101 SAI_xFRCR_FRL_1	891
5.173.2.2102 SAI_xFRCR_FRL_2	891
5.173.2.2103 SAI_xFRCR_FRL_3	891
5.173.2.2104 SAI_xFRCR_FRL_4	891
5.173.2.2105 SAI_xFRCR_FRL_5	891
5.173.2.2106 SAI_xFRCR_FRL_6	891
5.173.2.2107 SAI_xFRCR_FRL_7	891
5.173.2.2108 SAI_xFRCR_FSALL	892
5.173.2.2109 SAI_xFRCR_FSALL_0	892
5.173.2.2110 SAI_xFRCR_FSALL_1	892
5.173.2.2111 SAI_xFRCR_FSALL_2	892
5.173.2.2112 SAI_xFRCR_FSALL_3	892
5.173.2.2113 SAI_xFRCR_FSALL_4	892
5.173.2.2114 SAI_xFRCR_FSALL_5	892
5.173.2.2115 SAI_xFRCR_FSALL_6	893
5.173.2.2116 SAI_xFRCR_FSDEF	893
5.173.2.2117 SAI_xFRCR_FSOFF	893
5.173.2.2118 SAI_xFRCR_FSPOL	893

5.173.2.2119 SAI_xIMR_AFSDETIE	893
5.173.2.2120 SAI_xIMR_CNRDYIE	893
5.173.2.2121 SAI_xIMR_FREQIE	893
5.173.2.2122 SAI_xIMR_LFSDETIE	894
5.173.2.2123 SAI_xIMR_MUTEDETIE	894
5.173.2.2124 SAI_xIMR_OVRUDRIE	894
5.173.2.2125 SAI_xIMR_WCKCFGIE	894
5.173.2.2126 SAI_xSLOTR_FBOFF	894
5.173.2.2127 SAI_xSLOTR_FBOFF_0	894
5.173.2.2128 SAI_xSLOTR_FBOFF_1	894
5.173.2.2129 SAI_xSLOTR_FBOFF_2	895
5.173.2.2130 SAI_xSLOTR_FBOFF_3	895
5.173.2.2131 SAI_xSLOTR_FBOFF_4	895
5.173.2.2132 SAI_xSLOTR_NBSLOT	895
5.173.2.2133 SAI_xSLOTR_NBSLOT_0	895
5.173.2.2134 SAI_xSLOTR_NBSLOT_1	895
5.173.2.2135 SAI_xSLOTR_NBSLOT_2	895
5.173.2.2136 SAI_xSLOTR_NBSLOT_3	896
5.173.2.2137 SAI_xSLOTR_SLOTEN	896
5.173.2.2138 SAI_xSLOTR_SLOTsz	896
5.173.2.2139 SAI_xSLOTR_SLOTsz_0	896
5.173.2.2140 SAI_xSLOTR_SLOTsz_1	896
5.173.2.2141 SAI_xSR_AFSDET	896
5.173.2.2142 SAI_xSR_CNRDY	896
5.173.2.2143 SAI_xSR_FLVL	897
5.173.2.2144 SAI_xSR_FLVL_0	897
5.173.2.2145 SAI_xSR_FLVL_1	897
5.173.2.2146 SAI_xSR_FLVL_2	897
5.173.2.2147 SAI_xSR_FREQ	897
5.173.2.2148 SAI_xSR_LFSDET	897
5.173.2.2149 SAI_xSR_MUTEDET	897
5.173.2.2150 SAI_xSR_OVRUDR	898
5.173.2.2151 SAI_xSR_WCKCFG	898
5.173.2.2152 SDIO_ARG_CMDARG	898
5.173.2.2153 SDIO_CLKCR_BYPASS	898
5.173.2.2154 SDIO_CLKCR_CLKDIV	898
5.173.2.2155 SDIO_CLKCR_CLKEN	898
5.173.2.2156 SDIO_CLKCR_HWFC_EN	898
5.173.2.2157 SDIO_CLKCR_NEGEDGE	899
5.173.2.2158 SDIO_CLKCR_PWRSAV	899
5.173.2.2159 SDIO_CLKCR_WIDBUS	899
5.173.2.2160 SDIO_CLKCR_WIDBUS_0	899

5.173.2.2161 SDIO_CLKCR_WIDBUS_1	899
5.173.2.2162 SDIO_CMD_CEATACMD	899
5.173.2.2163 SDIO_CMD_CMDINDEX	899
5.173.2.2164 SDIO_CMD_CPSMEN	900
5.173.2.2165 SDIO_CMD_ENCMDCOMPL	900
5.173.2.2166 SDIO_CMD_NIEN	900
5.173.2.2167 SDIO_CMD_SDIOSUSPEND	900
5.173.2.2168 SDIO_CMD_WAITINT	900
5.173.2.2169 SDIO_CMD_WAITPEND	900
5.173.2.2170 SDIO_CMD_WAITRESP	900
5.173.2.2171 SDIO_CMD_WAITRESP_0	901
5.173.2.2172 SDIO_CMD_WAITRESP_1	901
5.173.2.2173 SDIO_DCOUNT_DATACOUNT	901
5.173.2.2174 SDIO_DCTRL_DBLOCKSIZE	901
5.173.2.2175 SDIO_DCTRL_DBLOCKSIZE_0	901
5.173.2.2176 SDIO_DCTRL_DBLOCKSIZE_1	901
5.173.2.2177 SDIO_DCTRL_DBLOCKSIZE_2	901
5.173.2.2178 SDIO_DCTRL_DBLOCKSIZE_3	901
5.173.2.2179 SDIO_DCTRL_DMAEN	902
5.173.2.2180 SDIO_DCTRL_DTDIR	902
5.173.2.2181 SDIO_DCTRL_DTEN	902
5.173.2.2182 SDIO_DCTRL_DTMODE	902
5.173.2.2183 SDIO_DCTRL_RWMOD	902
5.173.2.2184 SDIO_DCTRL_RWSTART	902
5.173.2.2185 SDIO_DCTRL_RWSTOP	902
5.173.2.2186 SDIO_DCTRL_SDIOEN	903
5.173.2.2187 SDIO_DLLEN_DATALENGTH	903
5.173.2.2188 SDIO_DTIMER_DATATIME	903
5.173.2.2189 SDIO_FIFO_FIFODATA	903
5.173.2.2190 SDIO_FIFOCNT_FIFOCOUNT	903
5.173.2.2191 SDIO_ICR_CCRCFAILC	903
5.173.2.2192 SDIO_ICR_CEATAENDC	903
5.173.2.2193 SDIO_ICR_CMDRENDC	904
5.173.2.2194 SDIO_ICR_CMDSENTC	904
5.173.2.2195 SDIO_ICR_CTIMEOUTC	904
5.173.2.2196 SDIO_ICR_DATAENDC	904
5.173.2.2197 SDIO_ICR_DBCKENDC	904
5.173.2.2198 SDIO_ICR_DCRCFAILC	904
5.173.2.2199 SDIO_ICR_DTIMEOUTC	904
5.173.2.2200 SDIO_ICR_RXOVERRC	905
5.173.2.2201 SDIO_ICR_SDIOITC	905
5.173.2.2202 SDIO_ICR_STBITERRC	905

5.173.2.2203 SDIO_ICR_TXUNDERRC	905
5.173.2.2204 SDIO_MASK_CCRCFAILIE	905
5.173.2.2205 SDIO_MASK_CEATAENDIE	905
5.173.2.2206 SDIO_MASK_CMDACTIE	905
5.173.2.2207 SDIO_MASK_CMDRENDIE	906
5.173.2.2208 SDIO_MASK_CMDSENTIE	906
5.173.2.2209 SDIO_MASK_CTIMEOUTIE	906
5.173.2.2210 SDIO_MASK_DATAENDIE	906
5.173.2.2211 SDIO_MASK_DBCKENDIE	906
5.173.2.2212 SDIO_MASK_DCRCFAILIE	906
5.173.2.2213 SDIO_MASK_DTIMEOUTIE	906
5.173.2.2214 SDIO_MASK_RXACTIE	907
5.173.2.2215 SDIO_MASK_RXDAVLIE	907
5.173.2.2216 SDIO_MASK_RXFIFOEIE	907
5.173.2.2217 SDIO_MASK_RXFIFOFIE	907
5.173.2.2218 SDIO_MASK_RXFIFOHFIE	907
5.173.2.2219 SDIO_MASK_RXOVERRIE	907
5.173.2.2220 SDIO_MASK_SDIOITIE	907
5.173.2.2221 SDIO_MASK_STBITERRIE	908
5.173.2.2222 SDIO_MASK_TXACTIE	908
5.173.2.2223 SDIO_MASK_TXDAVLIE	908
5.173.2.2224 SDIO_MASK_TXFIFOEIE	908
5.173.2.2225 SDIO_MASK_TXFIFOFIE	908
5.173.2.2226 SDIO_MASK_TXFIFOHEIE	908
5.173.2.2227 SDIO_MASK_TXUNDERRIE	908
5.173.2.2228 SDIO_POWER_PWRCTRL	909
5.173.2.2229 SDIO_POWER_PWRCTRL_0	909
5.173.2.2230 SDIO_POWER_PWRCTRL_1	909
5.173.2.2231 SDIO_RESP0_CARDSTATUS0	909
5.173.2.2232 SDIO_RESP1_CARDSTATUS1	909
5.173.2.2233 SDIO_RESP2_CARDSTATUS2	909
5.173.2.2234 SDIO_RESP3_CARDSTATUS3	909
5.173.2.2235 SDIO_RESP4_CARDSTATUS4	909
5.173.2.2236 SDIO_RESPCMD_RESPCMD	910
5.173.2.2237 SDIO_STA_CCRCFAIL	910
5.173.2.2238 SDIO_STA_CEATAEND	910
5.173.2.2239 SDIO_STA_CMDACT	910
5.173.2.2240 SDIO_STA_CMDREND	910
5.173.2.2241 SDIO_STA_CMDSENT	910
5.173.2.2242 SDIO_STA_CTIMEOUT	910
5.173.2.2243 SDIO_STA_DATAEND	911
5.173.2.2244 SDIO_STA_DBCKEND	911

5.173.2.2245 SDIO_STA_DCRCFAIL	911
5.173.2.2246 SDIO_STA_DTIMEOUT	911
5.173.2.2247 SDIO_STA_RXACT	911
5.173.2.2248 SDIO_STA_RXDAVL	911
5.173.2.2249 SDIO_STA_RXFIFOE	911
5.173.2.2250 SDIO_STA_RXFIFOF	912
5.173.2.2251 SDIO_STA_RXFIFOHF	912
5.173.2.2252 SDIO_STA_RXOVERR	912
5.173.2.2253 SDIO_STA_SDIOIT	912
5.173.2.2254 SDIO_STA_STBITERR	912
5.173.2.2255 SDIO_STA_TXACT	912
5.173.2.2256 SDIO_STA_TXDAVL	912
5.173.2.2257 SDIO_STA_TXFIFOE	913
5.173.2.2258 SDIO_STA_TXFIFOF	913
5.173.2.2259 SDIO_STA_TXFIFOHE	913
5.173.2.2260 SDIO_STA_TXUNDERR	913
5.173.2.2261 SPI_CR1_BIDIMODE	913
5.173.2.2262 SPI_CR1_BIDIOE	913
5.173.2.2263 SPI_CR1_BR	913
5.173.2.2264 SPI_CR1_BR_0	914
5.173.2.2265 SPI_CR1_BR_1	914
5.173.2.2266 SPI_CR1_BR_2	914
5.173.2.2267 SPI_CR1_CPHA	914
5.173.2.2268 SPI_CR1_CPOL	914
5.173.2.2269 SPI_CR1_CRCEN	914
5.173.2.2270 SPI_CR1_CRCNEXT	914
5.173.2.2271 SPI_CR1_DFF	915
5.173.2.2272 SPI_CR1_LSBFIRST	915
5.173.2.2273 SPI_CR1_MSTR	915
5.173.2.2274 SPI_CR1_RXONLY	915
5.173.2.2275 SPI_CR1_SPE	915
5.173.2.2276 SPI_CR1_SSI	915
5.173.2.2277 SPI_CR1_SSM	915
5.173.2.2278 SPI_CR2_ERRIE	916
5.173.2.2279 SPI_CR2_RXDMAEN	916
5.173.2.2280 SPI_CR2_RXNEIE	916
5.173.2.2281 SPI_CR2_SSOE	916
5.173.2.2282 SPI_CR2_TXDMAEN	916
5.173.2.2283 SPI_CR2_TXEIE	916
5.173.2.2284 SPI_CRCPR_CRCPOLY	916
5.173.2.2285 SPI_DR_DR	917
5.173.2.2286 SPI_I2SCFGR_CHLEN	917

5.173.2.2287 SPI_I2SCFGR_CKPOL	917
5.173.2.2288 SPI_I2SCFGR_DATLEN	917
5.173.2.2289 SPI_I2SCFGR_DATLEN_0	917
5.173.2.2290 SPI_I2SCFGR_DATLEN_1	917
5.173.2.2291 SPI_I2SCFGR_I2SCFG	917
5.173.2.2292 SPI_I2SCFGR_I2SCFG_0	918
5.173.2.2293 SPI_I2SCFGR_I2SCFG_1	918
5.173.2.2294 SPI_I2SCFGR_I2SE	918
5.173.2.2295 SPI_I2SCFGR_I2SMOD	918
5.173.2.2296 SPI_I2SCFGR_I2SSTD	918
5.173.2.2297 SPI_I2SCFGR_I2SSTD_0	918
5.173.2.2298 SPI_I2SCFGR_I2SSTD_1	918
5.173.2.2299 SPI_I2SCFGR_PCMSYNC	919
5.173.2.2300 SPI_I2SPR_I2SDIV	919
5.173.2.2301 SPI_I2SPR_MCKOE	919
5.173.2.2302 SPI_I2SPR_ODD	919
5.173.2.2303 SPI_RXCRCR_RXCRC	919
5.173.2.2304 SPI_SR_BSY	919
5.173.2.2305 SPI_SR_CHSIDE	919
5.173.2.2306 SPI_SR_CRCERR	920
5.173.2.2307 SPI_SR_MODF	920
5.173.2.2308 SPI_SR_OVR	920
5.173.2.2309 SPI_SR_RXNE	920
5.173.2.2310 SPI_SR_TXE	920
5.173.2.2311 SPI_SR_UDR	920
5.173.2.2312 SPI_TXCRCR_TXCRC	920
5.173.2.2313 SYSCFG_CMPCR_CMP_PD	921
5.173.2.2314 SYSCFG_CMPCR_READY	921
5.173.2.2315 SYSCFG_EXTICR1_EXTI0	921
5.173.2.2316 SYSCFG_EXTICR1_EXTI0_PA	921
5.173.2.2317 SYSCFG_EXTICR1_EXTI0_PB	921
5.173.2.2318 SYSCFG_EXTICR1_EXTI0_PC	921
5.173.2.2319 SYSCFG_EXTICR1_EXTI0_PD	921
5.173.2.2320 SYSCFG_EXTICR1_EXTI0_PE	922
5.173.2.2321 SYSCFG_EXTICR1_EXTI0_PF	922
5.173.2.2322 SYSCFG_EXTICR1_EXTI0_PG	922
5.173.2.2323 SYSCFG_EXTICR1_EXTI0_PH	922
5.173.2.2324 SYSCFG_EXTICR1_EXTI0_PI	922
5.173.2.2325 SYSCFG_EXTICR1_EXTI0_PJ	922
5.173.2.2326 SYSCFG_EXTICR1_EXTI0_PK	922
5.173.2.2327 SYSCFG_EXTICR1_EXTI1	922
5.173.2.2328 SYSCFG_EXTICR1_EXTI1_PA	923

5.173.2.2329 SYSCFG_EXTICR1_EXTI1_PB	923
5.173.2.2330 SYSCFG_EXTICR1_EXTI1_PC	923
5.173.2.2331 SYSCFG_EXTICR1_EXTI1_PD	923
5.173.2.2332 SYSCFG_EXTICR1_EXTI1_PE	923
5.173.2.2333 SYSCFG_EXTICR1_EXTI1_PF	923
5.173.2.2334 SYSCFG_EXTICR1_EXTI1_PG	923
5.173.2.2335 SYSCFG_EXTICR1_EXTI1_PH	924
5.173.2.2336 SYSCFG_EXTICR1_EXTI1_PI	924
5.173.2.2337 SYSCFG_EXTICR1_EXTI1_PJ	924
5.173.2.2338 SYSCFG_EXTICR1_EXTI1_PK	924
5.173.2.2339 SYSCFG_EXTICR1_EXTI2	924
5.173.2.2340 SYSCFG_EXTICR1_EXTI2_PA	924
5.173.2.2341 SYSCFG_EXTICR1_EXTI2_PB	924
5.173.2.2342 SYSCFG_EXTICR1_EXTI2_PC	925
5.173.2.2343 SYSCFG_EXTICR1_EXTI2_PD	925
5.173.2.2344 SYSCFG_EXTICR1_EXTI2_PE	925
5.173.2.2345 SYSCFG_EXTICR1_EXTI2_PF	925
5.173.2.2346 SYSCFG_EXTICR1_EXTI2_PG	925
5.173.2.2347 SYSCFG_EXTICR1_EXTI2_PH	925
5.173.2.2348 SYSCFG_EXTICR1_EXTI2_PI	925
5.173.2.2349 SYSCFG_EXTICR1_EXTI2_PJ	925
5.173.2.2350 SYSCFG_EXTICR1_EXTI2_PK	926
5.173.2.2351 SYSCFG_EXTICR1_EXTI3	926
5.173.2.2352 SYSCFG_EXTICR1_EXTI3_PA	926
5.173.2.2353 SYSCFG_EXTICR1_EXTI3_PB	926
5.173.2.2354 SYSCFG_EXTICR1_EXTI3_PC	926
5.173.2.2355 SYSCFG_EXTICR1_EXTI3_PD	926
5.173.2.2356 SYSCFG_EXTICR1_EXTI3_PE	926
5.173.2.2357 SYSCFG_EXTICR1_EXTI3_PF	927
5.173.2.2358 SYSCFG_EXTICR1_EXTI3_PG	927
5.173.2.2359 SYSCFG_EXTICR1_EXTI3_PH	927
5.173.2.2360 SYSCFG_EXTICR1_EXTI3_PI	927
5.173.2.2361 SYSCFG_EXTICR1_EXTI3_PJ	927
5.173.2.2362 SYSCFG_EXTICR1_EXTI3_PK	927
5.173.2.2363 SYSCFG_EXTICR2_EXTI4	927
5.173.2.2364 SYSCFG_EXTICR2_EXTI4_PA	928
5.173.2.2365 SYSCFG_EXTICR2_EXTI4_PB	928
5.173.2.2366 SYSCFG_EXTICR2_EXTI4_PC	928
5.173.2.2367 SYSCFG_EXTICR2_EXTI4_PD	928
5.173.2.2368 SYSCFG_EXTICR2_EXTI4_PE	928
5.173.2.2369 SYSCFG_EXTICR2_EXTI4_PF	928
5.173.2.2370 SYSCFG_EXTICR2_EXTI4_PG	928

5.173.2.2371 SYSCFG_EXTICR2_EXTI4_PH	929
5.173.2.2372 SYSCFG_EXTICR2_EXTI4_PI	929
5.173.2.2373 SYSCFG_EXTICR2_EXTI4_PJ	929
5.173.2.2374 SYSCFG_EXTICR2_EXTI4_PK	929
5.173.2.2375 SYSCFG_EXTICR2_EXTI5	929
5.173.2.2376 SYSCFG_EXTICR2_EXTI5_PA	929
5.173.2.2377 SYSCFG_EXTICR2_EXTI5_PB	929
5.173.2.2378 SYSCFG_EXTICR2_EXTI5_PC	930
5.173.2.2379 SYSCFG_EXTICR2_EXTI5_PD	930
5.173.2.2380 SYSCFG_EXTICR2_EXTI5_PE	930
5.173.2.2381 SYSCFG_EXTICR2_EXTI5_PF	930
5.173.2.2382 SYSCFG_EXTICR2_EXTI5_PG	930
5.173.2.2383 SYSCFG_EXTICR2_EXTI5_PH	930
5.173.2.2384 SYSCFG_EXTICR2_EXTI5_PI	930
5.173.2.2385 SYSCFG_EXTICR2_EXTI5_PJ	930
5.173.2.2386 SYSCFG_EXTICR2_EXTI5_PK	931
5.173.2.2387 SYSCFG_EXTICR2_EXTI6	931
5.173.2.2388 SYSCFG_EXTICR2_EXTI6_PA	931
5.173.2.2389 SYSCFG_EXTICR2_EXTI6_PB	931
5.173.2.2390 SYSCFG_EXTICR2_EXTI6_PC	931
5.173.2.2391 SYSCFG_EXTICR2_EXTI6_PD	931
5.173.2.2392 SYSCFG_EXTICR2_EXTI6_PE	931
5.173.2.2393 SYSCFG_EXTICR2_EXTI6_PF	932
5.173.2.2394 SYSCFG_EXTICR2_EXTI6_PG	932
5.173.2.2395 SYSCFG_EXTICR2_EXTI6_PH	932
5.173.2.2396 SYSCFG_EXTICR2_EXTI6_PI	932
5.173.2.2397 SYSCFG_EXTICR2_EXTI6_PJ	932
5.173.2.2398 SYSCFG_EXTICR2_EXTI6_PK	932
5.173.2.2399 SYSCFG_EXTICR2_EXTI7	932
5.173.2.2400 SYSCFG_EXTICR2_EXTI7_PA	933
5.173.2.2401 SYSCFG_EXTICR2_EXTI7_PB	933
5.173.2.2402 SYSCFG_EXTICR2_EXTI7_PC	933
5.173.2.2403 SYSCFG_EXTICR2_EXTI7_PD	933
5.173.2.2404 SYSCFG_EXTICR2_EXTI7_PE	933
5.173.2.2405 SYSCFG_EXTICR2_EXTI7_PF	933
5.173.2.2406 SYSCFG_EXTICR2_EXTI7_PG	933
5.173.2.2407 SYSCFG_EXTICR2_EXTI7_PH	934
5.173.2.2408 SYSCFG_EXTICR2_EXTI7_PI	934
5.173.2.2409 SYSCFG_EXTICR2_EXTI7_PJ	934
5.173.2.2410 SYSCFG_EXTICR2_EXTI7_PK	934
5.173.2.2411 SYSCFG_EXTICR3_EXTI10	934
5.173.2.2412 SYSCFG_EXTICR3_EXTI10_PA	934

5.173.2.2413 SYSCFG_EXTICR3_EXTI10_PB	934
5.173.2.2414 SYSCFG_EXTICR3_EXTI10_PC	935
5.173.2.2415 SYSCFG_EXTICR3_EXTI10_PD	935
5.173.2.2416 SYSCFG_EXTICR3_EXTI10_PE	935
5.173.2.2417 SYSCFG_EXTICR3_EXTI10_PF	935
5.173.2.2418 SYSCFG_EXTICR3_EXTI10_PG	935
5.173.2.2419 SYSCFG_EXTICR3_EXTI10_PH	935
5.173.2.2420 SYSCFG_EXTICR3_EXTI10_PI	935
5.173.2.2421 SYSCFG_EXTICR3_EXTI10_PJ	935
5.173.2.2422 SYSCFG_EXTICR3_EXTI11	936
5.173.2.2423 SYSCFG_EXTICR3_EXTI11_PA	936
5.173.2.2424 SYSCFG_EXTICR3_EXTI11_PB	936
5.173.2.2425 SYSCFG_EXTICR3_EXTI11_PC	936
5.173.2.2426 SYSCFG_EXTICR3_EXTI11_PD	936
5.173.2.2427 SYSCFG_EXTICR3_EXTI11_PE	936
5.173.2.2428 SYSCFG_EXTICR3_EXTI11_PF	936
5.173.2.2429 SYSCFG_EXTICR3_EXTI11_PG	937
5.173.2.2430 SYSCFG_EXTICR3_EXTI11_PH	937
5.173.2.2431 SYSCFG_EXTICR3_EXTI11_PI	937
5.173.2.2432 SYSCFG_EXTICR3_EXTI11_PJ	937
5.173.2.2433 SYSCFG_EXTICR3_EXTI8	937
5.173.2.2434 SYSCFG_EXTICR3_EXTI8_PA	937
5.173.2.2435 SYSCFG_EXTICR3_EXTI8_PB	937
5.173.2.2436 SYSCFG_EXTICR3_EXTI8_PC	938
5.173.2.2437 SYSCFG_EXTICR3_EXTI8_PD	938
5.173.2.2438 SYSCFG_EXTICR3_EXTI8_PE	938
5.173.2.2439 SYSCFG_EXTICR3_EXTI8_PF	938
5.173.2.2440 SYSCFG_EXTICR3_EXTI8_PG	938
5.173.2.2441 SYSCFG_EXTICR3_EXTI8_PH	938
5.173.2.2442 SYSCFG_EXTICR3_EXTI8_PI	938
5.173.2.2443 SYSCFG_EXTICR3_EXTI8_PJ	938
5.173.2.2444 SYSCFG_EXTICR3_EXTI9	939
5.173.2.2445 SYSCFG_EXTICR3_EXTI9_PA	939
5.173.2.2446 SYSCFG_EXTICR3_EXTI9_PB	939
5.173.2.2447 SYSCFG_EXTICR3_EXTI9_PC	939
5.173.2.2448 SYSCFG_EXTICR3_EXTI9_PD	939
5.173.2.2449 SYSCFG_EXTICR3_EXTI9_PE	939
5.173.2.2450 SYSCFG_EXTICR3_EXTI9_PF	939
5.173.2.2451 SYSCFG_EXTICR3_EXTI9_PG	940
5.173.2.2452 SYSCFG_EXTICR3_EXTI9_PH	940
5.173.2.2453 SYSCFG_EXTICR3_EXTI9_PI	940
5.173.2.2454 SYSCFG_EXTICR3_EXTI9_PJ	940

5.173.2.2455 SYSCFG_EXTICR4_EXTI12	940
5.173.2.2456 SYSCFG_EXTICR4_EXTI12_PA	940
5.173.2.2457 SYSCFG_EXTICR4_EXTI12_PB	940
5.173.2.2458 SYSCFG_EXTICR4_EXTI12_PC	941
5.173.2.2459 SYSCFG_EXTICR4_EXTI12_PD	941
5.173.2.2460 SYSCFG_EXTICR4_EXTI12_PE	941
5.173.2.2461 SYSCFG_EXTICR4_EXTI12_PF	941
5.173.2.2462 SYSCFG_EXTICR4_EXTI12_PG	941
5.173.2.2463 SYSCFG_EXTICR4_EXTI12_PH	941
5.173.2.2464 SYSCFG_EXTICR4_EXTI12_PI	941
5.173.2.2465 SYSCFG_EXTICR4_EXTI12_PJ	941
5.173.2.2466 SYSCFG_EXTICR4_EXTI13	942
5.173.2.2467 SYSCFG_EXTICR4_EXTI13_PA	942
5.173.2.2468 SYSCFG_EXTICR4_EXTI13_PB	942
5.173.2.2469 SYSCFG_EXTICR4_EXTI13_PC	942
5.173.2.2470 SYSCFG_EXTICR4_EXTI13_PD	942
5.173.2.2471 SYSCFG_EXTICR4_EXTI13_PE	942
5.173.2.2472 SYSCFG_EXTICR4_EXTI13_PF	942
5.173.2.2473 SYSCFG_EXTICR4_EXTI13_PG	943
5.173.2.2474 SYSCFG_EXTICR4_EXTI13_PH	943
5.173.2.2475 SYSCFG_EXTICR4_EXTI13_PI	943
5.173.2.2476 SYSCFG_EXTICR4_EXTI13_PJ	943
5.173.2.2477 SYSCFG_EXTICR4_EXTI14	943
5.173.2.2478 SYSCFG_EXTICR4_EXTI14_PA	943
5.173.2.2479 SYSCFG_EXTICR4_EXTI14_PB	943
5.173.2.2480 SYSCFG_EXTICR4_EXTI14_PC	944
5.173.2.2481 SYSCFG_EXTICR4_EXTI14_PD	944
5.173.2.2482 SYSCFG_EXTICR4_EXTI14_PE	944
5.173.2.2483 SYSCFG_EXTICR4_EXTI14_PF	944
5.173.2.2484 SYSCFG_EXTICR4_EXTI14_PG	944
5.173.2.2485 SYSCFG_EXTICR4_EXTI14_PH	944
5.173.2.2486 SYSCFG_EXTICR4_EXTI14_PI	944
5.173.2.2487 SYSCFG_EXTICR4_EXTI14_PJ	944
5.173.2.2488 SYSCFG_EXTICR4_EXTI15	945
5.173.2.2489 SYSCFG_EXTICR4_EXTI15_PA	945
5.173.2.2490 SYSCFG_EXTICR4_EXTI15_PB	945
5.173.2.2491 SYSCFG_EXTICR4_EXTI15_PC	945
5.173.2.2492 SYSCFG_EXTICR4_EXTI15_PD	945
5.173.2.2493 SYSCFG_EXTICR4_EXTI15_PE	945
5.173.2.2494 SYSCFG_EXTICR4_EXTI15_PF	945
5.173.2.2495 SYSCFG_EXTICR4_EXTI15_PG	946
5.173.2.2496 SYSCFG_EXTICR4_EXTI15_PH	946

5.173.2.2497 SYSCFG_EXTICR4_EXTI15_PI	946
5.173.2.2498 SYSCFG_EXTICR4_EXTI15_PJ	946
5.173.2.2499 SYSCFG_MEMRMP_FB_MODE	946
5.173.2.2500 SYSCFG_MEMRMP_MEM_MODE	946
5.173.2.2501 SYSCFG_MEMRMP_MEM_MODE_0	946
5.173.2.2502 SYSCFG_MEMRMP_MEM_MODE_1	946
5.173.2.2503 SYSCFG_MEMRMP_MEM_MODE_2	947
5.173.2.2504 SYSCFG_MEMRMP_SWP_FMC	947
5.173.2.2505 SYSCFG_MEMRMP_SWP_FMC_0	947
5.173.2.2506 SYSCFG_MEMRMP_SWP_FMC_1	947
5.173.2.2507 SYSCFG_PMC_ADC1DC2	947
5.173.2.2508 SYSCFG_PMC_ADC2DC2	947
5.173.2.2509 SYSCFG_PMC_ADC3DC2	947
5.173.2.2510 SYSCFG_PMC_ADCxDC2	948
5.173.2.2511 SYSCFG_PMC_MII_RMII_SEL	948
5.173.2.2512 TIM_ARR_ARR	948
5.173.2.2513 TIM_BDTR_AOE	948
5.173.2.2514 TIM_BDTR_BKE	948
5.173.2.2515 TIM_BDTR_BKP	948
5.173.2.2516 TIM_BDTR_DTG	948
5.173.2.2517 TIM_BDTR_DTG_0	949
5.173.2.2518 TIM_BDTR_DTG_1	949
5.173.2.2519 TIM_BDTR_DTG_2	949
5.173.2.2520 TIM_BDTR_DTG_3	949
5.173.2.2521 TIM_BDTR_DTG_4	949
5.173.2.2522 TIM_BDTR_DTG_5	949
5.173.2.2523 TIM_BDTR_DTG_6	949
5.173.2.2524 TIM_BDTR_DTG_7	949
5.173.2.2525 TIM_BDTR_LOCK	950
5.173.2.2526 TIM_BDTR_LOCK_0	950
5.173.2.2527 TIM_BDTR_LOCK_1	950
5.173.2.2528 TIM_BDTR_MOE	950
5.173.2.2529 TIM_BDTR_OSSI	950
5.173.2.2530 TIM_BDTR_OSSR	950
5.173.2.2531 TIM_CCER_CC1E	950
5.173.2.2532 TIM_CCER_CC1NE	951
5.173.2.2533 TIM_CCER_CC1NP	951
5.173.2.2534 TIM_CCER_CC1P	951
5.173.2.2535 TIM_CCER_CC2E	951
5.173.2.2536 TIM_CCER_CC2NE	951
5.173.2.2537 TIM_CCER_CC2NP	951
5.173.2.2538 TIM_CCER_CC2P	951

5.173.2.2539 TIM_CCER_CC3E	952
5.173.2.2540 TIM_CCER_CC3NE	952
5.173.2.2541 TIM_CCER_CC3NP	952
5.173.2.2542 TIM_CCER_CC3P	952
5.173.2.2543 TIM_CCER_CC4E	952
5.173.2.2544 TIM_CCER_CC4NP	952
5.173.2.2545 TIM_CCER_CC4P	952
5.173.2.2546 TIM_CCMR1_CC1S	953
5.173.2.2547 TIM_CCMR1_CC1S_0	953
5.173.2.2548 TIM_CCMR1_CC1S_1	953
5.173.2.2549 TIM_CCMR1_CC2S	953
5.173.2.2550 TIM_CCMR1_CC2S_0	953
5.173.2.2551 TIM_CCMR1_CC2S_1	953
5.173.2.2552 TIM_CCMR1_IC1F	953
5.173.2.2553 TIM_CCMR1_IC1F_0	954
5.173.2.2554 TIM_CCMR1_IC1F_1	954
5.173.2.2555 TIM_CCMR1_IC1F_2	954
5.173.2.2556 TIM_CCMR1_IC1F_3	954
5.173.2.2557 TIM_CCMR1_IC1PSC	954
5.173.2.2558 TIM_CCMR1_IC1PSC_0	954
5.173.2.2559 TIM_CCMR1_IC1PSC_1	954
5.173.2.2560 TIM_CCMR1_IC2F	954
5.173.2.2561 TIM_CCMR1_IC2F_0	955
5.173.2.2562 TIM_CCMR1_IC2F_1	955
5.173.2.2563 TIM_CCMR1_IC2F_2	955
5.173.2.2564 TIM_CCMR1_IC2F_3	955
5.173.2.2565 TIM_CCMR1_IC2PSC	955
5.173.2.2566 TIM_CCMR1_IC2PSC_0	955
5.173.2.2567 TIM_CCMR1_IC2PSC_1	955
5.173.2.2568 TIM_CCMR1_OC1CE	956
5.173.2.2569 TIM_CCMR1_OC1FE	956
5.173.2.2570 TIM_CCMR1_OC1M	956
5.173.2.2571 TIM_CCMR1_OC1M_0	956
5.173.2.2572 TIM_CCMR1_OC1M_1	956
5.173.2.2573 TIM_CCMR1_OC1M_2	956
5.173.2.2574 TIM_CCMR1_OC1PE	956
5.173.2.2575 TIM_CCMR1_OC2CE	957
5.173.2.2576 TIM_CCMR1_OC2FE	957
5.173.2.2577 TIM_CCMR1_OC2M	957
5.173.2.2578 TIM_CCMR1_OC2M_0	957
5.173.2.2579 TIM_CCMR1_OC2M_1	957
5.173.2.2580 TIM_CCMR1_OC2M_2	957

5.173.2.2581 TIM_CCMR1_OC2PE	957
5.173.2.2582 TIM_CCMR2_CC3S	958
5.173.2.2583 TIM_CCMR2_CC3S_0	958
5.173.2.2584 TIM_CCMR2_CC3S_1	958
5.173.2.2585 TIM_CCMR2_CC4S	958
5.173.2.2586 TIM_CCMR2_CC4S_0	958
5.173.2.2587 TIM_CCMR2_CC4S_1	958
5.173.2.2588 TIM_CCMR2_IC3F	958
5.173.2.2589 TIM_CCMR2_IC3F_0	959
5.173.2.2590 TIM_CCMR2_IC3F_1	959
5.173.2.2591 TIM_CCMR2_IC3F_2	959
5.173.2.2592 TIM_CCMR2_IC3F_3	959
5.173.2.2593 TIM_CCMR2_IC3PSC	959
5.173.2.2594 TIM_CCMR2_IC3PSC_0	959
5.173.2.2595 TIM_CCMR2_IC3PSC_1	959
5.173.2.2596 TIM_CCMR2_IC4F	959
5.173.2.2597 TIM_CCMR2_IC4F_0	960
5.173.2.2598 TIM_CCMR2_IC4F_1	960
5.173.2.2599 TIM_CCMR2_IC4F_2	960
5.173.2.2600 TIM_CCMR2_IC4F_3	960
5.173.2.2601 TIM_CCMR2_IC4PSC	960
5.173.2.2602 TIM_CCMR2_IC4PSC_0	960
5.173.2.2603 TIM_CCMR2_IC4PSC_1	960
5.173.2.2604 TIM_CCMR2_OC3CE	960
5.173.2.2605 TIM_CCMR2_OC3FE	961
5.173.2.2606 TIM_CCMR2_OC3M	961
5.173.2.2607 TIM_CCMR2_OC3M_0	961
5.173.2.2608 TIM_CCMR2_OC3M_1	961
5.173.2.2609 TIM_CCMR2_OC3M_2	961
5.173.2.2610 TIM_CCMR2_OC3PE	961
5.173.2.2611 TIM_CCMR2_OC4CE	961
5.173.2.2612 TIM_CCMR2_OC4FE	962
5.173.2.2613 TIM_CCMR2_OC4M	962
5.173.2.2614 TIM_CCMR2_OC4M_0	962
5.173.2.2615 TIM_CCMR2_OC4M_1	962
5.173.2.2616 TIM_CCMR2_OC4M_2	962
5.173.2.2617 TIM_CCMR2_OC4PE	962
5.173.2.2618 TIM_CCR1_CCR1	962
5.173.2.2619 TIM_CCR2_CCR2	963
5.173.2.2620 TIM_CCR3_CCR3	963
5.173.2.2621 TIM_CCR4_CCR4	963
5.173.2.2622 TIM_CNT_CNT	963

5.173.2.2623 TIM_CR1_ARPE	963
5.173.2.2624 TIM_CR1_CEN	963
5.173.2.2625 TIM_CR1_CKD	963
5.173.2.2626 TIM_CR1_CKD_0	964
5.173.2.2627 TIM_CR1_CKD_1	964
5.173.2.2628 TIM_CR1_CMS	964
5.173.2.2629 TIM_CR1_CMS_0	964
5.173.2.2630 TIM_CR1_CMS_1	964
5.173.2.2631 TIM_CR1_DIR	964
5.173.2.2632 TIM_CR1_OPM	964
5.173.2.2633 TIM_CR1_UDIS	965
5.173.2.2634 TIM_CR1_URS	965
5.173.2.2635 TIM_CR2_CCDS	965
5.173.2.2636 TIM_CR2_CCPC	965
5.173.2.2637 TIM_CR2_CCUS	965
5.173.2.2638 TIM_CR2_MMS	965
5.173.2.2639 TIM_CR2_MMS_0	965
5.173.2.2640 TIM_CR2_MMS_1	966
5.173.2.2641 TIM_CR2_MMS_2	966
5.173.2.2642 TIM_CR2_OIS1	966
5.173.2.2643 TIM_CR2_OIS1N	966
5.173.2.2644 TIM_CR2_OIS2	966
5.173.2.2645 TIM_CR2_OIS2N	966
5.173.2.2646 TIM_CR2_OIS3	966
5.173.2.2647 TIM_CR2_OIS3N	967
5.173.2.2648 TIM_CR2_OIS4	967
5.173.2.2649 TIM_CR2_TI1S	967
5.173.2.2650 TIM_DCR_DBAA	967
5.173.2.2651 TIM_DCR_DBAA_0	967
5.173.2.2652 TIM_DCR_DBAA_1	967
5.173.2.2653 TIM_DCR_DBAA_2	967
5.173.2.2654 TIM_DCR_DBAA_3	968
5.173.2.2655 TIM_DCR_DBAA_4	968
5.173.2.2656 TIM_DCR_DBBL	968
5.173.2.2657 TIM_DCR_DBBL_0	968
5.173.2.2658 TIM_DCR_DBBL_1	968
5.173.2.2659 TIM_DCR_DBBL_2	968
5.173.2.2660 TIM_DCR_DBBL_3	968
5.173.2.2661 TIM_DCR_DBBL_4	968
5.173.2.2662 TIM_DIER_BIE	969
5.173.2.2663 TIM_DIER_CC1DE	969
5.173.2.2664 TIM_DIER_CC1IE	969

5.173.2.2665 TIM_DIER_CC2DE	969
5.173.2.2666 TIM_DIER_CC2IE	969
5.173.2.2667 TIM_DIER_CC3DE	969
5.173.2.2668 TIM_DIER_CC3IE	969
5.173.2.2669 TIM_DIER_CC4DE	970
5.173.2.2670 TIM_DIER_CC4IE	970
5.173.2.2671 TIM_DIER_COMDE	970
5.173.2.2672 TIM_DIER_COMIE	970
5.173.2.2673 TIM_DIER_TDE	970
5.173.2.2674 TIM_DIER_TIE	970
5.173.2.2675 TIM_DIER_UDE	970
5.173.2.2676 TIM_DIER_UIE	971
5.173.2.2677 TIM_DMAR_DMAB	971
5.173.2.2678 TIM_EGR_BG	971
5.173.2.2679 TIM_EGR_CC1G	971
5.173.2.2680 TIM_EGR_CC2G	971
5.173.2.2681 TIM_EGR_CC3G	971
5.173.2.2682 TIM_EGR_CC4G	971
5.173.2.2683 TIM_EGR_COMG	972
5.173.2.2684 TIM_EGR_TG	972
5.173.2.2685 TIM_EGR_UG	972
5.173.2.2686 TIM_OR_ITR1_RMP	972
5.173.2.2687 TIM_OR_ITR1_RMP_0	972
5.173.2.2688 TIM_OR_ITR1_RMP_1	972
5.173.2.2689 TIM_OR_TI4_RMP	972
5.173.2.2690 TIM_OR_TI4_RMP_0	973
5.173.2.2691 TIM_OR_TI4_RMP_1	973
5.173.2.2692 TIM_PSC_PSC	973
5.173.2.2693 TIM_RCR_REP	973
5.173.2.2694 TIM_SMCR_ECE	973
5.173.2.2695 TIM_SMCR ETF	973
5.173.2.2696 TIM_SMCR ETF_0	973
5.173.2.2697 TIM_SMCR ETF_1	974
5.173.2.2698 TIM_SMCR ETF_2	974
5.173.2.2699 TIM_SMCR ETF_3	974
5.173.2.2700 TIM_SMCR ETP	974
5.173.2.2701 TIM_SMCR ETPS	974
5.173.2.2702 TIM_SMCR ETPS_0	974
5.173.2.2703 TIM_SMCR ETPS_1	974
5.173.2.2704 TIM_SMCR MSM	974
5.173.2.2705 TIM_SMCR SMS	975
5.173.2.2706 TIM_SMCR SMS_0	975

5.173.2.2707 TIM_SMCR_SMS_1	975
5.173.2.2708 TIM_SMCR_SMS_2	975
5.173.2.2709 TIM_SMCR_TS	975
5.173.2.2710 TIM_SMCR_TS_0	975
5.173.2.2711 TIM_SMCR_TS_1	975
5.173.2.2712 TIM_SMCR_TS_2	976
5.173.2.2713 TIM_SR_BIF	976
5.173.2.2714 TIM_SR_CC1IF	976
5.173.2.2715 TIM_SR_CC1OF	976
5.173.2.2716 TIM_SR_CC2IF	976
5.173.2.2717 TIM_SR_CC2OF	976
5.173.2.2718 TIM_SR_CC3IF	976
5.173.2.2719 TIM_SR_CC3OF	977
5.173.2.2720 TIM_SR_CC4IF	977
5.173.2.2721 TIM_SR_CC4OF	977
5.173.2.2722 TIM_SR_COMIF	977
5.173.2.2723 TIM_SR_TIF	977
5.173.2.2724 TIM_SR_UIF	977
5.173.2.2725 USART_BRR_DIV_Fraction	977
5.173.2.2726 USART_BRR_DIV_Mantissa	978
5.173.2.2727 USART_CR1_IDLEIE	978
5.173.2.2728 USART_CR1_M	978
5.173.2.2729 USART_CR1_OVER8	978
5.173.2.2730 USART_CR1_PCE	978
5.173.2.2731 USART_CR1_PEIE	978
5.173.2.2732 USART_CR1_PS	978
5.173.2.2733 USART_CR1_RE	979
5.173.2.2734 USART_CR1_RWU	979
5.173.2.2735 USART_CR1_RXNEIE	979
5.173.2.2736 USART_CR1_SBK	979
5.173.2.2737 USART_CR1_TCIE	979
5.173.2.2738 USART_CR1_TE	979
5.173.2.2739 USART_CR1_TXEIE	979
5.173.2.2740 USART_CR1 UE	980
5.173.2.2741 USART_CR1_WAKE	980
5.173.2.2742 USART_CR2_ADD	980
5.173.2.2743 USART_CR2_CLKEN	980
5.173.2.2744 USART_CR2_CPHA	980
5.173.2.2745 USART_CR2_CPOL	980
5.173.2.2746 USART_CR2_LBCL	980
5.173.2.2747 USART_CR2_LBDIE	981
5.173.2.2748 USART_CR2_LBDL	981

5.173.2.2749 USART_CR2_LINEN	981
5.173.2.2750 USART_CR2_STOP	981
5.173.2.2751 USART_CR2_STOP_0	981
5.173.2.2752 USART_CR2_STOP_1	981
5.173.2.2753 USART_CR3_CTSE	981
5.173.2.2754 USART_CR3_CTSIE	982
5.173.2.2755 USART_CR3_DMAR	982
5.173.2.2756 USART_CR3_DMAT	982
5.173.2.2757 USART_CR3_EIE	982
5.173.2.2758 USART_CR3_HDSEL	982
5.173.2.2759 USART_CR3_IREN	982
5.173.2.2760 USART_CR3_IRLP	982
5.173.2.2761 USART_CR3_NACK	983
5.173.2.2762 USART_CR3_ONEBIT	983
5.173.2.2763 USART_CR3_RTSE	983
5.173.2.2764 USART_CR3_SCEN	983
5.173.2.2765 USART_DR_DR	983
5.173.2.2766 USART_GTPR_GT	983
5.173.2.2767 USART_GTPR_PSC	983
5.173.2.2768 USART_GTPR_PSC_0	984
5.173.2.2769 USART_GTPR_PSC_1	984
5.173.2.2770 USART_GTPR_PSC_2	984
5.173.2.2771 USART_GTPR_PSC_3	984
5.173.2.2772 USART_GTPR_PSC_4	984
5.173.2.2773 USART_GTPR_PSC_5	984
5.173.2.2774 USART_GTPR_PSC_6	984
5.173.2.2775 USART_GTPR_PSC_7	984
5.173.2.2776 USART_SR_CTS	985
5.173.2.2777 USART_SR_FE	985
5.173.2.2778 USART_SR_IDLE	985
5.173.2.2779 USART_SR_LBD	985
5.173.2.2780 USART_SR_NE	985
5.173.2.2781 USART_SR_ORE	985
5.173.2.2782 USART_SR_PE	985
5.173.2.2783 USART_SR_RXNE	986
5.173.2.2784 USART_SR_TC	986
5.173.2.2785 USART_SR_TXE	986
5.173.2.2786 WWDG_CFR_EWI	986
5.173.2.2787 WWDG_CFR_W	986
5.173.2.2788 WWDG_CFR_W_0	986
5.173.2.2789 WWDG_CFR_W_1	986
5.173.2.2790 WWDG_CFR_W_2	987

5.173.2.2791 WWDG_CFR_W_3	987
5.173.2.2792 WWDG_CFR_W_4	987
5.173.2.2793 WWDG_CFR_W_5	987
5.173.2.2794 WWDG_CFR_W_6	987
5.173.2.2795 WWDG_CFR_WDGTB	987
5.173.2.2796 WWDG_CFR_WDGTB_0	987
5.173.2.2797 WWDG_CFR_WDGTB_1	987
5.173.2.2798 WWDG_CR_T	988
5.173.2.2799 WWDG_CR_T_0	988
5.173.2.2800 WWDG_CR_T_1	988
5.173.2.2801 WWDG_CR_T_2	988
5.173.2.2802 WWDG_CR_T_3	988
5.173.2.2803 WWDG_CR_T_4	988
5.173.2.2804 WWDG_CR_T_5	988
5.173.2.2805 WWDG_CR_T_6	988
5.173.2.2806 WWDG_CR_WDGA	989
5.173.2.2807 WWDG_SR_EWIF	989
5.174 Exported_macro	989
5.174.1 Detailed Description	990
5.175 Stm32f4xx_system	990
5.175.1 Detailed Description	991
5.176 STM32F4xx_System_Includes	991
5.177 STM32F4xx_System_Exported_types	991
5.177.1 Detailed Description	992
5.177.2 Variable Documentation	992
5.177.2.1 SystemCoreClock	992
5.178 STM32F4xx_System_Exported_Constants	992
5.179 STM32F4xx_System_Exported_Macros	992
5.180 STM32F4xx_System_Exported_Functions	992
5.180.1 Detailed Description	993
5.180.2 Function Documentation	993
5.180.2.1 SystemCoreClockUpdate()	993
5.180.2.2 SystemInit()	994
5.181 STM32F4xx_System_Private_Includes	994
5.182 STM32F4xx_System_Private_TypesDefinitions	994
5.183 STM32F4xx_System_Private_Defines	995
5.183.1 Detailed Description	995
5.183.2 Macro Definition Documentation	995
5.183.2.1 VECT_TAB_OFFSET	995
5.184 STM32F4xx_System_Private_Macros	995
5.185 STM32F4xx_System_Private_Variables	996
5.185.1 Detailed Description	996

5.186 STM32F4xx_System_Private_FunctionPrototypes	996
5.186.1 Detailed Description	996
5.187 STM32F4xx_System_Private_Functions	996
5.187.1 Detailed Description	997
5.187.2 Function Documentation	997
5.187.2.1 SystemCoreClockUpdate()	997
5.187.2.2 SystemInit()	998
6 Class Documentation	999
6.1 ADC_Common_TypeDef Struct Reference	999
6.2 ADC_TypeDef Struct Reference	999
6.2.1 Detailed Description	1000
6.3 APSR_Type Union Reference	1000
6.3.1 Detailed Description	1000
6.4 CAN_FIFOMailBox_TypeDef Struct Reference	1000
6.4.1 Detailed Description	1001
6.5 CAN_FilterRegister_TypeDef Struct Reference	1001
6.5.1 Detailed Description	1001
6.6 CAN_TxMailBox_TypeDef Struct Reference	1001
6.6.1 Detailed Description	1002
6.7 CAN_TypeDef Struct Reference	1002
6.7.1 Detailed Description	1003
6.8 CONTROL_Type Union Reference	1003
6.8.1 Detailed Description	1003
6.9 CoreDebug_TypeDef Struct Reference	1003
6.9.1 Detailed Description	1004
6.10 CRC_TypeDef Struct Reference	1004
6.10.1 Detailed Description	1004
6.10.2 Member Data Documentation	1004
6.10.2.1 CR	1004
6.10.2.2 DR	1005
6.10.2.3 IDR	1005
6.10.2.4 INIT	1005
6.10.2.5 POL	1005
6.10.2.6 RESERVED	1005
6.10.2.7 SR	1005
6.11 CRYP_TypeDef Struct Reference	1005
6.11.1 Detailed Description	1006
6.12 DAC_TypeDef Struct Reference	1006
6.12.1 Detailed Description	1007
6.13 DBGMCU_TypeDef Struct Reference	1007
6.13.1 Detailed Description	1007

6.14 DCMI_TypeDef Struct Reference	1008
6.14.1 Detailed Description	1008
6.15 DMA2D_TypeDef Struct Reference	1008
6.15.1 Detailed Description	1009
6.16 DMA_Stream_TypeDef Struct Reference	1009
6.16.1 Detailed Description	1009
6.17 DMA_TypeDef Struct Reference	1010
6.18 DWT_Type Struct Reference	1010
6.18.1 Detailed Description	1011
6.19 ETH_TypeDef Struct Reference	1011
6.19.1 Detailed Description	1012
6.20 EXTI_TypeDef Struct Reference	1012
6.20.1 Detailed Description	1013
6.20.2 Member Data Documentation	1013
6.20.2.1 EMR	1013
6.20.2.2 FTSR	1013
6.20.2.3 IMR	1013
6.20.2.4 PR	1013
6.20.2.5 RTSR	1014
6.20.2.6 SWIER	1014
6.21 FLASH_TypeDef Struct Reference	1014
6.21.1 Detailed Description	1014
6.22 GPIO_PinConfig_t Struct Reference	1014
6.22.1 Detailed Description	1015
6.22.2 Member Data Documentation	1015
6.22.2.1 GPIO_MODE	1015
6.22.2.2 GPIO_PinAltFunMode	1015
6.22.2.3 GPIO_PinNumber	1015
6.22.2.4 GPIO_PinOType	1015
6.22.2.5 GPIO_PinPuPdControl	1016
6.22.2.6 GPIO_PinSpeed	1016
6.23 GPIO_TypeDef Struct Reference	1016
6.23.1 Detailed Description	1016
6.23.2 Member Data Documentation	1017
6.23.2.1 AFR	1017
6.23.2.2 BSRR	1017
6.23.2.3 IDR	1017
6.23.2.4 LCKR	1017
6.23.2.5 MODER	1017
6.23.2.6 ODR	1017
6.23.2.7 OSPEEDR	1017
6.23.2.8 OTYPER	1018

6.23.2.9 PUPDR	1018
6.24 HASH_DIGEST_TypeDef Struct Reference	1018
6.24.1 Detailed Description	1018
6.25 HASH_TypeDef Struct Reference	1018
6.25.1 Detailed Description	1019
6.26 I2C_TypeDef Struct Reference	1019
6.26.1 Detailed Description	1020
6.26.2 Member Data Documentation	1020
6.26.2.1 CCR	1020
6.26.2.2 CR1	1020
6.26.2.3 CR2	1020
6.26.2.4 DR	1020
6.26.2.5 OAR1	1020
6.26.2.6 OAR2	1020
6.26.2.7 SR1	1021
6.26.2.8 SR2	1021
6.26.2.9 TRISE	1021
6.27 IPSR_Type Union Reference	1021
6.27.1 Detailed Description	1021
6.28 ITM_TypeDef Struct Reference	1022
6.28.1 Detailed Description	1022
6.29 IWDG_TypeDef Struct Reference	1023
6.29.1 Detailed Description	1023
6.30 LCD_t Struct Reference	1023
6.30.1 Detailed Description	1024
6.30.2 Member Data Documentation	1024
6.30.2.1 D0_PIN	1024
6.30.2.2 D1_PIN	1024
6.30.2.3 D2_PIN	1024
6.30.2.4 D3_PIN	1025
6.30.2.5 D4_PIN	1025
6.30.2.6 D5_PIN	1025
6.30.2.7 D6_PIN	1025
6.30.2.8 D7_PIN	1025
6.30.2.9 Display_Mode	1025
6.30.2.10 EN_PIN	1025
6.30.2.11 Entry_Mode	1025
6.30.2.12 GPIO_PORT	1026
6.30.2.13 Mode	1026
6.30.2.14 Rows	1026
6.30.2.15 RS_PIN	1026
6.31 LED_cfg_t Struct Reference	1026

6.31.1 Member Data Documentation	1027
6.31.1.1 LED_Mode	1027
6.31.1.2 LED_Pin	1027
6.31.1.3 LED_Port	1027
6.32 LTDC_Layer_TypeDef Struct Reference	1027
6.32.1 Detailed Description	1028
6.33 LTDC_TypeDef Struct Reference	1028
6.33.1 Detailed Description	1029
6.34 NVIC_Type Struct Reference	1029
6.34.1 Detailed Description	1029
6.35 NVIC_TypeDef Struct Reference	1029
6.35.1 Member Data Documentation	1030
6.35.1.1 IABR	1030
6.35.1.2 ICER	1030
6.35.1.3 ICPR	1030
6.35.1.4 IP	1030
6.35.1.5 ISER	1030
6.35.1.6 ISPR	1030
6.35.1.7 RESERVED0	1030
6.35.1.8 RESERVED1	1031
6.35.1.9 RESERVED2	1031
6.35.1.10 RESERVED3	1031
6.35.1.11 RESERVED4	1031
6.35.1.12 RESERVED5	1031
6.35.1.13 STIR	1031
6.36 PWR_TypeDef Struct Reference	1031
6.36.1 Detailed Description	1032
6.37 RCC_TypeDef Struct Reference	1032
6.37.1 Detailed Description	1033
6.37.2 Member Data Documentation	1033
6.37.2.1 AHB1ENR	1033
6.37.2.2 AHB1LPENR	1034
6.37.2.3 AHB1RSTR	1034
6.37.2.4 AHB2ENR	1034
6.37.2.5 AHB2LPENR	1034
6.37.2.6 AHB2RSTR	1034
6.37.2.7 APB1ENR	1034
6.37.2.8 APB1LPENR	1034
6.37.2.9 APB1RSTR	1034
6.37.2.10 APB2ENR	1035
6.37.2.11 APB2LPENR	1035
6.37.2.12 APB2RSTR	1035

6.37.2.13 BDCR	1035
6.37.2.14 CFGR	1035
6.37.2.15 CIR	1035
6.37.2.16 CR	1035
6.37.2.17 CSR	1035
6.37.2.18 DCKCFGREG	1036
6.37.2.19 PLLCFGREG	1036
6.37.2.20 PLLI2SCFGREG	1036
6.37.2.21 RESERVED0	1036
6.37.2.22 RESERVED1	1036
6.37.2.23 RESERVED2	1036
6.37.2.24 RESERVED3	1036
6.37.2.25 RESERVED4	1036
6.37.2.26 RESERVED5	1037
6.37.2.27 RESERVED6	1037
6.37.2.28 RESERVED7	1037
6.37.2.29 SSCGR	1037
6.38 RNG_TypeDef Struct Reference	1037
6.38.1 Detailed Description	1037
6.39 RTC_TypeDef Struct Reference	1038
6.39.1 Detailed Description	1039
6.40 S_I2C_Config_t Struct Reference	1039
6.40.1 Detailed Description	1039
6.40.2 Member Data Documentation	1039
6.40.2.1 General_Call_Address_Detection	1040
6.40.2.2 I2C_Ack_Conrol	1040
6.40.2.3 I2C_Mode	1040
6.40.2.4 I2C_Slave_address	1040
6.40.2.5 I2C_Speed	1040
6.40.2.6 I2C_stretchmode	1040
6.40.2.7 P_Slave_Event_CallBack	1040
6.41 S_I2C_Slave_address Struct Reference	1041
6.41.1 Detailed Description	1041
6.41.2 Member Data Documentation	1041
6.41.2.1 Enable_Dual_Address	1041
6.41.2.2 I2C_Addressing_Mode	1041
6.41.2.3 PrimaryAddress	1041
6.41.2.4 SecondaryAddress	1041
6.42 S_IRQ_SRC Struct Reference	1042
6.42.1 Detailed Description	1042
6.42.2 Member Data Documentation	1042
6.42.2.1 ERRI	1042

6.42.2.2 Reserved	1042
6.42.2.3 RXE	1042
6.42.2.4 TXE	1043
6.43 S_SPI_Config_t Struct Reference	1043
6.43.1 Detailed Description	1043
6.43.2 Member Data Documentation	1043
6.43.2.1 Clock_Phase	1043
6.43.2.2 Clock_Polarity	1044
6.43.2.3 Communication_Mode	1044
6.43.2.4 Device_Mode	1044
6.43.2.5 Frame_Format	1044
6.43.2.6 IRQ_Enable	1044
6.43.2.7 NSS	1044
6.43.2.8 P_IRQ_CallBack	1044
6.43.2.9 Payload_Length	1044
6.43.2.10 SPI_Prescaler	1045
6.44 SAI_Block_TypeDef Struct Reference	1045
6.45 SAI_TypeDef Struct Reference	1045
6.45.1 Detailed Description	1045
6.46 SCB_Type Struct Reference	1046
6.46.1 Detailed Description	1046
6.47 SCB_TypeDef Struct Reference	1046
6.47.1 Member Data Documentation	1047
6.47.1.1 AIRCR	1047
6.47.1.2 BFAR	1047
6.47.1.3 CCR	1047
6.47.1.4 CFSR	1047
6.47.1.5 CPUID	1047
6.47.1.6 HFSR	1047
6.47.1.7 ICSR	1047
6.47.1.8 MMAR	1048
6.47.1.9 RESERVED	1048
6.47.1.10 SCR	1048
6.47.1.11 SHCSR	1048
6.47.1.12 SHP	1048
6.47.1.13 VTOR	1048
6.48 SCnSCB_Type Struct Reference	1048
6.48.1 Detailed Description	1049
6.49 SDIO_TypeDef Struct Reference	1049
6.49.1 Detailed Description	1049
6.50 SPI_TypeDef Struct Reference	1050
6.50.1 Detailed Description	1050

6.50.2 Member Data Documentation	1050
6.50.2.1 CR1	1050
6.50.2.2 CR2	1051
6.50.2.3 CRCPR	1051
6.50.2.4 DR	1051
6.50.2.5 I2SCFGR	1051
6.50.2.6 I2SPR	1051
6.50.2.7 RXCRCR	1051
6.50.2.8 SR	1051
6.50.2.9 TXCRCR	1052
6.51 STK_config_t Struct Reference	1052
6.51.1 Detailed Description	1052
6.51.2 Member Data Documentation	1052
6.51.2.1 Callback_Function	1052
6.51.2.2 clock_config	1052
6.51.2.3 interrupt_config	1053
6.51.2.4 reload_value	1053
6.51.2.5 running_mode	1053
6.52 STK_TypeDef Struct Reference	1053
6.52.1 Member Data Documentation	1053
6.52.1.1 CALIB	1053
6.52.1.2 CTRL	1053
6.52.1.3 LOAD	1054
6.52.1.4 VAL	1054
6.53 SYSCFG_RegDef_t Struct Reference	1054
6.53.1 Member Data Documentation	1054
6.53.1.1 CFGR	1054
6.53.1.2 CMPCR	1054
6.53.1.3 EXTICR	1055
6.53.1.4 MEMRMP	1055
6.53.1.5 PMC	1055
6.53.1.6 RESERVED1	1055
6.53.1.7 RESERVED2	1055
6.54 SYSCFG_TypeDef Struct Reference	1055
6.54.1 Detailed Description	1056
6.55 SysTick_Type Struct Reference	1056
6.55.1 Detailed Description	1056
6.56 TIM1_TypeDef Struct Reference	1056
6.56.1 Member Data Documentation	1057
6.56.1.1 ARR	1057
6.56.1.2 BDTR	1057
6.56.1.3 CCER	1057

6.56.1.4 CCMR1	1057
6.56.1.5 CCMR2	1057
6.56.1.6 CCR1	1057
6.56.1.7 CCR2	1057
6.56.1.8 CCR3	1058
6.56.1.9 CCR4	1058
6.56.1.10 CNT	1058
6.56.1.11 CR1	1058
6.56.1.12 CR2	1058
6.56.1.13 DCR	1058
6.56.1.14 DIER	1058
6.56.1.15 DMAR	1058
6.56.1.16 EGR	1059
6.56.1.17 OR	1059
6.56.1.18 PSC	1059
6.56.1.19 RCR	1059
6.56.1.20 SMCR	1059
6.56.1.21 SR	1059
6.57 TIM_TypeDef Struct Reference	1059
6.57.1 Detailed Description	1060
6.58 TPI_TypeDef Struct Reference	1060
6.58.1 Detailed Description	1061
6.59 USART_cfg_t Struct Reference	1061
6.59.1 Detailed Description	1062
6.60 USART_TypeDef Struct Reference	1062
6.60.1 Detailed Description	1062
6.60.2 Member Data Documentation	1062
6.60.2.1 BRR	1063
6.60.2.2 CR1	1063
6.60.2.3 CR2	1063
6.60.2.4 CR3	1063
6.60.2.5 DR	1063
6.60.2.6 GTPR	1063
6.60.2.7 SR	1063
6.61 WWDG_TypeDef Struct Reference	1064
6.61.1 Detailed Description	1064
6.62 xPSR_TypeDef Union Reference	1064
6.62.1 Detailed Description	1064
7 File Documentation	1065
7.1 bsp/Inc/eeprom_flash.h File Reference	1065
7.1.1 Detailed Description	1066

7.2 bsp/Inc/keypad_driver.h File Reference	1067
7.2.1 Detailed Description	1068
7.3 bsp/Inc/lcd_driver.h File Reference	1069
7.3.1 Detailed Description	1072
7.4 bsp/Inc/led_driver.h File Reference	1072
7.4.1 Detailed Description	1074
7.5 bsp/Src/eeprom_flash.c File Reference	1074
7.5.1 Detailed Description	1075
7.6 bsp/Src/keypad_driver.c File Reference	1075
7.6.1 Detailed Description	1076
7.7 bsp/Src/lcd_driver.c File Reference	1076
7.7.1 Detailed Description	1077
7.8 bsp/Src/led_driver.c File Reference	1078
7.8.1 Detailed Description	1078
7.9 CMSIS/Inc/core_cm4.h File Reference	1079
7.9.1 Detailed Description	1089
7.10 CMSIS/Inc/core_cmFunc.h File Reference	1089
7.10.1 Detailed Description	1089
7.11 CMSIS/Inc/core_cmlInstr.h File Reference	1090
7.11.1 Detailed Description	1090
7.12 CMSIS/Inc/core_cmSimd.h File Reference	1091
7.12.1 Detailed Description	1091
7.13 CMSIS/Inc/stm32f4xx.h File Reference	1092
7.13.1 Detailed Description	1183
7.14 CMSIS/Inc/system_stm32f4xx.h File Reference	1184
7.14.1 Detailed Description	1185
7.15 drivers/Inc/Platform_Types.h File Reference	1185
7.15.1 Detailed Description	1187
7.16 drivers/Inc/stm32f401xx.h File Reference	1187
7.16.1 Detailed Description	1196
7.17 drivers/Inc/stm32f401xx_gpio_driver.h File Reference	1197
7.17.1 Detailed Description	1199
7.18 drivers/Inc/stm32f401xx_i2c_driver.h File Reference	1200
7.18.1 Detailed Description	1202
7.19 drivers/Inc/stm32f401xx_nvic_driver.h File Reference	1202
7.19.1 Detailed Description	1206
7.20 drivers/Inc/stm32f401xx_rcc_driver.h File Reference	1207
7.20.1 Detailed Description	1208
7.21 drivers/Inc/stm32f401xx_spi_driver.h File Reference	1209
7.21.1 Detailed Description	1211
7.22 drivers/Inc/stm32f401xx_systick_driver.h File Reference	1211
7.22.1 Detailed Description	1212

7.23 drivers/Inc/stm32f401xx_timer_driver.h File Reference	1213
7.23.1 Detailed Description	1214
7.24 drivers/Inc/stm32f401xx_usart_driver.h File Reference	1215
7.24.1 Detailed Description	1217
7.25 drivers/Src/stm32f401xx_gpio_driver.c File Reference	1217
7.25.1 Detailed Description	1218
7.26 drivers/Src/stm32f401xx_i2c_driver.c File Reference	1218
7.26.1 Detailed Description	1219
7.26.2 Macro Definition Documentation	1220
7.26.2.1 I2C1_Index	1220
7.26.2.2 I2C2_Index	1220
7.26.3 Variable Documentation	1220
7.26.3.1 Global_I2C_Config	1220
7.27 drivers/Src/stm32f401xx_spi_driver.c File Reference	1220
7.27.1 Detailed Description	1221
7.27.2 Macro Definition Documentation	1222
7.27.2.1 SPI1_Index	1222
7.27.2.2 SPI2_Index	1222
7.27.2.3 SPI_SR_RXNE	1222
7.27.2.4 SPI_SR_TXE	1222
7.27.3 Function Documentation	1222
7.27.3.1 SPI1_IRQHandler()	1222
7.27.3.2 SPI2_IRQHandler()	1223
7.27.4 Variable Documentation	1223
7.27.4.1 Global_SPI_Config	1223
7.28 Inc/app_states.h File Reference	1223
7.28.1 Detailed Description	1225
7.29 Inc/ecu.h File Reference	1225
7.29.1 Detailed Description	1227
7.30 Src/app_states.c File Reference	1228
7.30.1 Detailed Description	1229
7.31 Src/ecu.c File Reference	1229
7.31.1 Detailed Description	1230
7.32 Src/sysmem.c File Reference	1231
7.32.1 Detailed Description	1232
7.32.2 Function Documentation	1232
7.32.2.1 _sbrk()	1232
7.33 Src/system_stm32f4xx.c File Reference	1233
7.33.1 Detailed Description	1234
7.33.2 5. This file configures the system clock as follows:	1234
7.33.2.1 Supported STM32F40xxx/41xxx devices	1235
7.33.2.2 System Clock source PLL (HSE)	1235

7.33.2.3 SYSCLK(Hz) 168000000	1235
7.33.2.4 HCLK(Hz) 168000000	1235
7.33.2.5 AHB Prescaler 1	1235
7.33.2.6 APB1 Prescaler 4	1235
7.33.2.7 APB2 Prescaler 2	1235
7.33.2.8 HSE Frequency(Hz) 25000000	1235
7.33.2.9 PLL_M 25	1235
7.33.2.10 PLL_N 336	1235
7.33.2.11 PLL_P 2	1235
7.33.2.12 PLL_Q 7	1235
7.33.2.13 PLLI2S_N NA	1235
7.33.2.14 PLLI2S_R NA	1235
7.33.2.15 I2S input clock NA	1235
7.33.2.16 VDD(V) 3.3	1235
7.33.2.17 Main regulator output voltage Scale1 mode	1235
7.33.2.18 Flash Latency(WS) 5	1235
7.33.2.19 Prefetch Buffer ON	1235
7.33.2.20 Instruction cache ON	1235
7.33.2.21 Data cache ON	1235
7.33.2.22 SDIO and RNG clock	1236
7.33.3 =====	1236
7.33.3.1 Supported STM32F42xxx/43xxx devices	1236
7.33.3.2 System Clock source PLL (HSE)	1236
7.33.3.3 SYSCLK(Hz) 180000000	1236
7.33.3.4 HCLK(Hz) 180000000	1236
7.33.3.5 AHB Prescaler 1	1236
7.33.3.6 APB1 Prescaler 4	1236
7.33.3.7 APB2 Prescaler 2	1236
7.33.3.8 HSE Frequency(Hz) 25000000	1236
7.33.3.9 PLL_M 25	1236
7.33.3.10 PLL_N 360	1236
7.33.3.11 PLL_P 2	1236
7.33.3.12 PLL_Q 7	1236
7.33.3.13 PLLI2S_N NA	1236
7.33.3.14 PLLI2S_R NA	1236
7.33.3.15 I2S input clock NA	1236
7.33.3.16 VDD(V) 3.3	1236
7.33.3.17 Main regulator output voltage Scale1 mode	1236
7.33.3.18 Flash Latency(WS) 5	1236
7.33.3.19 Prefetch Buffer ON	1236
7.33.3.20 Instruction cache ON	1236
7.33.3.21 Data cache ON	1236

7.33.3.22 SDIO and RNG clock 	1237
7.33.4 ======	1237
7.33.4.1 Supported STM32F401xx devices	1237
7.33.4.2 System Clock source PLL (HSE)	1237
7.33.4.3 SYCLK(Hz) 84000000	1237
7.33.4.4 HCLK(Hz) 84000000	1237
7.33.4.5 AHB Prescaler 1	1237
7.33.4.6 APB1 Prescaler 2	1237
7.33.4.7 APB2 Prescaler 1	1237
7.33.4.8 HSE Frequency(Hz) 25000000	1237
7.33.4.9 PLL_M 25	1237
7.33.4.10 PLL_N 336	1237
7.33.4.11 PLL_P 4	1237
7.33.4.12 PLL_Q 7	1237
7.33.4.13 PLLI2S_N NA	1237
7.33.4.14 PLLI2S_R NA	1237
7.33.4.15 I2S input clock NA	1237
7.33.4.16 VDD(V) 3.3	1237
7.33.4.17 Main regulator output voltage Scale1 mode	1237
7.33.4.18 Flash Latency(WS) 2	1237
7.33.4.19 Prefetch Buffer ON	1237
7.33.4.20 Instruction cache ON	1237
7.33.4.21 Data cache ON	1237
7.33.4.22 SDIO and RNG clock 	1238
7.33.5 ======	1238
7.33.5.1 Supported STM32F411xx/STM32F410xx devices	1238
7.33.5.2 System Clock source PLL (HSI)	1238
7.33.5.3 SYCLK(Hz) 100000000	1238
7.33.5.4 HCLK(Hz) 100000000	1238
7.33.5.5 AHB Prescaler 1	1238
7.33.5.6 APB1 Prescaler 2	1238
7.33.5.7 APB2 Prescaler 1	1238
7.33.5.8 HSI Frequency(Hz) 16000000	1238
7.33.5.9 PLL_M 16	1238
7.33.5.10 PLL_N 400	1238
7.33.5.11 PLL_P 4	1238
7.33.5.12 PLL_Q 7	1238
7.33.5.13 PLLI2S_N NA	1238
7.33.5.14 PLLI2S_R NA	1238
7.33.5.15 I2S input clock NA	1238
7.33.5.16 VDD(V) 3.3	1238
7.33.5.17 Main regulator output voltage Scale1 mode	1238

7.33.5.18 Flash Latency(WS) 3	1238
7.33.5.19 Prefetch Buffer ON	1238
7.33.5.20 Instruction cache ON	1238
7.33.5.21 Data cache ON	1238
7.33.5.22 SDIO and RNG clock 	1239
7.33.6 =====	1239
7.33.6.1 Supported STM32F446xx devices	1239
7.33.6.2 System Clock source PLL (HSE)	1239
7.33.6.3 SYSCLK(Hz) 180000000	1239
7.33.6.4 HCLK(Hz) 180000000	1239
7.33.6.5 AHB Prescaler 1	1239
7.33.6.6 APB1 Prescaler 4	1239
7.33.6.7 APB2 Prescaler 2	1239
7.33.6.8 HSE Frequency(Hz) 8000000	1239
7.33.6.9 PLL_M 8	1239
7.33.6.10 PLL_N 360	1239
7.33.6.11 PLL_P 2	1239
7.33.6.12 PLL_Q 7	1239
7.33.6.13 PLL_R NA	1239
7.33.6.14 PLLI2S_M NA	1239
7.33.6.15 PLLI2S_N NA	1239
7.33.6.16 PLLI2S_P NA	1239
7.33.6.17 PLLI2S_Q NA	1239
7.33.6.18 PLLI2S_R NA	1239
7.33.6.19 I2S input clock NA	1239
7.33.6.20 VDD(V) 3.3	1239
7.33.6.21 Main regulator output voltage Scale1 mode	1239
7.33.6.22 Flash Latency(WS) 5	1239
7.33.6.23 Prefetch Buffer ON	1239
7.33.6.24 Instruction cache ON	1239
7.33.6.25 Data cache ON	1239
7.33.6.26 SDIO and RNG clock 	1240

Chapter 1

PIR-Sensor-Library-for-Proteus

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

AT24C1024B EEPROM Driver	15
EEPROM Constants	15
API Function Prototypes	16
KEYPAD Driver	18
API Function Prototypes	20
LCD Driver	22
LCD Commands	23
LCD Row Positions	24
API Function Prototypes	25
LED Driver	31
Enum for specifying the LED mode.	32
Structure for configuring an LED.	33
API Function Prototypes	33
SERVO MOTOR Driver	37
Macros for the servo motor directions	38
API Function Prototypes	38
Platform Standard Types	148
STM32F401xx MCU Header File	156
Memory Base Addresses	157
Cortex-M4 Peripherals Base Addresses	157
AHB Peripherals Base Addresses	158
APB2 Peripherals Base Addresses	158
APB1 Peripherals Base Addresses	159
Peripheral Register Definitions	160
NVIC Register Map	161
SCB Register Map	162
SysTick Timer Register Map	162
GPIO Register Map	163
RCC Register Map	163
EXTI Register Map	164
SYSCFG Register Map	164
USART Register Map	165
SPI Register Map	165
I2C Register Map	166

CRC Register Map	166
TIM1 Register Map	167
Peripheral Instances	167
NVIC Instance	169
SCB Instance	169
SysTick Timer Instance	170
GPIO Instances	171
RCC Instance	172
EXTI Instance	173
SYSCFG Instance	174
Timer Instances	174
USART Instances	175
SPI Instances	176
I2C Instances	177
CRC Instance	178
I2C Registers	179
I2C CR1 Register Bit Definitions	180
I2C CR2 Register Bit Definitions	187
I2C OAR2 Register Bit Definitions	191
I2C SR1 Register Bit Definitions	192
I2C SR2 Register Bit Definitions	199
I2C CCR Register Bit Definitions	203
I2C TRISE Register Bit Definitions	205
RCC Bit Position Definitions	206
RCC_CR Bit Position Definitions	207
RCC_PLLCFGR Bit Position Definitions	210
RCC_CFGR Bit Position Definitions	211
RCC_CIR Bit Position Definitions	214
RCC_AHB1RSTR Bit Position Definitions	217
RCC_AHB2RSTR Bit Position Definitions	220
RCC_AHB3RSTR Bit Position Definitions	222
RCC_APB1RSTR Bit Position Definitions	222
RCC_APB2RSTR Bit Position Definitions	227
RCC_AHB1LPENR Bit Position Definitions	229
RCC_AHB2LPENR Bit Position Definitions	232
RCC_AHB3LPENR Bit Position Definitions	233
RCC_APB1LPENR Bit Position Definitions	234
RCC_APB2LPENR Bit Position Definitions	238
RCC_BDCR Bit Position Definitions	240
RCC_CSR Bit Position Definitions	242
RCC_SSCGR Bit Position Definitions	244
RCC_PLLI2SCFGR Bit Position Definitions	245
GPIO Base Address to Code Conversion Macros	246
GPIO Driver	247
GPIO Pin Numbers	248
GPIO Pin Modes	249
GPIO Pin Speeds	251
GPIO Pin States	252
GPIO Return Lock Status	253
GPIO Pin Modes	254
GPIO Output Speeds	255
GPIO Pull-up/Pull-down Configurations	257
GPIO Output Types	258
API Function Prototypes	259
I2C Driver	267
I2C Speed Definitions	271
I2C Stretch Mode Definitions	273

I2C Mode Definitions	273
I2C Addressing Modes	274
I2C Acknowledge Control Definitions	275
I2C General Call Enable Definitions	276
API Function Prototypes	277
NVIC Driver	285
NVIC Register Definitions	287
NVIC Configuration Macros	287
Interrupt Status	288
Priority Groups	288
Interrupt Priorities	289
Interrupt Requests	290
API Function Prototypes	291
NVIC IRQ Enable/Disable Macros	300
RCC Driver	305
Clock Source Selection	306
Peripheral Selection	308
API Function Prototypes	310
SPI Driver	315
SPI Mode Definitions	317
SPI Communication Mode Definitions	317
SPI Payload Length Definitions	318
SPI Frame Format Definitions	319
SPI Clock Polarity Definitions	320
SPI Clock Phase Definitions	321
SPI NSS Definitions	322
SPI Prescaler Definitions	323
SPI IRQ Enable Definitions	325
API Function Prototypes	326
SYSTICK Driver	331
SysTick Interrupt Configuration	332
SysTick Clock Configuration	333
SysTick Running Mode Configuration	334
API Function Prototypes	334
TIMER Driver	339
TIM2 Register group	340
API Function Prototypes	341
USART Driver	344
Enumeration and Structures for USART	344
Macros for USART Configuration	345
API Function Prototypes	346
Application States	350
Includes	354
State Macros	355
Prototypes	356
State Handlers	357
ECU Definitions and APIs	359
Includes	354
Enum representing the result of ID check.	360
Configuration Macros	360
ECU APIs	361
Functions_Declarations	369
Global_Variables_Definitions	370
API_Definitions	371
CMSIS	379
CMSIS Global Defines	42

Defines and Type Definitions	42
Status and Control Registers	43
Nested Vectored Interrupt Controller (NVIC)	49
System Control Block (SCB)	50
System Controls not in SCB (SCnSCB)	67
System Tick Timer (SysTick)	70
Instrumentation Trace Macrocell (ITM)	73
Data Watchpoint and Trace (DWT)	79
Trace Port Interface (TPI)	89
Core Debug Registers (CoreDebug)	99
Core Definitions	108
Functions and Instructions Reference	111
NVIC Functions	112
SysTick Functions	118
ITM Functions	119
CMSIS Core Register Access Functions	147
NVIC Functions	112
SysTick Functions	118
ITM Functions	119
Status and Control Registers	43
Nested Vectored Interrupt Controller (NVIC)	49
System Control Block (SCB)	50
System Controls not in SCB (SCnSCB)	67
System Tick Timer (SysTick)	70
Instrumentation Trace Macrocell (ITM)	73
Data Watchpoint and Trace (DWT)	79
Trace Port Interface (TPI)	89
Core Debug Registers (CoreDebug)	99
Core Definitions	108
Functions and Instructions Reference	111
NVIC Functions	112
SysTick Functions	118
ITM Functions	119
CMSIS Core Register Access Functions	147
NVIC Functions	112
SysTick Functions	118
ITM Functions	119
CMSIS Core Register Access Functions	147
CMSIS Core Instruction Interface	148
CMSIS SIMD Intrinsics	148
Stm32f4xx	519
Library_configuration_section	520
Configuration_section_for_CMSIS	522
Exported_types	527
Peripheral_registers_structures	530
Peripheral_memory_map	532
Peripheral_declaration	536
Exported_constants	539
Peripheral_Registers_Bits_Definition	539
Exported_macro	989
Stm32f4xx_system	990
STM32F4xx_System_Includes	991
STM32F4xx_System_Exported_types	991
STM32F4xx_System_Exported_Constants	992
STM32F4xx_System_Exported_Macros	992
STM32F4xx_System_Exported_Functions	992
STM32F4xx_System_Private_Includes	994
STM32F4xx_System_Private_TypesDefinitions	994

STM32F4xx_System_Private_Defines	995
STM32F4xx_System_Private_Macros	995
STM32F4xx_System_Private_Variables	996
STM32F4xx_System_Private_FunctionPrototypes	996
STM32F4xx_System_Private_Functions	996

Chapter 3

Class Index

3.1 Class List

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

ADC_Common_TypeDef	999
ADC_TypeDef	
Analog to Digital Converter	
999	
APSR_Type	
Union type to access the Application Program Status Register (APSR)	1000
CAN_FIFOMailBox_TypeDef	
Controller Area Network FIFOMailBox	1000
CAN_FilterRegister_TypeDef	
Controller Area Network FilterRegister	1001
CAN_TxMailBox_TypeDef	
Controller Area Network TxMailBox	1001
CAN_TypeDef	
Controller Area Network	1002
CONTROL_Type	
Union type to access the Control Registers (CONTROL)	1003
CoreDebug_Type	
Structure type to access the Core Debug Register (CoreDebug)	1003
CRC_TypeDef	
CRC calculation unit	1004
CRYP_TypeDef	
Crypto Processor	1005
DAC_TypeDef	
Digital to Analog Converter	1006
DBGMCU_TypeDef	
Debug MCU	1007
DCMI_TypeDef	
DCMI	1008
DMA2D_TypeDef	
DMA2D Controller	1008
DMA_Stream_TypeDef	
DMA Controller	1009
DMA_TypeDef	
Structure type to access the Data Watchpoint and Trace Register (DWT)	1010
DWT_Type	

ETH_TypeDef	Ethernet MAC	1011
EXTI_TypeDef	External Interrupt/Event Controller	1012
FLASH_TypeDef	FLASH Registers	1014
GPIO_PinConfig_t	Configuration structure for GPIO pins	1014
GPIO_TypeDef	General Purpose I/O	1016
HASH_DIGEST_TypeDef	HASH_DIGEST	1018
HASH_TypeDef	HASH	1018
I2C_TypeDef	Inter-integrated Circuit Interface	1019
IPSR_Type	Union type to access the Interrupt Program Status Register (IPSR)	1021
ITM_Type	Structure type to access the Instrumentation Trace Macrocell Register (ITM)	1022
IWDG_TypeDef	Independent WATCHDOG	1023
LCD_t	Structure for configuring the LCD	1023
LED_cfg_t	1026
LTDC_Layer_TypeDef	LCD-TFT Display layer x Controller	1027
LTDC_TypeDef	LCD-TFT Display Controller	1028
NVIC_Type	Structure type to access the Nested Vectored Interrupt Controller (NVIC)	1029
NVIC_TypeDef	1029
PWR_TypeDef	Power Control	1031
RCC_TypeDef	Reset and Clock Control	1032
RNG_TypeDef	RNG	1037
RTC_TypeDef	Real-Time Clock	1038
S_I2C_Config_t	Structure for I2C configuration	1039
S_I2C_Slave_address	Structure for I2C slave address configuration	1041
S_IRQ_SRC	Structure to identify the source of SPI interrupts	1042
S_SPI_Config_t	Configuration structure for SPI	1043
SAI_Block_TypeDef	1045
SAI_TypeDef	Serial Audio Interface	1045
SCB_Type	Structure type to access the System Control Block (SCB)	1046
SCB_TypeDef	1046
SCnSCB_Type	Structure type to access the System Control and ID Register not in the SCB	1048
SDIO_TypeDef	SD host Interface	1049

SPI_TypeDef	Serial Peripheral Interface	1050
STK_config_t	SysTick Configuration Structure Definition	1052
STK_TypeDef	1053
SYSCFG_RegDef_t	1054
SYSCFG_TypeDef	System configuration controller	1055
SysTick_Type	Structure type to access the System Timer (SysTick)	1056
TIM1_TypeDef	1056
TIM_TypeDef	TIM	1059
TPI_Type	Structure type to access the Trace Port Interface Register (TPI)	1060
USART_cfg_t	Configuration structure for USART (Universal Synchronous Asynchronous Receiver Transmitter) peripheral	1061
USART_TypeDef	Universal Synchronous Asynchronous Receiver Transmitter	1062
WWDG_TypeDef	Window WATCHDOG	1064
xPSR_Type	Union type to access the Special-Purpose Program Status Registers (xPSR)	1064

Chapter 4

File Index

4.1 File List

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

bsp/Inc/eeprom_flash.h	Header file for EEPROM flash memory operations	1065
bsp/Inc/keypad_driver.h	Header file for the STM32F401xx microcontroller keypad driver	1067
bsp/Inc/lcd_driver.h	Header file for the STM32F401xx microcontroller LCD driver	1069
bsp/Inc/led_driver.h	Header file for the STM32F401xx microcontroller LED driver	1072
bsp/Inc/servo_motor_driver.h	??
bsp/Src/eeprom_flash.c	Implementation of EEPROM flash memory operations for STM32F401xx MCU	1074
bsp/Src/keypad_driver.c	Source file for the keypad driver, implementing functionality to initialize and read input from a keypad connected to the STM32F401xx MCU	1075
bsp/Src/lcd_driver.c	Contains functions for controlling an LCD using an STM32F401xx MCU	1076
bsp/Src/led_driver.c	LED Driver Functions for STM32F401xx MCU	1078
CMSIS/Inc/core_cm4.h	CMSIS Cortex-M4 Core Peripheral Access Layer Header File	1079
CMSIS/Inc/core_cmFunc.h	CMSIS Cortex-M Core Function Access Header File	1089
CMSIS/Inc/core_cmlInstr.h	CMSIS Cortex-M Core Instruction Access Header File	1090
CMSIS/Inc/core_cmSimd.h	CMSIS Cortex-M SIMD Header File	1091
CMSIS/Inc/stm32f4xx.h	CMSIS Cortex-M4 Device Peripheral Access Layer Header File. This file contains all the peripheral register's definitions, bits definitions and memory mapping for STM32F4xx devices. 1092	
CMSIS/Inc/stm32f4xx_it.h	??
CMSIS/Inc/system_stm32f4xx.h	CMSIS Cortex-M4 Device System Source File for STM32F4xx devices. 1184	
drivers/Inc/Platform_Types.h	Defines data types and macros for STM32F401xx microcontrollers	1185

drivers/Inc/stm32f401xx.h	
Header file containing all the necessary information about the STM32F401xx MCU	1187
drivers/Inc/stm32f401xx_gpio_driver.h	
Header file for GPIO driver for STM32F401xx microcontroller	1197
drivers/Inc/stm32f401xx_i2c_driver.h	
Header file for STM32F401xx I2C driver	1200
drivers/Inc/stm32f401xx_nvic_driver.h	
Header file for NVIC (Nested Vectored Interrupt Controller) driver for STM32F401xx microcontroller	1202
drivers/Inc/stm32f401xx_rcc_driver.h	
Header file for RCC (Reset and Clock Control) driver for STM32F401xx MCU	1207
drivers/Inc/stm32f401xx_spi_driver.h	
This file contains definitions and function prototypes for the STM32F401xx SPI driver	1209
drivers/Inc/stm32f401xx_systick_driver.h	
Header file for configuring and controlling the SysTick timer in the STM32F401xx MCU. This file contains the necessary structures, macros, and function prototypes for SysTick initialization, configuration, and handling delays	1211
drivers/Inc/stm32f401xx_timer_driver.h	
Header file providing timer functionalities for the STM32F401xx MCU. This file contains the necessary definitions, macros, and functions to manage Timer 2 (TIM2) in the STM32F401xx microcontroller	1213
drivers/Inc/stm32f401xx_usart_driver.h	
USART driver header file for STM32F401 series microcontrollers	1215
drivers/Src/stm32f401xx_gpio_driver.c	
GPIO driver for STM32F401xx microcontroller	1217
drivers/Src/stm32f401xx_i2c_driver.c	
I2C Peripheral Driver Implementation for STM32F103C6	1218
drivers/Src/stm32f401xx_spi_driver.c	
Contains the implementation of the SPI driver APIs for STM32F401	1220
Inc/app_states.h	
Header file containing definitions and macros for managing application states on the STM32F401xx MCU	1223
Inc/ecu.h	
Header file containing all the necessary information about the STM32F401xx MCU	1225
Src/app_states.c	
Application state machine implementation for managing entry/exit gates and parking slots using the STM32F401xx MCU	1228
Src/ecu.c	
Header file containing all necessary information and functionality for managing STM32F401xx MCU-based peripherals	1229
Src/sysmem.c	
STM32CubeIDE System Memory calls file	1231
Src/system_stm32f4xx.c	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File. This file contains the system clock configuration for STM32F4xx devices	1233

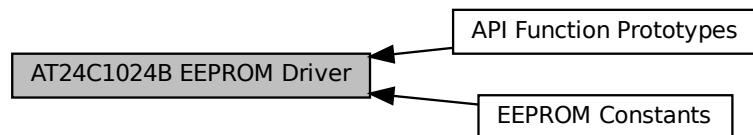
Chapter 5

Module Documentation

5.1 AT24C1024B EEPROM Driver

AT24C1024B EEPROM driver for STM32F401xx MCUs.

Collaboration diagram for AT24C1024B EEPROM Driver:



Modules

- [EEPROM Constants](#)
- [API Function Prototypes](#)

5.1.1 Detailed Description

AT24C1024B EEPROM driver for STM32F401xx MCUs.

5.2 EEPROM Constants

Collaboration diagram for EEPROM Constants:



Macros

- `#define EEPROM_Slave_address 0x50`
AT24C1024B EEPROM slave address for I2C communication.

5.2.1 Detailed Description

5.2.2 Macro Definition Documentation

5.2.2.1 EEPROM_Slave_address

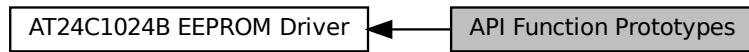
```
#define EEPROM_Slave_address 0x50
```

AT24C1024B EEPROM slave address for I2C communication.

This address is used to identify the AT24C1024B EEPROM device on the I2C bus.

5.3 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- `void EEPROM_Init (void)`
Initializes the EEPROM for communication.
- `unsigned char EEPROM_Write_NBytes (unsigned int Memory_address, unsigned char *bytes, uint8_t Data_Length)`
Writes multiple bytes to the EEPROM.
- `unsigned char EEPROM_Read_byte (unsigned int Memory_address, uint8_t *dataout, uint8_t datalen)`
Reads a byte from the EEPROM.

5.3.1 Detailed Description

5.3.2 Function Documentation

5.3.2.1 EEPROM_Init()

```
void EEPROM_Init (
    void )
```

Initializes the EEPROM for communication.

This function sets up the necessary configuration to communicate with the EEPROM over I2C. It should be called before any read or write operations are performed.

Initializes the EEPROM for communication.

This function configures the I2C1 peripheral of the STM32F401xx MCU for communication with the EEPROM. It sets up the I2C speed, address detection, and other relevant parameters.

- Configures PB6 as I2C1_SCL and PB7 as I2C1_SDA.
- Enables general call address detection.
- Enables acknowledgment control.
- Sets I2C speed to 100kHz.
- Configures I2C mode and stretch mode.

5.3.2.2 EEPROM_Read_byte()

```
unsigned char EEPROM_Read_byte (
    unsigned int Memory_address,
    uint8_t * dataout,
    uint8_t datalen )
```

Reads a byte from the EEPROM.

This function reads a single byte of data from a specified memory address in the EEPROM.

Parameters

in	<i>Memory_address</i>	The address in the EEPROM from which to read the data.
out	<i>dataout</i>	Pointer to a variable where the read byte will be stored.
in	<i>datalen</i>	The number of bytes to read. In this case, should be 1 for a single byte read.

Returns

Status of the read operation. Returns 0 for success, non-zero for failure.

This function reads a single byte from a specified memory address in the EEPROM.

Parameters

in	<i>Memory_address</i>	The address in the EEPROM from which to read the data.
out	<i>dataout</i>	Pointer to a variable where the read byte will be stored.
in	<i>datalen</i>	The number of bytes to read. For this function, should be 1.

Generated by Doxygen

Returns

Status of the read operation. Returns 0 for success.

< High byte of the memory address

< Low byte of the memory address

5.3.2.3 EEPROM_Write_NBytes()

```
unsigned char EEPROM_Write_NBytes (
    unsigned int Memory_address,
    unsigned char * bytes,
    uint8_t Data_Length )
```

Writes multiple bytes to the EEPROM.

This function writes a block of data to a specified memory address in the EEPROM.

Parameters

in	<i>Memory_address</i>	The starting address in the EEPROM where the data will be written.
in	<i>bytes</i>	Pointer to the data buffer to be written to the EEPROM.
in	<i>Data_Length</i>	The number of bytes to write to the EEPROM.

Returns

Status of the write operation. Returns 0 for success, non-zero for failure.

Writes multiple bytes to the EEPROM.

This function writes multiple bytes to a specified memory address in the EEPROM.

Parameters

in	<i>Memory_address</i>	The starting address in the EEPROM where data will be written.
in	<i>bytes</i>	Pointer to the data buffer to be written to the EEPROM.
in	<i>Data_Length</i>	The number of bytes to write.

Returns

Status of the write operation. Returns 0 for success.

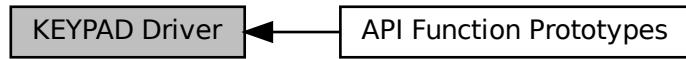
< High byte of the memory address

< Low byte of the memory address

5.4 KEYPAD Driver

Driver for interfacing with a keypad for STM32F401xx MCUs.

Collaboration diagram for KEYPAD Driver:



Modules

- [API Function Prototypes](#)

Macros

- `#define KEYPAD_PORT GPIOE`
Defines the GPIO port and pins for the keypad.
- `#define KEYPAD_ROWS 4`
- `#define ROW0 GPIO_PIN_0`
- `#define ROW1 GPIO_PIN_1`
- `#define ROW2 GPIO_PIN_3`
- `#define ROW3 GPIO_PIN_4`
- `#define KEYPAD_COLS 3`
- `#define COL0 GPIO_PIN_5`
- `#define COL1 GPIO_PIN_6`
- `#define COL2 GPIO_PIN_7`

5.4.1 Detailed Description

Driver for interfacing with a keypad for STM32F401xx MCUs.

5.4.2 Macro Definition Documentation

5.4.2.1 KEYPAD_PORT

```
#define KEYPAD_PORT GPIOE
```

Defines the GPIO port and pins for the keypad.

The following macros are used to configure the keypad:

- KEYPAD_PORT: GPIO port used for the keypad.
- KEYPAD_ROWS: Number of rows in the keypad.
- ROW0, ROW1, ROW2, ROW3: GPIO pins for each row.
- KEYPAD_COLS: Number of columns in the keypad.
- COL0, COL1, COL2: GPIO pins for each column.

5.5 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void `keypad_init` (void)
Initializes the keypad.
- `uint8 keypad_Get_Pressed_Key` (void)
Checks for a pressed key and returns its value.

5.5.1 Detailed Description

5.5.2 Function Documentation

5.5.2.1 `keypad_Get_Pressed_Key()`

```
uint8 keypad_Get_Pressed_Key (
    void )
```

Checks for a pressed key and returns its value.

This function scans the keypad to determine if any key is pressed and returns the corresponding key value. If no key is pressed, it returns the value 'F'.

Parameters

in	<i>None</i>	
out	<i>None</i>	

Returns

`uint8` Value of the pressed key, or 'F' if no key is pressed.

Note

None

Checks for a pressed key and returns its value.

This function scans the keypad to detect any pressed key and returns the corresponding character. If no key is pressed, it returns 'F'.

Returns

Value of the pressed key, or 'F' if no key is pressed.

5.5.2.2 keypad_init()

```
void keypad_init (
    void )
```

Initializes the keypad.

This function configures the GPIO pins for the keypad as defined by the macros in the "Macros Configuration" section. It sets up the keypad for operation by initializing the necessary hardware.

Parameters

in	<i>None</i>	
out	<i>None</i>	

Returns

None

Note

User must define GPIO pins for rows and columns in Keypad_PINS_define.

Initializes the keypad.

This function sets up the GPIO pins for keypad rows as outputs and columns as inputs. It also sets the default state of the rows to low and configures pull-down resistors for the column pins.

Returns

None

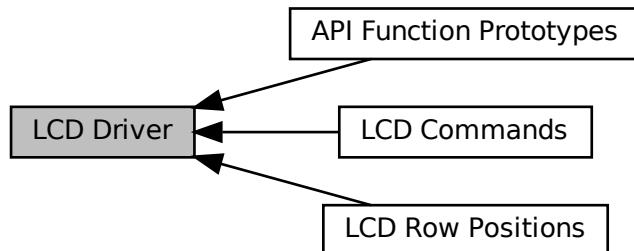
Note

User must define GPIO pins for rows and columns in Keypad_PINS_define.

5.6 LCD Driver

LCD (Liquid Crystal Display) driver for STM32F401xx MCUs.

Collaboration diagram for LCD Driver:



Modules

- [LCD Commands](#)
- [LCD Row Positions](#)
- [API Function Prototypes](#)

Classes

- struct [LCD_t](#)
Structure for configuring the LCD.

Enumerations

- enum [LCD_MODE_t](#) { [LCD_8BIT](#) , [LCD_4BIT](#) }
Enum for specifying the LCD mode.
- enum [LCD_ROWS_t](#) { [LCD_2ROWS](#) , [LCD_4ROWS](#) }
Enum for specifying the number of rows on the LCD.

5.6.1 Detailed Description

LCD (Liquid Crystal Display) driver for STM32F401xx MCUs.

5.6.2 Enumeration Type Documentation

5.6.2.1 LCD_MODE_t

```
enum LCD\_MODE\_t
```

Enum for specifying the LCD mode.

Enumerator

LCD_8BIT	8-bit mode for LCD communication.
LCD_4BIT	4-bit mode for LCD communication.

5.6.2.2 LCD_ROWS_t

```
enum LCD_ROWS_t
```

Enum for specifying the number of rows on the LCD.

Enumerator

LCD_2ROWS	LCD with 2 rows.
LCD_4ROWS	LCD with 4 rows.

5.7 LCD Commands

Collaboration diagram for LCD Commands:

**Macros**

- #define [LCD_CLEAR_DISPLAY](#) (0x01)
Command to clear the LCD display.
- #define [LCD_RETURN_HOME](#) (0x02)
Command to return the cursor to the home position.
- #define [LCD_ENTRY_MODE_DEC_SHIFT_OFF](#) (0x04)
Command to set the entry mode to decrement and shift off.
- #define [LCD_ENTRY_MODE_DEC_SHIFT_ON](#) (0x05)
Command to set the entry mode to decrement and shift on.
- #define [LCD_ENTRY_MODE_INC_SHIFT_OFF](#) (0x06)
Command to set the entry mode to increment and shift off.
- #define [LCD_ENTRY_MODE_INC_SHIFT_ON](#) (0x07)
Command to set the entry mode to increment and shift on.
- #define [LCD_CURSOR_MOVE_SHIFT_LEFT](#) (0x10)

- `#define LCD_CURSOR_MOVE_SHIFT_RIGHT (0x14)`

Command to move the cursor left.
- `#define LCD_DISPLAY_SHIFT_LEFT (0x18)`

Command to move the cursor right.
- `#define LCD_DISPLAY_SHIFT_RIGHT (0x1C)`

Command to shift the display left.
- `#define LCD_DISPLAY_ON_UNDERLINE_OFF_CURSOR_OFF (0x0C)`

Command to turn on the display with underline off and cursor off.
- `#define LCD_DISPLAY_ON_UNDERLINE_OFF_CURSOR_ON (0x0D)`

Command to turn on the display with underline off and cursor on.
- `#define LCD_DISPLAY_ON_UNDERLINE_ON_CURSOR_OFF (0x0E)`

Command to turn on the display with underline on and cursor off.
- `#define LCD_DISPLAY_ON_UNDERLINE_ON_CURSOR_ON (0x0F)`

Command to turn on the display with underline on and cursor on.
- `#define LCD_DISPLAY_OFF_CURSOR_OFF (0x08)`

Command to turn off the display with cursor off.
- `#define LCD_8BIT_MODE_2_LINE (0x38)`

Command to set the display in 8-bit mode with 2 lines.
- `#define LCD_4BIT_MODE_2_LINE (0x28)`

Command to set the display in 4-bit mode with 2 lines.

5.7.1 Detailed Description

5.8 LCD Row Positions

Collaboration diagram for LCD Row Positions:



Macros

- `#define LCD_FIRST_ROW (0x80)`

Command to set the cursor position to the first row.
- `#define LCD_SECOND_ROW (0xC0)`

Command to set the cursor position to the second row.
- `#define LCD_THIRD_ROW (0x94)`

Command to set the cursor position to the third row.
- `#define LCD_FOURTH_ROW (0xD4)`

Command to set the cursor position to the fourth row.

5.8.1 Detailed Description

5.9 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void `LCD_Init (LCD_t *LCD_cfg)`
Initializes the LCD with user-defined configurations.
- void `LCD_Send_Command (LCD_t *LCD_cfg, uint8 command)`
Sends a command to the LCD to be executed.
- void `LCD_Send_Char (LCD_t *LCD_cfg, uint8 Char)`
Sends a character to the LCD to be displayed.
- void `LCD_Send_Char_Pos (LCD_t *LCD_cfg, uint8 Char, uint8 row, uint8 column)`
Sends a character to the LCD to be displayed at a specific location.
- void `LCD_Send_String (LCD_t *LCD_cfg, uint8 *string)`
Sends a string to the LCD to be displayed.
- void `LCD_Send_String_Pos (LCD_t *LCD_cfg, uint8 *string, uint8 row, uint8 column)`
Sends a string to the LCD to be displayed at a specific location.
- void `LCD_Send_Enable_Signal (LCD_t *LCD_cfg)`
Sends an enable signal to the LCD.
- void `LCD_Set_Cursor (LCD_t *LCD_cfg, uint8 row, uint8 column)`
Sets the location of the cursor on the LCD.

5.9.1 Detailed Description

5.9.2 Function Documentation

5.9.2.1 LCD_Init()

```
void LCD_Init (
    LCD_t * LCD_cfg )
```

Initializes the LCD with user-defined configurations.

This function sets up the LCD based on the configuration specified in the provided structure. It initializes the LCD in the mode and row settings defined by the user.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
----	----------------	--

Returns

None

Note

User must set configurations using the [LCD Commands](#) macros.

Initializes the LCD with user-defined configurations.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
----	----------------	--

This function initializes the LCD by setting up the GPIO pins, sending function set commands, and configuring display and entry modes. It also clears the display.

Note

User must set configurations as defined in `LCD_CONFIG_define`.

5.9.2.2 LCD_Send_Char()

```
void LCD_Send_Char (
    LCD_t * LCD_cfg,
    uint8 Char )
```

Sends a character to the LCD to be displayed.

This function sends an ASCII character to the LCD to be displayed at the current cursor position.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>Char</i>	ASCII character to be displayed on the screen.

Returns

None

Note

None

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>Char</i>	ASCII character to be displayed.

This function sends a character to the LCD to be displayed. The character is sent in either 8-bit or 4-bit mode depending on the LCD configuration.

5.9.2.3 LCD_Send_Char_Pos()

```
void LCD_Send_Char_Pos (
    LCD_t * LCD_cfg,
    uint8 Char,
    uint8 row,
    uint8 column )
```

Sends a character to the LCD to be displayed at a specific location.

This function sends an ASCII character to the LCD to be displayed at a specific row and column location.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>Char</i>	ASCII character to be displayed on the screen.
in	<i>row</i>	Row number where the character should be displayed (see LCD Row Positions).
in	<i>column</i>	Column number where the character should be displayed (1...16).

Returns

None

Note

None

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>Char</i>	ASCII character to be displayed on the screen.
in	<i>row</i>	Selects the row number of the displayed character (LCD Row Positions).
in	<i>column</i>	Selects the column number of the displayed character (1...16).

Returns

None

This function sets the cursor to the specified row and column and sends the given character to the LCD for display.

5.9.2.4 LCD_Send_Command()

```
void LCD_Send_Command (
    LCD_t * LCD_cfg,
    uint8 command )
```

Sends a command to the LCD to be executed.

This function sends a command to the LCD to perform a specific action, such as clearing the display or shifting the cursor.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>command</i>	Command to be executed (see LCD Commands).

Returns

None

Note

None

Sends a command to the LCD to be executed.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>command</i>	Command to be executed as defined in LCD Commands .

This function sends a command to the LCD for execution. The command is sent in either 8-bit or 4-bit mode depending on the LCD configuration.

5.9.2.5 LCD_Send_Enable_Signal()

```
void LCD_Send_Enable_Signal (
    LCD_t * LCD_cfg )
```

Sends an enable signal to the LCD.

This function triggers the enable signal to the LCD, which is necessary for executing commands and data transfers. The signal is typically sent for a short duration to latch the data or command on the LCD.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
----	----------------	--

Returns

None

Note

This function assumes that the enable pin is properly configured in the `LCD_cfg` structure.

Parameters

in	<code>LCD_cfg</code>	Pointer to the structure containing LCD configuration.
----	----------------------	--

Returns

None

This function generates an enable pulse for the LCD by toggling the enable pin.

5.9.2.6 `LCD_Send_String()`

```
void LCD_Send_String (
    LCD_t * LCD_cfg,
    uint8 * string )
```

Sends a string to the LCD to be displayed.

This function sends a string of characters to the LCD to be displayed from the current cursor position.

Parameters

in	<code>LCD_cfg</code>	Pointer to the structure containing LCD configuration.
in	<code>string</code>	Pointer to a string of characters to be displayed on the LCD.

Returns

None

Note

None

Parameters

in	<code>LCD_cfg</code>	Pointer to the structure containing LCD configuration.
in	<code>string</code>	Pointer to a string of characters to be displayed on the LCD.

Returns

None

This function sends a string to the LCD, character by character, until the null-terminator is encountered.

5.9.2.7 LCD_Send_String_Pos()

```
void LCD_Send_String_Pos (
    LCD_t * LCD_cfg,
    uint8 * string,
    uint8 row,
    uint8 column )
```

Sends a string to the LCD to be displayed at a specific location.

This function sends a string of characters to the LCD to be displayed at a specific row and column location.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>string</i>	Pointer to a string of characters to be displayed on the LCD.
in	<i>row</i>	Row number where the string should be displayed (see LCD Row Positions).
in	<i>column</i>	Column number where the string should be displayed (1...16).

Returns

None

Note

None

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>string</i>	Pointer to a string of characters to be displayed on the LCD.
in	<i>row</i>	Selects the row number of the displayed character (LCD Row Positions).
in	<i>column</i>	Selects the column number of the displayed character (1...16).

Returns

None

This function sets the cursor to the specified row and column, then sends the given string to the LCD for display.

5.9.2.8 LCD_Set_Cursor()

```
void LCD_Set_Cursor (
    LCD_t * LCD_cfg,
```

```
    uint8 row,  
    uint8 column )
```

Sets the location of the cursor on the LCD.

This function positions the cursor to the specified row and column on the LCD. This is useful for placing characters at specific locations on the display.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>row</i>	Selects the row number of the displayed character (see LCD Row Positions).
in	<i>column</i>	Selects the column number of the displayed character (1...16).

Returns

None

Note

This function assumes that the LCD is initialized and configured correctly according to the `LCD_cfg` structure.

Parameters

in	<i>LCD_cfg</i>	Pointer to the structure containing LCD configuration.
in	<i>row</i>	Selects the row number of the displayed character (LCD Row Positions).
in	<i>column</i>	Selects the column number of the displayed character (1...16).

Returns

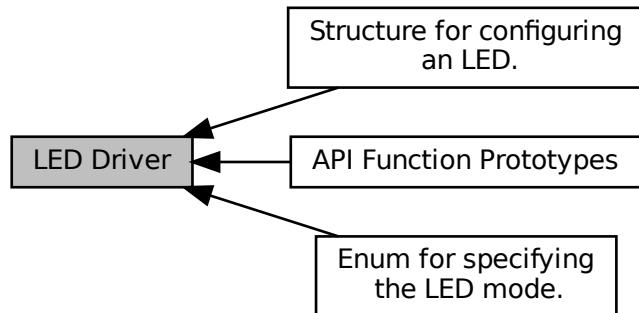
None

This function sets the cursor position to the specified row and column by sending the appropriate command to the LCD.

5.10 LED Driver

LED driver for STM32F401xx MCUs.

Collaboration diagram for LED Driver:



Modules

- [Enum for specifying the LED mode.](#)
- [Structure for configuring an LED.](#)
- [API Function Prototypes](#)

5.10.1 Detailed Description

LED driver for STM32F401xx MCUs.

5.11 Enum for specifying the LED mode.

Collaboration diagram for Enum for specifying the LED mode.:



Enumerations

- enum [LED_Mode_t](#) { [LED_Active_High](#) , [LED_Active_Low](#) , [LED_Mode_max](#) }

5.11.1 Detailed Description

5.11.2 Enumeration Type Documentation

5.11.2.1 LED_Mode_t

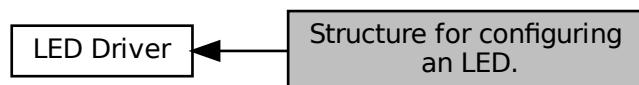
```
enum LED_Mode_t
```

Enumerator

LED_Active_High	LED is active when the GPIO pin is high.
LED_Active_Low	LED is active when the GPIO pin is low.
LED_Mode_max	Maximum value for LED mode enumeration.

5.12 Structure for configuring an LED.

Collaboration diagram for Structure for configuring an LED.:



Classes

- struct [LED_cfg_t](#)

5.12.1 Detailed Description

5.13 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void `LED_Init` (const `LED_cfg_t` *led_cfg)
Initializes the LED with the specified configuration.
- void `LED_TurnOn` (const `LED_cfg_t` *led_cfg)
Turns on the LED.
- void `LED_TurnOff` (const `LED_cfg_t` *led_cfg)
Turns off the LED.
- void `LED_Toggle` (const `LED_cfg_t` *led_cfg)
Toggles the state of the LED.

5.13.1 Detailed Description

5.13.2 Function Documentation

5.13.2.1 `LED_Init()`

```
void LED_Init (
    const LED_cfg_t * led_cfg )
```

Initializes the LED with the specified configuration.

This function configures the GPIO pin according to the settings provided in the `led_cfg` structure. The LED is initially off after initialization.

Parameters

in	<code>led_cfg</code>	Pointer to the structure holding the LED configuration.
----	----------------------	---

Returns

None

Note

Ensure that the GPIO port and pin configuration in `led_cfg` are correctly set before calling this function.

Initializes the LED with the specified configuration.

Parameters

in	<code>led_cfg</code>	Pointer to the structure holding the LED configuration.
----	----------------------	---

Returns

None

This function initializes the LED by configuring the GPIO pin according to the settings specified in the `led_cfg` structure. The LED is initially turned off based on its mode (active high or active low).

- If the LED mode is `LED_Active_High`, the LED is turned off by setting the GPIO pin to reset.
- If the LED mode is `LED_Active_Low`, the LED is turned off by setting the GPIO pin to set.

5.13.2.2 LED_Toggle()

```
void LED_Toggle (
    const LED_cfg_t * led_cfg )
```

Toggles the state of the LED.

This function changes the LED state from on to off or off to on by toggling the GPIO pin according to the configuration provided in `led_cfg`.

Parameters

in	<code>led_cfg</code>	Pointer to the structure holding the LED configuration.
----	----------------------	---

Returns

None

Note

The behavior of this function depends on the LED mode configured in `led_cfg`. For `LED_Active_High`, the pin is toggled between high and low; for `LED_Active_Low`, the pin is toggled between low and high.

Toggles the state of the LED.

Parameters

in	<code>led_cfg</code>	Pointer to the structure holding the LED configuration.
----	----------------------	---

Returns

None

This function toggles the state of the LED by changing the state of the GPIO pin. The LED's state is switched between on and off.

5.13.2.3 LED_TurnOff()

```
void LED_TurnOff (
    const LED_cfg_t * led_cfg )
```

Turns off the LED.

This function sets the GPIO pin to the appropriate state to turn off the LED based on the configuration provided in `led_cfg`.

Parameters

in	<code>led_cfg</code>	Pointer to the structure holding the LED configuration.
----	----------------------	---

Returns

None

Note

The behavior of this function depends on the LED mode configured in `led_cfg`. For `LED_Active_High`, the pin is set low; for `LED_Active_Low`, the pin is set high.

Parameters

in	<code>led_cfg</code>	Pointer to the structure holding the LED configuration.
----	----------------------	---

Returns

None

This function turns off the LED based on its mode. If the mode is active high, the GPIO pin is reset. If the mode is active low, the GPIO pin is set.

5.13.2.4 LED_TurnOn()

```
void LED_TurnOn (
    const LED_cfg_t * led_cfg )
```

Turns on the LED.

This function sets the GPIO pin to the appropriate state to turn on the LED based on the configuration provided in `led_cfg`.

Parameters

in	<code>led_cfg</code>	Pointer to the structure holding the LED configuration.
----	----------------------	---

Returns

None

Note

The behavior of this function depends on the LED mode configured in `led_cfg`. For `LED_Active_High`, the pin is set high; for `LED_Active_Low`, the pin is set low.

Parameters

in	<code>led_cfg</code>	Pointer to the structure holding the LED configuration.
----	----------------------	---

Returns

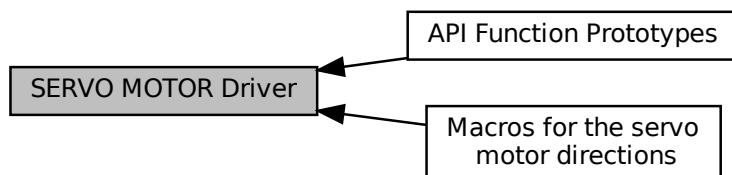
None

This function turns on the LED based on its mode. If the mode is active high, the GPIO pin is set. If the mode is active low, the GPIO pin is reset.

5.14 SERVO MOTOR Driver

SERVO MOTOR driver for STM32F401xx MCUs.

Collaboration diagram for SERVO MOTOR Driver:



Modules

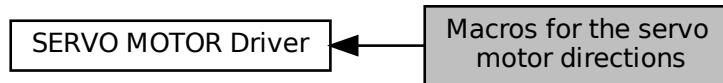
- Macros for the servo motor directions
- API Function Prototypes

5.14.1 Detailed Description

SERVO MOTOR driver for STM32F401xx MCUs.

5.15 Macros for the servo motor directions

Collaboration diagram for Macros for the servo motor directions:



Macros

- #define SERVO_UP 0
Macro for the servo motor direction to move the gate upwards.
- #define SERVO_DOWN 1
Macro for the servo motor direction to move the gate downwards.

5.15.1 Detailed Description

5.16 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void Servo1_Entry_Gate_Init (void)
Initializes the servo motor for the entry gate.
- void Servo1_Entry_Gate (uint8 Direction)
Controls the servo motor for the entry gate.
- void Servo2_Exit_Gate_Init (void)
Initializes the servo motor for the exit gate.
- void Servo2_Exit_Gate (uint8 Direction)
Controls the servo motor for the exit gate.

5.16.1 Detailed Description

5.16.2 Function Documentation

5.16.2.1 Servo1_Entry_Gate()

```
void Servo1_Entry_Gate (
    uint8 Direction )
```

Controls the servo motor for the entry gate.

This function sets the servo motor to the specified direction (up or down) based on the `Direction` parameter.

Parameters

in	<code>Direction</code>	Specifies the direction of the servo motor. This can be either SERVO_UP to move the gate upwards or SERVO_DOWN to move the gate downwards.
----	------------------------	--

Returns

None

Note

Ensure that `Servo1_Entry_Gate_Init()` is called prior to using this function.

Controls the servo motor for the entry gate.

Parameters

in	<code>Direction</code>	Direction for the servo motor movement (SERVO_UP or SERVO_DOWN).
----	------------------------	--

Returns

None

This function controls the servo motor connected to pin B8 to either move the entry gate up or down.

- If `Direction` is SERVO_UP, the servo moves to +90 degrees.
- If `Direction` is SERVO_DOWN, the servo moves to -90 degrees. The movement duration is controlled by setting the pin high, waiting for the required duration, and then setting the pin low.

5.16.2.2 Servo1_Entry_Gate_Init()

```
void Servo1_Entry_Gate_Init (
    void )
```

Initializes the servo motor for the entry gate.

This function configures the necessary GPIO and timer settings for controlling the servo motor associated with the entry gate.

Returns

None

Note

This function must be called before operating the entry gate servo motor.

Initializes the servo motor for the entry gate.

Parameters

None	<input type="button" value=""/>
------	---------------------------------

Returns

None

This function configures the GPIO pin B8 as an output pin to control Servo Motor 1. The pin configuration includes setting the mode to output, push-pull operation type, no pull-up/pull-down, and fast speed.

5.16.2.3 Servo2_Exit_Gate()

```
void Servo2_Exit_Gate (
    uint8 Direction )
```

Controls the servo motor for the exit gate.

This function sets the servo motor to the specified direction (up or down) based on the *Direction* parameter.

Parameters

in	<i>Direction</i>	Specifies the direction of the servo motor. This can be either SERVO_UP to move the gate upwards or SERVO_DOWN to move the gate downwards.
----	------------------	--

Returns

None

Note

Ensure that `Servo2_Exit_Gate_Init()` is called prior to using this function.

Controls the servo motor for the exit gate.

Parameters

in	<i>Direction</i>	Direction for the servo motor movement (SERVO_UP or SERVO_DOWN).
----	------------------	--

Returns

None

This function controls the servo motor connected to pin B9 to either move the exit gate up or down.

- If *Direction* is SERVO_UP, the servo moves to +90 degrees.
- If *Direction* is SERVO_DOWN, the servo moves to -90 degrees. The movement duration is controlled by setting the pin high, waiting for the required duration, and then setting the pin low.

5.16.2.4 Servo2_Exit_Gate_Init()

```
void Servo2_Exit_Gate_Init (
    void )
```

Initializes the servo motor for the exit gate.

This function configures the necessary GPIO and timer settings for controlling the servo motor associated with the exit gate.

Returns

None

Note

This function must be called before operating the exit gate servo motor.

Initializes the servo motor for the exit gate.

Parameters

<i>None</i>	<input type="checkbox"/>
-------------	--------------------------

Returns

None

This function configures the GPIO pin B9 as an output pin to control Servo Motor 2. The pin configuration includes setting the mode to output, push-pull operation type, no pull-up/pull-down, and fast speed.

5.17 CMSIS Global Defines

Collaboration diagram for CMSIS Global Defines:



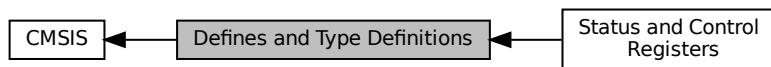
IO Type Qualifiers are used

- to specify the access to peripheral variables.
- for automatic generation of peripheral register debug information.

5.18 Defines and Type Definitions

Type definitions and defines for Cortex-M processor based devices.

Collaboration diagram for Defines and Type Definitions:



Modules

- Status and Control Registers
Core Register type definitions.

5.18.1 Detailed Description

Type definitions and defines for Cortex-M processor based devices.

5.19 Status and Control Registers

Core Register type definitions.

Collaboration diagram for Status and Control Registers:



Modules

- [Nested Vectored Interrupt Controller \(NVIC\)](#)

Type definitions for the NVIC Registers.

Classes

- union [APSR_Type](#)
Union type to access the Application Program Status Register (APSR).
- union [IPSR_Type](#)
Union type to access the Interrupt Program Status Register (IPSR).
- union [xPSR_Type](#)
Union type to access the Special-Purpose Program Status Registers (xPSR).
- union [CONTROL_Type](#)
Union type to access the Control Registers (CONTROL).

- #define [APSR_N_Pos](#) 31
- #define [APSR_N_Msk](#) (1UL << [APSR_N_Pos](#))
- #define [APSR_Z_Pos](#) 30
- #define [APSR_Z_Msk](#) (1UL << [APSR_Z_Pos](#))
- #define [APSR_C_Pos](#) 29
- #define [APSR_C_Msk](#) (1UL << [APSR_C_Pos](#))
- #define [APSR_V_Pos](#) 28
- #define [APSR_V_Msk](#) (1UL << [APSR_V_Pos](#))
- #define [APSR_Q_Pos](#) 27
- #define [APSR_Q_Msk](#) (1UL << [APSR_Q_Pos](#))
- #define [APSR_GE_Pos](#) 16
- #define [APSR_GE_Msk](#) (0xFUL << [APSR_GE_Pos](#))
- #define [IPSR_ISR_Pos](#) 0
- #define [IPSR_ISR_Msk](#) (0x1FFUL /*<< [IPSR_ISR_Pos*/\)](#)
- #define [xPSR_N_Pos](#) 31
- #define [xPSR_N_Msk](#) (1UL << [xPSR_N_Pos](#))
- #define [xPSR_Z_Pos](#) 30
- #define [xPSR_Z_Msk](#) (1UL << [xPSR_Z_Pos](#))
- #define [xPSR_C_Pos](#) 29
- #define [xPSR_C_Msk](#) (1UL << [xPSR_C_Pos](#))

- #define xPSR_V_Pos 28
- #define xPSR_V_Msk (1UL << xPSR_V_Pos)
- #define xPSR_Q_Pos 27
- #define xPSR_Q_Msk (1UL << xPSR_Q_Pos)
- #define xPSR_IT_Pos 25
- #define xPSR_IT_Msk (3UL << xPSR_IT_Pos)
- #define xPSR_T_Pos 24
- #define xPSR_T_Msk (1UL << xPSR_T_Pos)
- #define xPSR_GE_Pos 16
- #define xPSR_GE_Msk (0xFUL << xPSR_GE_Pos)
- #define xPSR_ISR_Pos 0
- #define xPSR_ISR_Msk (0x1FFUL /*<< xPSR_ISR_Pos*/)
- #define CONTROL_FPCA_Pos 2
- #define CONTROL_FPCA_Msk (1UL << CONTROL_FPCA_Pos)
- #define CONTROL_SPSEL_Pos 1
- #define CONTROL_SPSEL_Msk (1UL << CONTROL_SPSEL_Pos)
- #define CONTROL_nPRIV_Pos 0
- #define CONTROL_nPRIV_Msk (1UL /*<< CONTROL_nPRIV_Pos*/)

5.19.1 Detailed Description

Core Register type definitions.

5.19.2 Macro Definition Documentation

5.19.2.1 APSR_C_Msk

```
#define APSR_C_Msk (1UL << APSR_C_Pos)
```

APSR: C Mask

5.19.2.2 APSR_C_Pos

```
#define APSR_C_Pos 29
```

APSR: C Position

5.19.2.3 APSR_GE_Msk

```
#define APSR_GE_Msk (0xFUL << APSR_GE_Pos)
```

APSR: GE Mask

5.19.2.4 APSR_GE_Pos

```
#define APSR_GE_Pos 16
```

APSR: GE Position

5.19.2.5 APSR_N_Msk

```
#define APSR_N_Msk (1UL << APSR_N_Pos)
```

APSR: N Mask

5.19.2.6 APSR_N_Pos

```
#define APSR_N_Pos 31
```

APSR: N Position

5.19.2.7 APSR_Q_Msk

```
#define APSR_Q_Msk (1UL << APSR_Q_Pos)
```

APSR: Q Mask

5.19.2.8 APSR_Q_Pos

```
#define APSR_Q_Pos 27
```

APSR: Q Position

5.19.2.9 APSR_V_Msk

```
#define APSR_V_Msk (1UL << APSR_V_Pos)
```

APSR: V Mask

5.19.2.10 APSR_V_Pos

```
#define APSR_V_Pos 28
```

APSR: V Position

5.19.2.11 APSR_Z_Msk

```
#define APSR_Z_Msk (1UL << APSR_Z_Pos)
```

APSR: Z Mask

5.19.2.12 APSR_Z_Pos

```
#define APSR_Z_Pos 30
```

APSR: Z Position

5.19.2.13 CONTROL_FPCA_Msk

```
#define CONTROL_FPCA_Msk (1UL << CONTROL_FPCA_Pos)
```

CONTROL: FPCA Mask

5.19.2.14 CONTROL_FPCA_Pos

```
#define CONTROL_FPCA_Pos 2
```

CONTROL: FPCA Position

5.19.2.15 CONTROL_nPRIV_Msk

```
#define CONTROL_nPRIV_Msk (1UL /*<< CONTROL_nPRIV_Pos*/)
```

CONTROL: nPRIV Mask

5.19.2.16 CONTROL_nPRIV_Pos

```
#define CONTROL_nPRIV_Pos 0
```

CONTROL: nPRIV Position

5.19.2.17 CONTROL_SPSEL_Msk

```
#define CONTROL_SPSEL_Msk (1UL << CONTROL_SPSEL_Pos)
```

CONTROL: SPSEL Mask

5.19.2.18 CONTROL_SPSEL_Pos

```
#define CONTROL_SPSEL_Pos 1
```

CONTROL: SPSEL Position

5.19.2.19 IPSR_ISR_Msk

```
#define IPSR_ISR_Msk (0x1FFUL /*<< IPSR_ISR_Pos*/)
```

IPSR: ISR Mask

5.19.2.20 IPSR_ISR_Pos

```
#define IPSR_ISR_Pos 0
```

IPSR: ISR Position

5.19.2.21 xPSR_C_Msk

```
#define xPSR_C_Msk (1UL << xPSR_C_Pos)
```

xPSR: C Mask

5.19.2.22 xPSR_C_Pos

```
#define xPSR_C_Pos 29
```

xPSR: C Position

5.19.2.23 xPSR_GE_Msk

```
#define xPSR_GE_Msk (0xFUL << xPSR_GE_Pos)
```

xPSR: GE Mask

5.19.2.24 xPSR_GE_Pos

```
#define xPSR_GE_Pos 16
```

xPSR: GE Position

5.19.2.25 xPSR_ISR_Msk

```
#define xPSR_ISR_Msk (0x1FFUL /*<< xPSR_ISR_Pos*/)
```

xPSR: ISR Mask

5.19.2.26 xPSR_ISR_Pos

```
#define xPSR_ISR_Pos 0
```

xPSR: ISR Position

5.19.2.27 xPSR_IT_Msk

```
#define xPSR_IT_Msk (3UL << xPSR_IT_Pos)
```

xPSR: IT Mask

5.19.2.28 xPSR_IT_Pos

```
#define xPSR_IT_Pos 25
```

xPSR: IT Position

5.19.2.29 xPSR_N_Msk

```
#define xPSR_N_Msk (1UL << xPSR_N_Pos)
```

xPSR: N Mask

5.19.2.30 xPSR_N_Pos

```
#define xPSR_N_Pos 31
```

xPSR: N Position

5.19.2.31 xPSR_Q_Msk

```
#define xPSR_Q_Msk (1UL << xPSR_Q_Pos)
```

xPSR: Q Mask

5.19.2.32 xPSR_Q_Pos

```
#define xPSR_Q_Pos 27
```

xPSR: Q Position

5.19.2.33 xPSR_T_Msk

```
#define xPSR_T_Msk (1UL << xPSR_T_Pos)
```

xPSR: T Mask

5.19.2.34 xPSR_T_Pos

```
#define xPSR_T_Pos 24
```

xPSR: T Position

5.19.2.35 xPSR_V_Msk

```
#define xPSR_V_Msk (1UL << xPSR_V_Pos)
```

xPSR: V Mask

5.19.2.36 xPSR_V_Pos

```
#define xPSR_V_Pos 28
```

xPSR: V Position

5.19.2.37 xPSR_Z_Msk

```
#define xPSR_Z_Msk (1UL << xPSR_Z_Pos)
```

xPSR: Z Mask

5.19.2.38 xPSR_Z_Pos

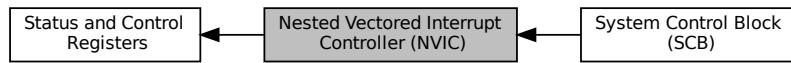
```
#define xPSR_Z_Pos 30
```

xPSR: Z Position

5.20 Nested Vectored Interrupt Controller (NVIC)

Type definitions for the NVIC Registers.

Collaboration diagram for Nested Vectored Interrupt Controller (NVIC):



Modules

- [System Control Block \(SCB\)](#)

Type definitions for the System Control Block Registers.

Classes

- struct [NVIC_Type](#)

Structure type to access the Nested Vectored Interrupt Controller (NVIC).

- #define [NVIC_STIR_INTID_Pos](#) 0
- #define [NVIC_STIR_INTID_Msk](#) (0x1FFUL /*<< [NVIC_STIR_INTID_Pos](#)*/)

5.20.1 Detailed Description

Type definitions for the NVIC Registers.

5.20.2 Macro Definition Documentation

5.20.2.1 NVIC_STIR_INTID_Msk

```
#define NVIC_STIR_INTID_Msk (0x1FFUL /*<< NVIC_STIR_INTID_Pos*/)
```

STIR: INTLINESNUM Mask

5.20.2.2 NVIC_STIR_INTID_Pos

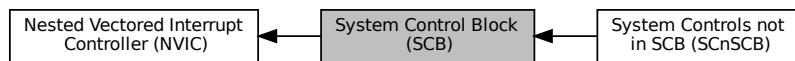
```
#define NVIC_STIR_INTID_Pos 0
```

STIR: INTLINESNUM Position

5.21 System Control Block (SCB)

Type definitions for the System Control Block Registers.

Collaboration diagram for System Control Block (SCB):



Modules

- [System Controls not in SCB \(SCnSCB\)](#)

Type definitions for the System Control and ID Register not in the SCB.

Classes

- struct `SCB_Type`

Structure type to access the System Control Block (SCB).
- `#define SCB_CPUID_IMPLEMENTER_Pos 24`
- `#define SCB_CPUID_IMPLEMENTER_Msk (0xFFUL << SCB_CPUID_IMPLEMENTER_Pos)`
- `#define SCB_CPUID_VARIANT_Pos 20`
- `#define SCB_CPUID_VARIANT_Msk (0xFUL << SCB_CPUID_VARIANT_Pos)`
- `#define SCB_CPUID_ARCHITECTURE_Pos 16`
- `#define SCB_CPUID_ARCHITECTURE_Msk (0xFUL << SCB_CPUID_ARCHITECTURE_Pos)`
- `#define SCB_CPUID_PARTNO_Pos 4`
- `#define SCB_CPUID_PARTNO_Msk (0xFFFFUL << SCB_CPUID_PARTNO_Pos)`
- `#define SCB_CPUID_REVISION_Pos 0`
- `#define SCB_CPUID_REVISION_Msk (0xFUL /*<< SCB_CPUID_REVISION_Pos*/)`
- `#define SCB_ICSR_NMIPENDSET_Pos 31`
- `#define SCB_ICSR_NMIPENDSET_Msk (1UL << SCB_ICSR_NMIPENDSET_Pos)`
- `#define SCB_ICSR_PENDSVSET_Pos 28`
- `#define SCB_ICSR_PENDSVSET_Msk (1UL << SCB_ICSR_PENDSVSET_Pos)`
- `#define SCB_ICSR_PENDSVCLR_Pos 27`
- `#define SCB_ICSR_PENDSVCLR_Msk (1UL << SCB_ICSR_PENDSVCLR_Pos)`
- `#define SCB_ICSR_PENDSTSET_Pos 26`
- `#define SCB_ICSR_PENDSTSET_Msk (1UL << SCB_ICSR_PENDSTSET_Pos)`
- `#define SCB_ICSR_PENDSTCLR_Pos 25`
- `#define SCB_ICSR_PENDSTCLR_Msk (1UL << SCB_ICSR_PENDSTCLR_Pos)`
- `#define SCB_ICSR_ISRPREEMPT_Pos 23`
- `#define SCB_ICSR_ISRPREEMPT_Msk (1UL << SCB_ICSR_ISRPREEMPT_Pos)`
- `#define SCB_ICSR_ISRPENDING_Pos 22`
- `#define SCB_ICSR_ISRPENDING_Msk (1UL << SCB_ICSR_ISRPENDING_Pos)`
- `#define SCB_ICSR_VECTPENDING_Pos 12`
- `#define SCB_ICSR_VECTPENDING_Msk (0x1FFUL << SCB_ICSR_VECTPENDING_Pos)`
- `#define SCB_ICSR_RETTOBASE_Pos 11`
- `#define SCB_ICSR_RETTOBASE_Msk (1UL << SCB_ICSR_RETTOBASE_Pos)`
- `#define SCB_ICSR_VECTACTIVE_Pos 0`
- `#define SCB_ICSR_VECTACTIVE_Msk (0x1FFUL /*<< SCB_ICSR_VECTACTIVE_Pos*/)`
- `#define SCB_VTOR_TBLOFF_Pos 7`
- `#define SCB_VTOR_TBLOFF_Msk (0x1FFFFFFUL << SCB_VTOR_TBLOFF_Pos)`
- `#define SCB_AIRCR_VECTKEY_Pos 16`
- `#define SCB_AIRCR_VECTKEY_Msk (0xFFFFUL << SCB_AIRCR_VECTKEY_Pos)`
- `#define SCB_AIRCR_VECTKEYSTAT_Pos 16`
- `#define SCB_AIRCR_VECTKEYSTAT_Msk (0xFFFFUL << SCB_AIRCR_VECTKEYSTAT_Pos)`
- `#define SCB_AIRCR_ENDIANESS_Pos 15`
- `#define SCB_AIRCR_ENDIANESS_Msk (1UL << SCB_AIRCR_ENDIANESS_Pos)`
- `#define SCB_AIRCR_PRIGROUP_Pos 8`
- `#define SCB_AIRCR_PRIGROUP_Msk (7UL << SCB_AIRCR_PRIGROUP_Pos)`
- `#define SCB_AIRCR_SYSRESETREQ_Pos 2`
- `#define SCB_AIRCR_SYSRESETREQ_Msk (1UL << SCB_AIRCR_SYSRESETREQ_Pos)`
- `#define SCB_AIRCR_VECTCLRACTIVE_Pos 1`
- `#define SCB_AIRCR_VECTCLRACTIVE_Msk (1UL << SCB_AIRCR_VECTCLRACTIVE_Pos)`
- `#define SCB_AIRCR_VECTRESET_Pos 0`
- `#define SCB_AIRCR_VECTRESET_Msk (1UL /*<< SCB_AIRCR_VECTRESET_Pos*/)`
- `#define SCB_SCR_SEVONPEND_Pos 4`
- `#define SCB_SCR_SEVONPEND_Msk (1UL << SCB_SCR_SEVONPEND_Pos)`
- `#define SCB_SCR_SLEEPDEEP_Pos 2`

- #define SCB_SCR_SLEEPDEEP_Msk (1UL << SCB_SCR_SLEEPDEEP_Pos)
- #define SCB_SCR_SLEEPONEXIT_Pos 1
- #define SCB_SCR_SLEEPONEXIT_Msk (1UL << SCB_SCR_SLEEPONEXIT_Pos)
- #define SCB_CCR_STKALIGN_Pos 9
- #define SCB_CCR_STKALIGN_Msk (1UL << SCB_CCR_STKALIGN_Pos)
- #define SCB_CCR_BFHFNIGN_Pos 8
- #define SCB_CCR_BFHFNIGN_Msk (1UL << SCB_CCR_BFHFNIGN_Pos)
- #define SCB_CCR_DIV_0_TRP_Pos 4
- #define SCB_CCR_DIV_0_TRP_Msk (1UL << SCB_CCR_DIV_0_TRP_Pos)
- #define SCB_CCR_UNALIGN_TRP_Pos 3
- #define SCB_CCR_UNALIGN_TRP_Msk (1UL << SCB_CCR_UNALIGN_TRP_Pos)
- #define SCB_CCR_USERSETMPEND_Pos 1
- #define SCB_CCR_USERSETMPEND_Msk (1UL << SCB_CCR_USERSETMPEND_Pos)
- #define SCB_CCR_NONBASETHRDENA_Pos 0
- #define SCB_CCR_NONBASETHRDENA_Msk (1UL /*<< SCB_CCR_NONBASETHRDENA_Pos*/)
- #define SCB_SHCSR_USGFAULTENA_Pos 18
- #define SCB_SHCSR_USGFAULTENA_Msk (1UL << SCB_SHCSR_USGFAULTENA_Pos)
- #define SCB_SHCSR_BUSFAULTENA_Pos 17
- #define SCB_SHCSR_BUSFAULTENA_Msk (1UL << SCB_SHCSR_BUSFAULTENA_Pos)
- #define SCB_SHCSR_MEMFAULTENA_Pos 16
- #define SCB_SHCSR_MEMFAULTENA_Msk (1UL << SCB_SHCSR_MEMFAULTENA_Pos)
- #define SCB_SHCSR_SVCALLPENDED_Pos 15
- #define SCB_SHCSR_SVCALLPENDED_Msk (1UL << SCB_SHCSR_SVCALLPENDED_Pos)
- #define SCB_SHCSR_BUSFAULTPENDED_Pos 14
- #define SCB_SHCSR_BUSFAULTPENDED_Msk (1UL << SCB_SHCSR_BUSFAULTPENDED_Pos)
- #define SCB_SHCSR_MEMFAULTPENDED_Pos 13
- #define SCB_SHCSR_MEMFAULTPENDED_Msk (1UL << SCB_SHCSR_MEMFAULTPENDED_Pos)
- #define SCB_SHCSR_USGFAULTPENDED_Pos 12
- #define SCB_SHCSR_USGFAULTPENDED_Msk (1UL << SCB_SHCSR_USGFAULTPENDED_Pos)
- #define SCB_SHCSR_SYSTICKACT_Pos 11
- #define SCB_SHCSR_SYSTICKACT_Msk (1UL << SCB_SHCSR_SYSTICKACT_Pos)
- #define SCB_SHCSR_PENDSVACT_Pos 10
- #define SCB_SHCSR_PENDSVACT_Msk (1UL << SCB_SHCSR_PENDSVACT_Pos)
- #define SCB_SHCSR_MONITORACT_Pos 8
- #define SCB_SHCSR_MONITORACT_Msk (1UL << SCB_SHCSR_MONITORACT_Pos)
- #define SCB_SHCSR_SVCALLACT_Pos 7
- #define SCB_SHCSR_SVCALLACT_Msk (1UL << SCB_SHCSR_SVCALLACT_Pos)
- #define SCB_SHCSR_USGFAULTACT_Pos 3
- #define SCB_SHCSR_USGFAULTACT_Msk (1UL << SCB_SHCSR_USGFAULTACT_Pos)
- #define SCB_SHCSR_BUSFAULTACT_Pos 1
- #define SCB_SHCSR_BUSFAULTACT_Msk (1UL << SCB_SHCSR_BUSFAULTACT_Pos)
- #define SCB_SHCSR_MEMFAULTACT_Pos 0
- #define SCB_SHCSR_MEMFAULTACT_Msk (1UL /*<< SCB_SHCSR_MEMFAULTACT_Pos*/)
- #define SCB_CFSR_USGFAULTSR_Pos 16
- #define SCB_CFSR_USGFAULTSR_Msk (0xFFFFUL << SCB_CFSR_USGFAULTSR_Pos)
- #define SCB_CFSR_BUSFAULTSR_Pos 8
- #define SCB_CFSR_BUSFAULTSR_Msk (0xFFUL << SCB_CFSR_BUSFAULTSR_Pos)
- #define SCB_CFSR_MEMFAULTSR_Pos 0
- #define SCB_CFSR_MEMFAULTSR_Msk (0xFFUL /*<< SCB_CFSR_MEMFAULTSR_Pos*/)
- #define SCB_HFSR_DEBUGEVT_Pos 31
- #define SCB_HFSR_DEBUGEVT_Msk (1UL << SCB_HFSR_DEBUGEVT_Pos)
- #define SCB_HFSR_FORCED_Pos 30
- #define SCB_HFSR_FORCED_Msk (1UL << SCB_HFSR_FORCED_Pos)
- #define SCB_HFSR_VECTTBL_Pos 1
- #define SCB_HFSR_VECTTBL_Msk (1UL << SCB_HFSR_VECTTBL_Pos)

- #define SCB_DFSR_EXTERNAL_Pos 4
- #define SCB_DFSR_EXTERNAL_Msk (1UL << SCB_DFSR_EXTERNAL_Pos)
- #define SCB_DFSR_VCATCH_Pos 3
- #define SCB_DFSR_VCATCH_Msk (1UL << SCB_DFSR_VCATCH_Pos)
- #define SCB_DFSR_DWTTRAP_Pos 2
- #define SCB_DFSR_DWTTRAP_Msk (1UL << SCB_DFSR_DWTTRAP_Pos)
- #define SCB_DFSR_BKPT_Pos 1
- #define SCB_DFSR_BKPT_Msk (1UL << SCB_DFSR_BKPT_Pos)
- #define SCB_DFSR_HALTED_Pos 0
- #define SCB_DFSR_HALTED_Msk (1UL /*<< SCB_DFSR_HALTED_Pos*/)

5.21.1 Detailed Description

Type definitions for the System Control Block Registers.

5.21.2 Macro Definition Documentation

5.21.2.1 SCB_AIRCR_ENDIANESS_Msk

```
#define SCB_AIRCR_ENDIANESS_Msk (1UL << SCB_AIRCR_ENDIANESS_Pos)
```

SCB AIRCR: ENDIANESS Mask

5.21.2.2 SCB_AIRCR_ENDIANESS_Pos

```
#define SCB_AIRCR_ENDIANESS_Pos 15
```

SCB AIRCR: ENDIANESS Position

5.21.2.3 SCB_AIRCR_PRIGROUP_Msk

```
#define SCB_AIRCR_PRIGROUP_Msk (7UL << SCB_AIRCR_PRIGROUP_Pos)
```

SCB AIRCR: PRIGROUP Mask

5.21.2.4 SCB_AIRCR_PRIGROUP_Pos

```
#define SCB_AIRCR_PRIGROUP_Pos 8
```

SCB AIRCR: PRIGROUP Position

5.21.2.5 SCB_AIRCR_SYSRESETREQ_Msk

```
#define SCB_AIRCR_SYSRESETREQ_Msk (1UL << SCB_AIRCR_SYSRESETREQ_Pos)
```

SCB AIRCR: SYSRESETREQ Mask

5.21.2.6 SCB_AIRCR_SYSRESETREQ_Pos

```
#define SCB_AIRCR_SYSRESETREQ_Pos 2
```

SCB AIRCR: SYSRESETREQ Position

5.21.2.7 SCB_AIRCR_VECTCLRACTIVE_Msk

```
#define SCB_AIRCR_VECTCLRACTIVE_Msk (1UL << SCB_AIRCR_VECTCLRACTIVE_Pos)
```

SCB AIRCR: VECTCLRACTIVE Mask

5.21.2.8 SCB_AIRCR_VECTCLRACTIVE_Pos

```
#define SCB_AIRCR_VECTCLRACTIVE_Pos 1
```

SCB AIRCR: VECTCLRACTIVE Position

5.21.2.9 SCB_AIRCR_VECTKEY_Msk

```
#define SCB_AIRCR_VECTKEY_Msk (0xFFFFUL << SCB_AIRCR_VECTKEY_Pos)
```

SCB AIRCR: VECTKEY Mask

5.21.2.10 SCB_AIRCR_VECTKEY_Pos

```
#define SCB_AIRCR_VECTKEY_Pos 16
```

SCB AIRCR: VECTKEY Position

5.21.2.11 SCB_AIRCR_VECTKEYSTAT_Msk

```
#define SCB_AIRCR_VECTKEYSTAT_Msk (0xFFFFUL << SCB_AIRCR_VECTKEYSTAT_Pos)
```

SCB AIRCR: VECTKEYSTAT Mask

5.21.2.12 SCB_AIRCR_VECTKEYSTAT_Pos

```
#define SCB_AIRCR_VECTKEYSTAT_Pos 16
```

SCB AIRCR: VECTKEYSTAT Position

5.21.2.13 SCB_AIRCR_VECTRESET_Msk

```
#define SCB_AIRCR_VECTRESET_Msk (1UL /*<< SCB_AIRCR_VECTRESET_Pos*/)
```

SCB AIRCR: VECTRESET Mask

5.21.2.14 SCB_AIRCR_VECTRESET_Pos

```
#define SCB_AIRCR_VECTRESET_Pos 0
```

SCB AIRCR: VECTRESET Position

5.21.2.15 SCB_CCR_BFHFNIGN_Msk

```
#define SCB_CCR_BFHFNIGN_Msk (1UL << SCB_CCR_BFHFNIGN_Pos)
```

SCB CCR: BFHFNMIGN Mask

5.21.2.16 SCB_CCR_BFHFNIGN_Pos

```
#define SCB_CCR_BFHFNIGN_Pos 8
```

SCB CCR: BFHFNMIGN Position

5.21.2.17 SCB_CCR_DIV_0_TRP_Msk

```
#define SCB_CCR_DIV_0_TRP_Msk (1UL << SCB_CCR_DIV_0_TRP_Pos)
```

SCB CCR: DIV_0_TRP Mask

5.21.2.18 SCB_CCR_DIV_0_TRP_Pos

```
#define SCB_CCR_DIV_0_TRP_Pos 4
```

SCB CCR: DIV_0_TRP Position

5.21.2.19 SCB_CCR_NONBASETHRDENA_Msk

```
#define SCB_CCR_NONBASETHRDENA_Msk (1UL /*<< SCB_CCR_NONBASETHRDENA_Pos*/)
```

SCB CCR: NONBASETHRDENA Mask

5.21.2.20 SCB_CCR_NONBASETHRDENA_Pos

```
#define SCB_CCR_NONBASETHRDENA_Pos 0
```

SCB CCR: NONBASETHRDENA Position

5.21.2.21 SCB_CCR_STKALIGN_Msk

```
#define SCB_CCR_STKALIGN_Msk (1UL << SCB_CCR_STKALIGN_Pos)
```

SCB CCR: STKALIGN Mask

5.21.2.22 SCB_CCR_STKALIGN_Pos

```
#define SCB_CCR_STKALIGN_Pos 9
```

SCB CCR: STKALIGN Position

5.21.2.23 SCB_CCR_UNALIGN_TRP_Msk

```
#define SCB_CCR_UNALIGN_TRP_Msk (1UL << SCB_CCR_UNALIGN_TRP_Pos)
```

SCB CCR: UNALIGN_TRP Mask

5.21.2.24 SCB_CCR_UNALIGN_TRP_Pos

```
#define SCB_CCR_UNALIGN_TRP_Pos 3
```

SCB CCR: UNALIGN_TRP Position

5.21.2.25 SCB_CCR_USERSETMPEND_Msk

```
#define SCB_CCR_USERSETMPEND_Msk (1UL << SCB_CCR_USERSETMPEND_Pos)
```

SCB CCR: USERSETMPEND Mask

5.21.2.26 SCB_CCR_USERSETMPEND_Pos

```
#define SCB_CCR_USERSETMPEND_Pos 1
```

SCB CCR: USERSETMPEND Position

5.21.2.27 SCB_CFSR_BUSFAULTSR_Msk

```
#define SCB_CFSR_BUSFAULTSR_Msk (0xFFUL << SCB_CFSR_BUSFAULTSR_Pos)
```

SCB CFSR: Bus Fault Status Register Mask

5.21.2.28 SCB_CFSR_BUSFAULTSR_Pos

```
#define SCB_CFSR_BUSFAULTSR_Pos 8
```

SCB CFSR: Bus Fault Status Register Position

5.21.2.29 SCB_CFSR_MEMFAULTSR_Msk

```
#define SCB_CFSR_MEMFAULTSR_Msk (0xFFUL /*<< SCB_CFSR_MEMFAULTSR_Pos*/)
```

SCB CFSR: Memory Manage Fault Status Register Mask

5.21.2.30 SCB_CFSR_MEMFAULTSR_Pos

```
#define SCB_CFSR_MEMFAULTSR_Pos 0
```

SCB CFSR: Memory Manage Fault Status Register Position

5.21.2.31 SCB_CFSR_USGFAULTSR_Msk

```
#define SCB_CFSR_USGFAULTSR_Msk (0xFFFFUL << SCB_CFSR_USGFAULTSR_Pos)
```

SCB CFSR: Usage Fault Status Register Mask

5.21.2.32 SCB_CFSR_USGFAULTSR_Pos

```
#define SCB_CFSR_USGFAULTSR_Pos 16
```

SCB CFSR: Usage Fault Status Register Position

5.21.2.33 SCB_CPUID_ARCHITECTURE_Msk

```
#define SCB_CPUID_ARCHITECTURE_Msk (0xFUL << SCB_CPUID_ARCHITECTURE_Pos)
```

SCB CPUID: ARCHITECTURE Mask

5.21.2.34 SCB_CPUID_ARCHITECTURE_Pos

```
#define SCB_CPUID_ARCHITECTURE_Pos 16
```

SCB CPUID: ARCHITECTURE Position

5.21.2.35 SCB_CPUID_IMPLEMENTER_Msk

```
#define SCB_CPUID_IMPLEMENTER_Msk (0xFFUL << SCB_CPUID_IMPLEMENTER_Pos)
```

SCB CPUID: IMPLEMENTER Mask

5.21.2.36 SCB_CPUID_IMPLEMENTER_Pos

```
#define SCB_CPUID_IMPLEMENTER_Pos 24
```

SCB CPUID: IMPLEMENTER Position

5.21.2.37 SCB_CPUID_PARTNO_Msk

```
#define SCB_CPUID_PARTNO_Msk (0xFFFFUL << SCB_CPUID_PARTNO_Pos)
```

SCB CPUID: PARTNO Mask

5.21.2.38 SCB_CPUID_PARTNO_Pos

```
#define SCB_CPUID_PARTNO_Pos 4
```

SCB CPUID: PARTNO Position

5.21.2.39 SCB_CPUID_REVISION_Msk

```
#define SCB_CPUID_REVISION_Msk (0xFUL /*<< SCB_CPUID_REVISION_Pos*/)
```

SCB CPUID: REVISION Mask

5.21.2.40 SCB_CPUID_REVISION_Pos

```
#define SCB_CPUID_REVISION_Pos 0
```

SCB CPUID: REVISION Position

5.21.2.41 SCB_CPUID_VARIANT_Msk

```
#define SCB_CPUID_VARIANT_Msk (0xFUL << SCB_CPUID_VARIANT_Pos)
```

SCB CPUID: VARIANT Mask

5.21.2.42 SCB_CPUID_VARIANT_Pos

```
#define SCB_CPUID_VARIANT_Pos 20
```

SCB CPUID: VARIANT Position

5.21.2.43 SCB_DFSR_BKPT_Msk

```
#define SCB_DFSR_BKPT_Msk (1UL << SCB_DFSR_BKPT_Pos)
```

SCB DFSR: BKPT Mask

5.21.2.44 SCB_DFSR_BKPT_Pos

```
#define SCB_DFSR_BKPT_Pos 1
```

SCB DFSR: BKPT Position

5.21.2.45 SCB_DFSR_DWTTRAP_Msk

```
#define SCB_DFSR_DWTTRAP_Msk (1UL << SCB_DFSR_DWTTRAP_Pos)
```

SCB DFSR: DWTTRAP Mask

5.21.2.46 SCB_DFSR_DWTTRAP_Pos

```
#define SCB_DFSR_DWTTRAP_Pos 2
```

SCB DFSR: DWTTRAP Position

5.21.2.47 SCB_DFSR_EXTERNAL_Msk

```
#define SCB_DFSR_EXTERNAL_Msk (1UL << SCB_DFSR_EXTERNAL_Pos)
```

SCB DFSR: EXTERNAL Mask

5.21.2.48 SCB_DFSR_EXTERNAL_Pos

```
#define SCB_DFSR_EXTERNAL_Pos 4
```

SCB DFSR: EXTERNAL Position

5.21.2.49 SCB_DFSR_HALTED_Msk

```
#define SCB_DFSR_HALTED_Msk (1UL /*<< SCB_DFSR_HALTED_Pos*/)
```

SCB DFSR: HALTED Mask

5.21.2.50 SCB_DFSR_HALTED_Pos

```
#define SCB_DFSR_HALTED_Pos 0
```

SCB DFSR: HALTED Position

5.21.2.51 SCB_DFSR_VCATCH_Msk

```
#define SCB_DFSR_VCATCH_Msk (1UL << SCB_DFSR_VCATCH_Pos)
```

SCB DFSR: VCATCH Mask

5.21.2.52 SCB_DFSR_VCATCH_Pos

```
#define SCB_DFSR_VCATCH_Pos 3
```

SCB DFSR: VCATCH Position

5.21.2.53 SCB_HFSR_DEBUGEV_T_Msk

```
#define SCB_HFSR_DEBUGEVT_Msk (1UL << SCB_HFSR_DEBUGEVT_Pos)
```

SCB HFSR: DEBUGEV_T Mask

5.21.2.54 SCB_HFSR_DEBUGEV_T_Pos

```
#define SCB_HFSR_DEBUGEVT_Pos 31
```

SCB HFSR: DEBUGEV_T Position

5.21.2.55 SCB_HFSR_FORCED_Msk

```
#define SCB_HFSR_FORCED_Msk (1UL << SCB_HFSR_FORCED_Pos)
```

SCB HFSR: FORCED Mask

5.21.2.56 SCB_HFSR_FORCED_Pos

```
#define SCB_HFSR_FORCED_Pos 30
```

SCB HFSR: FORCED Position

5.21.2.57 SCB_HFSR_VECTTBL_Msk

```
#define SCB_HFSR_VECTTBL_Msk (1UL << SCB_HFSR_VECTTBL_Pos)
```

SCB HFSR: VECTTBL Mask

5.21.2.58 SCB_HFSR_VECTTBL_Pos

```
#define SCB_HFSR_VECTTBL_Pos 1
```

SCB HFSR: VECTTBL Position

5.21.2.59 SCB_ICSR_ISRPENDING_Msk

```
#define SCB_ICSR_ISRPENDING_Msk (1UL << SCB_ICSR_ISRPENDING_Pos)
```

SCB ICSR: ISRPENDING Mask

5.21.2.60 SCB_ICSR_ISRPENDING_Pos

```
#define SCB_ICSR_ISRPENDING_Pos 22
```

SCB ICSR: ISRPENDING Position

5.21.2.61 SCB_ICSR_ISRPREEMPT_Msk

```
#define SCB_ICSR_ISRPREEMPT_Msk (1UL << SCB_ICSR_ISRPREEMPT_Pos)
```

SCB ICSR: ISRPREEMPT Mask

5.21.2.62 SCB_ICSR_ISRPREEMPT_Pos

```
#define SCB_ICSR_ISRPREEMPT_Pos 23
```

SCB ICSR: ISRPREEMPT Position

5.21.2.63 SCB_ICSR_NMIPENDSET_Msk

```
#define SCB_ICSR_NMIPENDSET_Msk (1UL << SCB_ICSR_NMIPENDSET_Pos)
```

SCB ICSR: NMIPENDSET Mask

5.21.2.64 SCB_ICSR_NMIPENDSET_Pos

```
#define SCB_ICSR_NMIPENDSET_Pos 31
```

SCB ICSR: NMIPENDSET Position

5.21.2.65 SCB_ICSR_PENDSTCLR_Msk

```
#define SCB_ICSR_PENDSTCLR_Msk (1UL << SCB_ICSR_PENDSTCLR_Pos)
```

SCB ICSR: PENDSTCLR Mask

5.21.2.66 SCB_ICSR_PENDSTCLR_Pos

```
#define SCB_ICSR_PENDSTCLR_Pos 25
```

SCB ICSR: PENDSTCLR Position

5.21.2.67 SCB_ICSR_PENDSTSET_Msk

```
#define SCB_ICSR_PENDSTSET_Msk (1UL << SCB_ICSR_PENDSTSET_Pos)
```

SCB ICSR: PENDSTSET Mask

5.21.2.68 SCB_ICSR_PENDSTSET_Pos

```
#define SCB_ICSR_PENDSTSET_Pos 26
```

SCB ICSR: PENDSTSET Position

5.21.2.69 SCB_ICSR_PENDSVCLR_Msk

```
#define SCB_ICSR_PENDSVCLR_Msk (1UL << SCB_ICSR_PENDSVCLR_Pos)
```

SCB ICSR: PENDSVCLR Mask

5.21.2.70 SCB_ICSR_PENDSVCLR_Pos

```
#define SCB_ICSR_PENDSVCLR_Pos 27
```

SCB ICSR: PENDSVCLR Position

5.21.2.71 SCB_ICSR_PENDSVSET_Msk

```
#define SCB_ICSR_PENDSVSET_Msk (1UL << SCB_ICSR_PENDSVSET_Pos)
```

SCB ICSR: PENDSVSET Mask

5.21.2.72 SCB_ICSR_PENDSVSET_Pos

```
#define SCB_ICSR_PENDSVSET_Pos 28
```

SCB ICSR: PENDSVSET Position

5.21.2.73 SCB_ICSR_RETTOBASE_Msk

```
#define SCB_ICSR_RETTOBASE_Msk (1UL << SCB_ICSR_RETTOBASE_Pos)
```

SCB ICSR: RETTOBASE Mask

5.21.2.74 SCB_ICSR_RETTOBASE_Pos

```
#define SCB_ICSR_RETTOBASE_Pos 11
```

SCB ICSR: RETTOBASE Position

5.21.2.75 SCB_ICSR_VECTACTIVE_Msk

```
#define SCB_ICSR_VECTACTIVE_Msk (0x1FFUL /*<< SCB_ICSR_VECTACTIVE_Pos*/)
```

SCB ICSR: VECTACTIVE Mask

5.21.2.76 SCB_ICSR_VECTACTIVE_Pos

```
#define SCB_ICSR_VECTACTIVE_Pos 0
```

SCB ICSR: VECTACTIVE Position

5.21.2.77 SCB_ICSR_VECTPENDING_Msk

```
#define SCB_ICSR_VECTPENDING_Msk (0x1FFUL << SCB_ICSR_VECTPENDING_Pos)
```

SCB ICSR: VECTPENDING Mask

5.21.2.78 SCB_ICSR_VECTPENDING_Pos

```
#define SCB_ICSR_VECTPENDING_Pos 12
```

SCB ICSR: VECTPENDING Position

5.21.2.79 SCB_SCR_SEVONPEND_Msk

```
#define SCB_SCR_SEVONPEND_Msk (1UL << SCB_SCR_SEVONPEND_Pos)
```

SCB SCR: SEVONPEND Mask

5.21.2.80 SCB_SCR_SEVONPEND_Pos

```
#define SCB_SCR_SEVONPEND_Pos 4
```

SCB SCR: SEVONPEND Position

5.21.2.81 SCB_SCR_SLEEPDEEP_Msk

```
#define SCB_SCR_SLEEPDEEP_Msk (1UL << SCB_SCR_SLEEPDEEP_Pos)
```

SCB SCR: SLEEPDEEP Mask

5.21.2.82 SCB_SCR_SLEEPDEEP_Pos

```
#define SCB_SCR_SLEEPDEEP_Pos 2
```

SCB SCR: SLEEPDEEP Position

5.21.2.83 SCB_SCR_SLEEPONEXIT_Msk

```
#define SCB_SCR_SLEEPONEXIT_Msk (1UL << SCB_SCR_SLEEPONEXIT_Pos)
```

SCB SCR: SLEEPONEXIT Mask

5.21.2.84 SCB_SCR_SLEEPONEXIT_Pos

```
#define SCB_SCR_SLEEPONEXIT_Pos 1
```

SCB SCR: SLEEPONEXIT Position

5.21.2.85 SCB_SHCSR_BUSFAULTACT_Msk

```
#define SCB_SHCSR_BUSFAULTACT_Msk (1UL << SCB_SHCSR_BUSFAULTACT_Pos)
```

SCB SHCSR: BUSFAULTACT Mask

5.21.2.86 SCB_SHCSR_BUSFAULTACT_Pos

```
#define SCB_SHCSR_BUSFAULTACT_Pos 1
```

SCB SHCSR: BUSFAULTACT Position

5.21.2.87 SCB_SHCSR_BUSFAULTENA_Msk

```
#define SCB_SHCSR_BUSFAULTENA_Msk (1UL << SCB_SHCSR_BUSFAULTENA_Pos)
```

SCB SHCSR: BUSFAULTENA Mask

5.21.2.88 SCB_SHCSR_BUSFAULTENA_Pos

```
#define SCB_SHCSR_BUSFAULTENA_Pos 17
```

SCB SHCSR: BUSFAULTENA Position

5.21.2.89 SCB_SHCSR_BUSFAULTPENDED_Msk

```
#define SCB_SHCSR_BUSFAULTPENDED_Msk (1UL << SCB_SHCSR_BUSFAULTPENDED_Pos)
```

SCB SHCSR: BUSFAULTPENDED Mask

5.21.2.90 SCB_SHCSR_BUSFAULTPENDED_Pos

```
#define SCB_SHCSR_BUSFAULTPENDED_Pos 14
```

SCB SHCSR: BUSFAULTPENDED Position

5.21.2.91 SCB_SHCSR_MEMFAULTACT_Msk

```
#define SCB_SHCSR_MEMFAULTACT_Msk (1UL /*<< SCB_SHCSR_MEMFAULTACT_Pos*/)
```

SCB SHCSR: MEMFAULTACT Mask

5.21.2.92 SCB_SHCSR_MEMFAULTACT_Pos

```
#define SCB_SHCSR_MEMFAULTACT_Pos 0
```

SCB SHCSR: MEMFAULTACT Position

5.21.2.93 SCB_SHCSR_MEMFAULTENA_Msk

```
#define SCB_SHCSR_MEMFAULTENA_Msk (1UL << SCB_SHCSR_MEMFAULTENA_Pos)
```

SCB SHCSR: MEMFAULTENA Mask

5.21.2.94 SCB_SHCSR_MEMFAULTENA_Pos

```
#define SCB_SHCSR_MEMFAULTENA_Pos 16
```

SCB SHCSR: MEMFAULTENA Position

5.21.2.95 SCB_SHCSR_MEMFAULTPENDED_Msk

```
#define SCB_SHCSR_MEMFAULTPENDED_Msk (1UL << SCB_SHCSR_MEMFAULTPENDED_Pos)
```

SCB SHCSR: MEMFAULTPENDED Mask

5.21.2.96 SCB_SHCSR_MEMFAULTPENDED_Pos

```
#define SCB_SHCSR_MEMFAULTPENDED_Pos 13
```

SCB SHCSR: MEMFAULTPENDED Position

5.21.2.97 SCB_SHCSR_MONITORACT_Msk

```
#define SCB_SHCSR_MONITORACT_Msk (1UL << SCB_SHCSR_MONITORACT_Pos)
```

SCB SHCSR: MONITORACT Mask

5.21.2.98 SCB_SHCSR_MONITORACT_Pos

```
#define SCB_SHCSR_MONITORACT_Pos 8
```

SCB SHCSR: MONITORACT Position

5.21.2.99 SCB_SHCSR_PENDSVACT_Msk

```
#define SCB_SHCSR_PENDSVACT_Msk (1UL << SCB_SHCSR_PENDSVACT_Pos)
```

SCB SHCSR: PENDSVACT Mask

5.21.2.100 SCB_SHCSR_PENDSVACT_Pos

```
#define SCB_SHCSR_PENDSVACT_Pos 10
```

SCB SHCSR: PENDSVACT Position

5.21.2.101 SCB_SHCSR_SVCALLACT_Msk

```
#define SCB_SHCSR_SVCALLACT_Msk (1UL << SCB_SHCSR_SVCALLACT_Pos)
```

SCB SHCSR: SVCALLACT Mask

5.21.2.102 SCB_SHCSR_SVCALLACT_Pos

```
#define SCB_SHCSR_SVCALLACT_Pos 7
```

SCB SHCSR: SVCALLACT Position

5.21.2.103 SCB_SHCSR_SVCALLPENDED_Msk

```
#define SCB_SHCSR_SVCALLPENDED_Msk (1UL << SCB_SHCSR_SVCALLPENDED_Pos)
```

SCB SHCSR: SVCALLPENDED Mask

5.21.2.104 SCB_SHCSR_SVCALLPENDED_Pos

```
#define SCB_SHCSR_SVCALLPENDED_Pos 15
```

SCB SHCSR: SVCALLPENDED Position

5.21.2.105 SCB_SHCSR_SYSTICKACT_Msk

```
#define SCB_SHCSR_SYSTICKACT_Msk (1UL << SCB_SHCSR_SYSTICKACT_Pos)
```

SCB SHCSR: SYSTICKACT Mask

5.21.2.106 SCB_SHCSR_SYSTICKACT_Pos

```
#define SCB_SHCSR_SYSTICKACT_Pos 11
```

SCB SHCSR: SYSTICKACT Position

5.21.2.107 SCB_SHCSR_USGFAULTACT_Msk

```
#define SCB_SHCSR_USGFAULTACT_Msk (1UL << SCB_SHCSR_USGFAULTACT_Pos)
```

SCB SHCSR: USGFAULTACT Mask

5.21.2.108 SCB_SHCSR_USGFAULTACT_Pos

```
#define SCB_SHCSR_USGFAULTACT_Pos 3
```

SCB SHCSR: USGFAULTACT Position

5.21.2.109 SCB_SHCSR_USGFAULTENA_Msk

```
#define SCB_SHCSR_USGFAULTENA_Msk (1UL << SCB_SHCSR_USGFAULTENA_Pos)
```

SCB SHCSR: USGFAULTENA Mask

5.21.2.110 SCB_SHCSR_USGFAULTENA_Pos

```
#define SCB_SHCSR_USGFAULTENA_Pos 18
```

SCB SHCSR: USGFAULTENA Position

5.21.2.111 SCB_SHCSR_USGFAULTPENDED_Msk

```
#define SCB_SHCSR_USGFAULTPENDED_Msk (1UL << SCB_SHCSR_USGFAULTPENDED_Pos)
```

SCB SHCSR: USGFAULTPENDED Mask

5.21.2.112 SCB_SHCSR_USGFAULTPENDED_Pos

```
#define SCB_SHCSR_USGFAULTPENDED_Pos 12
```

SCB SHCSR: USGFAULTPENDED Position

5.21.2.113 SCB_VTOR_TBLOFF_Msk

```
#define SCB_VTOR_TBLOFF_Msk (0x1FFFFFFFUL << SCB_VTOR_TBLOFF_Pos)
```

SCB VTOR: TBLOFF Mask

5.21.2.114 SCB_VTOR_TBLOFF_Pos

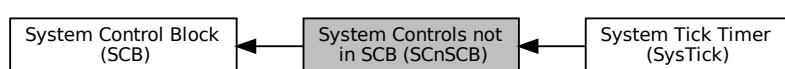
```
#define SCB_VTOR_TBLOFF_Pos 7
```

SCB VTOR: TBLOFF Position

5.22 System Controls not in SCB (SCnSCB)

Type definitions for the System Control and ID Register not in the SCB.

Collaboration diagram for System Controls not in SCB (SCnSCB):



Modules

- [System Tick Timer \(SysTick\)](#)

Type definitions for the System Timer Registers.

Classes

- struct [SCnSCB_Type](#)

Structure type to access the System Control and ID Register not in the SCB.

- `#define SCnSCB_ICTR_INTLINESNUM_Pos 0`
- `#define SCnSCB_ICTR_INTLINESNUM_Msk (0xFUL /*<< SCnSCB_ICTR_INTLINESNUM_Pos*/)`
- `#define SCnSCB_ACTLR_DISOOFP_Pos 9`
- `#define SCnSCB_ACTLR_DISOOFP_Msk (1UL << SCnSCB_ACTLR_DISOOFP_Pos)`
- `#define SCnSCB_ACTLR_DISFPCA_Pos 8`
- `#define SCnSCB_ACTLR_DISFPCA_Msk (1UL << SCnSCB_ACTLR_DISFPCA_Pos)`
- `#define SCnSCB_ACTLR_DISFOLD_Pos 2`
- `#define SCnSCB_ACTLR_DISFOLD_Msk (1UL << SCnSCB_ACTLR_DISFOLD_Pos)`
- `#define SCnSCB_ACTLR_DISDEFWBUF_Pos 1`
- `#define SCnSCB_ACTLR_DISDEFWBUF_Msk (1UL << SCnSCB_ACTLR_DISDEFWBUF_Pos)`
- `#define SCnSCB_ACTLR_DISMCYCINT_Pos 0`
- `#define SCnSCB_ACTLR_DISMCYCINT_Msk (1UL /*<< SCnSCB_ACTLR_DISMCYCINT_Pos*/)`

5.22.1 Detailed Description

Type definitions for the System Control and ID Register not in the SCB.

5.22.2 Macro Definition Documentation

5.22.2.1 SCnSCB_ACTLR_DISDEFWBUF_Msk

```
#define SCnSCB_ACTLR_DISDEFWBUF_Msk (1UL << SCnSCB_ACTLR_DISDEFWBUF_Pos)
```

ACTLR: DISDEFWBUF Mask

5.22.2.2 SCnSCB_ACTLR_DISDEFWBUF_Pos

```
#define SCnSCB_ACTLR_DISDEFWBUF_Pos 1
```

ACTLR: DISDEFWBUF Position

5.22.2.3 SCnSCB_ACTLR_DISFOLD_Msk

```
#define SCnSCB_ACTLR_DISFOLD_Msk (1UL << SCnSCB_ACTLR_DISFOLD_Pos)
```

ACTLR: DISFOLD Mask

5.22.2.4 SCnSCB_ACTLR_DISFOLD_Pos

```
#define SCnSCB_ACTLR_DISFOLD_Pos 2
```

ACTLR: DISFOLD Position

5.22.2.5 SCnSCB_ACTLR_DISFPCA_Msk

```
#define SCnSCB_ACTLR_DISFPCA_Msk (1UL << SCnSCB_ACTLR_DISFPCA_Pos)
```

ACTLR: DISFPCA Mask

5.22.2.6 SCnSCB_ACTLR_DISFPCA_Pos

```
#define SCnSCB_ACTLR_DISFPCA_Pos 8
```

ACTLR: DISFPCA Position

5.22.2.7 SCnSCB_ACTLR_DISMCYCINT_Msk

```
#define SCnSCB_ACTLR_DISMCYCINT_Msk (1UL /*<< SCnSCB_ACTLR_DISMCYCINT_Pos*/)
```

ACTLR: DISMCYCINT Mask

5.22.2.8 SCnSCB_ACTLR_DISMCYCINT_Pos

```
#define SCnSCB_ACTLR_DISMCYCINT_Pos 0
```

ACTLR: DISMCYCINT Position

5.22.2.9 SCnSCB_ACTLR_DISOOFP_Msk

```
#define SCnSCB_ACTLR_DISOOFP_Msk (1UL << SCnSCB_ACTLR_DISOOFP_Pos)
```

ACTLR: DISOOFP Mask

5.22.2.10 SCnSCB_ACTLR_DISOOFP_Pos

```
#define SCnSCB_ACTLR_DISOOFP_Pos 9
```

ACTLR: DISOOFP Position

5.22.2.11 SCnSCB_ICTR_INTLINESNUM_Msk

```
#define SCnSCB_ICTR_INTLINESNUM_Msk (0xFUL /*<< SCnSCB_ICTR_INTLINESNUM_Pos*/)
```

ICTR: INTLINESNUM Mask

5.22.2.12 SCnSCB_ICTR_INTLINESNUM_Pos

```
#define SCnSCB_ICTR_INTLINESNUM_Pos 0
```

ICTR: INTLINESNUM Position

5.23 System Tick Timer (SysTick)

Type definitions for the System Timer Registers.

Collaboration diagram for System Tick Timer (SysTick):



Modules

- Instrumentation Trace Macrocell (ITM)

Type definitions for the Instrumentation Trace Macrocell (ITM)

Classes

- struct [SysTick_Type](#)

Structure type to access the System Timer (SysTick).

- #define [SysTick_CTRL_COUNTFLAG_Pos](#) 16
- #define [SysTick_CTRL_COUNTFLAG_Msk](#) (1UL << [SysTick_CTRL_COUNTFLAG_Pos](#))
- #define [SysTick_CTRL_CLKSOURCE_Pos](#) 2
- #define [SysTick_CTRL_CLKSOURCE_Msk](#) (1UL << [SysTick_CTRL_CLKSOURCE_Pos](#))
- #define [SysTick_CTRL_TICKINT_Pos](#) 1
- #define [SysTick_CTRL_TICKINT_Msk](#) (1UL << [SysTick_CTRL_TICKINT_Pos](#))
- #define [SysTick_CTRL_ENABLE_Pos](#) 0
- #define [SysTick_CTRL_ENABLE_Msk](#) (1UL /*<< [SysTick_CTRL_ENABLE_Pos*/\)](#)
- #define [SysTick_LOAD_RELOAD_Pos](#) 0
- #define [SysTick_LOAD_RELOAD_Msk](#) (0xFFFFFFFFUL /*<< [SysTick_LOAD_RELOAD_Pos*/\)](#)
- #define [SysTick_VAL_CURRENT_Pos](#) 0
- #define [SysTick_VAL_CURRENT_Msk](#) (0xFFFFFFFFUL /*<< [SysTick_VAL_CURRENT_Pos*/\)](#)
- #define [SysTick_CALIB_NOREF_Pos](#) 31
- #define [SysTick_CALIB_NOREF_Msk](#) (1UL << [SysTick_CALIB_NOREF_Pos](#))
- #define [SysTick_CALIB_SKEW_Pos](#) 30
- #define [SysTick_CALIB_SKEW_Msk](#) (1UL << [SysTick_CALIB_SKEW_Pos](#))
- #define [SysTick_CALIB_TENMS_Pos](#) 0
- #define [SysTick_CALIB_TENMS_Msk](#) (0xFFFFFFFFUL /*<< [SysTick_CALIB_TENMS_Pos*/\)](#)

5.23.1 Detailed Description

Type definitions for the System Timer Registers.

5.23.2 Macro Definition Documentation

5.23.2.1 SysTick_CALIB_NOREF_Msk

```
#define SysTick_CALIB_NOREF_Msk (1UL << SysTick_CALIB_NOREF_Pos)
```

SysTick CALIB: NOREF Mask

5.23.2.2 SysTick_CALIB_NOREF_Pos

```
#define SysTick_CALIB_NOREF_Pos 31
```

SysTick CALIB: NOREF Position

5.23.2.3 SysTick_CALIB_SKEW_Msk

```
#define SysTick_CALIB_SKEW_Msk (1UL << SysTick_CALIB_SKEW_Pos)
```

SysTick CALIB: SKEW Mask

5.23.2.4 SysTick_CALIB_SKEW_Pos

```
#define SysTick_CALIB_SKEW_Pos 30
```

SysTick CALIB: SKEW Position

5.23.2.5 SysTick_CALIB_TENMS_Msk

```
#define SysTick_CALIB_TENMS_Msk (0xFFFFFUL /*<< SysTick_CALIB_TENMS_Pos*/)
```

SysTick CALIB: TENMS Mask

5.23.2.6 SysTick_CALIB_TENMS_Pos

```
#define SysTick_CALIB_TENMS_Pos 0
```

SysTick CALIB: TENMS Position

5.23.2.7 SysTick_CTRL_CLKSOURCE_Msk

```
#define SysTick_CTRL_CLKSOURCE_Msk (1UL << SysTick_CTRL_CLKSOURCE_Pos)
```

SysTick CTRL: CLKSOURCE Mask

5.23.2.8 SysTick_CTRL_CLKSOURCE_Pos

```
#define SysTick_CTRL_CLKSOURCE_Pos 2
```

SysTick CTRL: CLKSOURCE Position

5.23.2.9 SysTick_CTRL_COUNTFLAG_Msk

```
#define SysTick_CTRL_COUNTFLAG_Msk (1UL << SysTick_CTRL_COUNTFLAG_Pos)
```

SysTick CTRL: COUNTFLAG Mask

5.23.2.10 SysTick_CTRL_COUNTFLAG_Pos

```
#define SysTick_CTRL_COUNTFLAG_Pos 16
```

SysTick CTRL: COUNTFLAG Position

5.23.2.11 SysTick_CTRL_ENABLE_Msk

```
#define SysTick_CTRL_ENABLE_Msk (1UL /*<< SysTick_CTRL_ENABLE_Pos*/)
```

SysTick CTRL: ENABLE Mask

5.23.2.12 SysTick_CTRL_ENABLE_Pos

```
#define SysTick_CTRL_ENABLE_Pos 0
```

SysTick CTRL: ENABLE Position

5.23.2.13 SysTick_CTRL_TICKINT_Msk

```
#define SysTick_CTRL_TICKINT_Msk (1UL << SysTick_CTRL_TICKINT_Pos)
```

SysTick CTRL: TICKINT Mask

5.23.2.14 SysTick_CTRL_TICKINT_Pos

```
#define SysTick_CTRL_TICKINT_Pos 1
```

SysTick CTRL: TICKINT Position

5.23.2.15 SysTick_LOAD_RELOAD_Msk

```
#define SysTick_LOAD_RELOAD_Msk (0xFFFFFFFFUL /*<< SysTick_LOAD_RELOAD_Pos*/)
```

SysTick LOAD: RELOAD Mask

5.23.2.16 SysTick_LOAD_RELOAD_Pos

```
#define SysTick_LOAD_RELOAD_Pos 0
```

SysTick LOAD: RELOAD Position

5.23.2.17 SysTick_VAL_CURRENT_Msk

```
#define SysTick_VAL_CURRENT_Msk (0xFFFFFFFFUL /*<< SysTick_VAL_CURRENT_Pos*/)
```

SysTick VAL: CURRENT Mask

5.23.2.18 SysTick_VAL_CURRENT_Pos

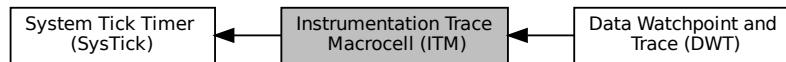
```
#define SysTick_VAL_CURRENT_Pos 0
```

SysTick VAL: CURRENT Position

5.24 Instrumentation Trace Macrocell (ITM)

Type definitions for the Instrumentation Trace Macrocell (ITM)

Collaboration diagram for Instrumentation Trace Macrocell (ITM):



Modules

- [Data Watchpoint and Trace \(DWT\)](#)

Type definitions for the Data Watchpoint and Trace (DWT)

Classes

- struct `ITM_Type`
Structure type to access the Instrumentation Trace Macrocell Register (ITM).
- `#define ITM_TPR_PRIVMASK_Pos 0`
- `#define ITM_TPR_PRIVMASK_Msk (0xFUL /*<< ITM_TPR_PRIVMASK_Pos*/)`
- `#define ITM_TCR_BUSY_Pos 23`
- `#define ITM_TCR_BUSY_Msk (1UL << ITM_TCR_BUSY_Pos)`
- `#define ITM_TCR_TraceBusID_Pos 16`
- `#define ITM_TCR_TraceBusID_Msk (0x7FUL << ITM_TCR_TraceBusID_Pos)`
- `#define ITM_TCR_GTSFREQ_Pos 10`
- `#define ITM_TCR_GTSFREQ_Msk (3UL << ITM_TCR_GTSFREQ_Pos)`
- `#define ITM_TCR_TSPrescale_Pos 8`
- `#define ITM_TCR_TSPrescale_Msk (3UL << ITM_TCR_TSPrescale_Pos)`
- `#define ITM_TCR_SWOENA_Pos 4`
- `#define ITM_TCR_SWOENA_Msk (1UL << ITM_TCR_SWOENA_Pos)`
- `#define ITM_TCR_DWTENA_Pos 3`
- `#define ITM_TCR_DWTENA_Msk (1UL << ITM_TCR_DWTENA_Pos)`
- `#define ITM_TCR_SYNCENA_Pos 2`
- `#define ITM_TCR_SYNCENA_Msk (1UL << ITM_TCR_SYNCENA_Pos)`
- `#define ITM_TCR_TSENA_Pos 1`
- `#define ITM_TCR_TSENA_Msk (1UL << ITM_TCR_TSENA_Pos)`
- `#define ITM_TCR_ITMENA_Pos 0`
- `#define ITM_TCR_ITMENA_Msk (1UL /*<< ITM_TCR_ITMENA_Pos*/)`
- `#define ITM_IWR_ATVALIDM_Pos 0`
- `#define ITM_IWR_ATVALIDM_Msk (1UL /*<< ITM_IWR_ATVALIDM_Pos*/)`
- `#define ITM_IRR_ATREADYM_Pos 0`
- `#define ITM_IRR_ATREADYM_Msk (1UL /*<< ITM_IRR_ATREADYM_Pos*/)`
- `#define ITM_IMCR_INTEGRATION_Pos 0`
- `#define ITM_IMCR_INTEGRATION_Msk (1UL /*<< ITM_IMCR_INTEGRATION_Pos*/)`
- `#define ITM_LSR_ByteAcc_Pos 2`
- `#define ITM_LSR_ByteAcc_Msk (1UL << ITM_LSR_ByteAcc_Pos)`
- `#define ITM_LSR_Access_Pos 1`
- `#define ITM_LSR_Access_Msk (1UL << ITM_LSR_Access_Pos)`
- `#define ITM_LSR_Present_Pos 0`
- `#define ITM_LSR_Present_Msk (1UL /*<< ITM_LSR_Present_Pos*/)`

5.24.1 Detailed Description

Type definitions for the Instrumentation Trace Macrocell (ITM)

5.24.2 Macro Definition Documentation

5.24.2.1 ITM_IMCR_INTEGRATION_Msk

```
#define ITM_IMCR_INTEGRATION_Msk (1UL /*<< ITM_IMCR_INTEGRATION_Pos*/)
```

ITM IMCR: INTEGRATION Mask

5.24.2.2 ITM_IMCR_INTEGRATION_Pos

```
#define ITM_IMCR_INTEGRATION_Pos 0
```

ITM IMCR: INTEGRATION Position

5.24.2.3 ITM_IRR_ATREADYM_Msk

```
#define ITM_IRR_ATREADYM_Msk (1UL /*<< ITM_IRR_ATREADYM_Pos*/)
```

ITM IRR: ATREADYM Mask

5.24.2.4 ITM_IRR_ATREADYM_Pos

```
#define ITM_IRR_ATREADYM_Pos 0
```

ITM IRR: ATREADYM Position

5.24.2.5 ITM_IWR_ATVALIDM_Msk

```
#define ITM_IWR_ATVALIDM_Msk (1UL /*<< ITM_IWR_ATVALIDM_Pos*/)
```

ITM IWR: ATVALIDM Mask

5.24.2.6 ITM_IWR_ATVALIDM_Pos

```
#define ITM_IWR_ATVALIDM_Pos 0
```

ITM IWR: ATVALIDM Position

5.24.2.7 ITM_LSR_Access_Msk

```
#define ITM_LSR_Access_Msk (1UL << ITM_LSR_Access_Pos)
```

ITM LSR: Access Mask

5.24.2.8 ITM_LSR_Access_Pos

```
#define ITM_LSR_Access_Pos 1
```

ITM LSR: Access Position

5.24.2.9 ITM_LSR_BytAcc_Msk

```
#define ITM_LSR_BytAcc_Msk (1UL << ITM_LSR_BytAcc_Pos)
```

ITM LSR: ByteAcc Mask

5.24.2.10 ITM_LSR_ByteAcc_Pos

```
#define ITM_LSR_ByteAcc_Pos 2
```

ITM LSR: ByteAcc Position

5.24.2.11 ITM_LSR_Present_Msk

```
#define ITM_LSR_Present_Msk (1UL /*<< ITM_LSR_Present_Pos*/)
```

ITM LSR: Present Mask

5.24.2.12 ITM_LSR_Present_Pos

```
#define ITM_LSR_Present_Pos 0
```

ITM LSR: Present Position

5.24.2.13 ITM_TCR_BUSY_Msk

```
#define ITM_TCR_BUSY_Msk (1UL << ITM_TCR_BUSY_Pos)
```

ITM TCR: BUSY Mask

5.24.2.14 ITM_TCR_BUSY_Pos

```
#define ITM_TCR_BUSY_Pos 23
```

ITM TCR: BUSY Position

5.24.2.15 ITM_TCR_DWTENA_Msk

```
#define ITM_TCR_DWTENA_Msk (1UL << ITM_TCR_DWTENA_Pos)
```

ITM TCR: DWTENA Mask

5.24.2.16 ITM_TCR_DWTENA_Pos

```
#define ITM_TCR_DWTENA_Pos 3
```

ITM TCR: DWTENA Position

5.24.2.17 ITM_TCR_GTSFREQ_Msk

```
#define ITM_TCR_GTSFREQ_Msk (3UL << ITM_TCR_GTSFREQ_Pos)
```

ITM TCR: Global timestamp frequency Mask

5.24.2.18 ITM_TCR_GTSFREQ_Pos

```
#define ITM_TCR_GTSFREQ_Pos 10
```

ITM TCR: Global timestamp frequency Position

5.24.2.19 ITM_TCR_ITMENA_Msk

```
#define ITM_TCR_ITMENA_Msk (1UL /*<< ITM_TCR_ITMENA_Pos*/)
```

ITM TCR: ITM Enable bit Mask

5.24.2.20 ITM_TCR_ITMENA_Pos

```
#define ITM_TCR_ITMENA_Pos 0
```

ITM TCR: ITM Enable bit Position

5.24.2.21 ITM_TCR_SWOENA_Msk

```
#define ITM_TCR_SWOENA_Msk (1UL << ITM_TCR_SWOENA_Pos)
```

ITM TCR: SWOENA Mask

5.24.2.22 ITM_TCR_SWOENA_Pos

```
#define ITM_TCR_SWOENA_Pos 4
```

ITM TCR: SWOENA Position

5.24.2.23 ITM_TCR_SYNCENA_Msk

```
#define ITM_TCR_SYNCENA_Msk (1UL << ITM_TCR_SYNCENA_Pos)
```

ITM TCR: SYNCENA Mask

5.24.2.24 ITM_TCR_SYNCENA_Pos

```
#define ITM_TCR_SYNCENA_Pos 2
```

ITM TCR: SYNCENA Position

5.24.2.25 ITM_TCR_TraceBusID_Msk

```
#define ITM_TCR_TraceBusID_Msk (0x7FUL << ITM_TCR_TraceBusID_Pos)
```

ITM TCR: ATBID Mask

5.24.2.26 ITM_TCR_TraceBusID_Pos

```
#define ITM_TCR_TraceBusID_Pos 16
```

ITM TCR: ATBID Position

5.24.2.27 ITM_TCR_TSENA_Msk

```
#define ITM_TCR_TSENA_Msk (1UL << ITM_TCR_TSENA_Pos)
```

ITM TCR: TSENA Mask

5.24.2.28 ITM_TCR_TSENA_Pos

```
#define ITM_TCR_TSENA_Pos 1
```

ITM TCR: TSENA Position

5.24.2.29 ITM_TCR_TSPrescale_Msk

```
#define ITM_TCR_TSPrescale_Msk (3UL << ITM_TCR_TSPrescale_Pos)
```

ITM TCR: TSPrescale Mask

5.24.2.30 ITM_TCR_TSPrescale_Pos

```
#define ITM_TCR_TSPrescale_Pos 8
```

ITM TCR: TSPrescale Position

5.24.2.31 ITM_TPR_PRIVMASK_Msk

```
#define ITM_TPR_PRIVMASK_Msk (0xFUL /*<< ITM_TPR_PRIVMASK_Pos*/)
```

ITM TPR: PRIVMASK Mask

5.24.2.32 ITM_TPR_PRIVMASK_Pos

```
#define ITM_TPR_PRIVMASK_Pos 0
```

ITM TPR: PRIVMASK Position

5.25 Data Watchpoint and Trace (DWT)

Type definitions for the Data Watchpoint and Trace (DWT)

Collaboration diagram for Data Watchpoint and Trace (DWT):



Modules

- [Trace Port Interface \(TPI\)](#)

Type definitions for the Trace Port Interface (TPI)

Classes

- struct [DWT_Type](#)

Structure type to access the Data Watchpoint and Trace Register (DWT).

- #define [DWT_CTRL_NUMCOMP_Pos](#) 28
- #define [DWT_CTRL_NUMCOMP_Msk](#) (0xFUL << [DWT_CTRL_NUMCOMP_Pos](#))
- #define [DWT_CTRL_NOTRCPKT_Pos](#) 27
- #define [DWT_CTRL_NOTRCPKT_Msk](#) (0x1UL << [DWT_CTRL_NOTRCPKT_Pos](#))
- #define [DWT_CTRL_NOEXTTRIG_Pos](#) 26
- #define [DWT_CTRL_NOEXTTRIG_Msk](#) (0x1UL << [DWT_CTRL_NOEXTTRIG_Pos](#))
- #define [DWT_CTRL_NOCYCCNT_Pos](#) 25
- #define [DWT_CTRL_NOCYCCNT_Msk](#) (0x1UL << [DWT_CTRL_NOCYCCNT_Pos](#))
- #define [DWT_CTRL_NOPRFCNT_Pos](#) 24
- #define [DWT_CTRL_NOPRFCNT_Msk](#) (0x1UL << [DWT_CTRL_NOPRFCNT_Pos](#))
- #define [DWT_CTRL_CYCEVTENA_Pos](#) 22
- #define [DWT_CTRL_CYCEVTENA_Msk](#) (0x1UL << [DWT_CTRL_CYCEVTENA_Pos](#))
- #define [DWT_CTRL_FOLDEVTENA_Pos](#) 21
- #define [DWT_CTRL_FOLDEVTENA_Msk](#) (0x1UL << [DWT_CTRL_FOLDEVTENA_Pos](#))
- #define [DWT_CTRL_LSUEVTENA_Pos](#) 20
- #define [DWT_CTRL_LSUEVTENA_Msk](#) (0x1UL << [DWT_CTRL_LSUEVTENA_Pos](#))
- #define [DWT_CTRL_SLEEPEVTENA_Pos](#) 19
- #define [DWT_CTRL_SLEEPEVTENA_Msk](#) (0x1UL << [DWT_CTRL_SLEEPEVTENA_Pos](#))
- #define [DWT_CTRL_EXCEVTENA_Pos](#) 18
- #define [DWT_CTRL_EXCEVTENA_Msk](#) (0x1UL << [DWT_CTRL_EXCEVTENA_Pos](#))
- #define [DWT_CTRL_CPIEVTENA_Pos](#) 17
- #define [DWT_CTRL_CPIEVTENA_Msk](#) (0x1UL << [DWT_CTRL_CPIEVTENA_Pos](#))
- #define [DWT_CTRL_EXCTRCENA_Pos](#) 16
- #define [DWT_CTRL_EXCTRCENA_Msk](#) (0x1UL << [DWT_CTRL_EXCTRCENA_Pos](#))
- #define [DWT_CTRL_PCSAMPLENA_Pos](#) 12
- #define [DWT_CTRL_PCSAMPLENA_Msk](#) (0x1UL << [DWT_CTRL_PCSAMPLENA_Pos](#))
- #define [DWT_CTRL_SYNCTAP_Pos](#) 10

- #define DWT_CTRL_SYNCTAP_Msk (0x3UL << DWT_CTRL_SYNCTAP_Pos)
- #define DWT_CTRL_CYCTAP_Pos 9
- #define DWT_CTRL_CYCTAP_Msk (0x1UL << DWT_CTRL_CYCTAP_Pos)
- #define DWT_CTRL_POSTINIT_Pos 5
- #define DWT_CTRL_POSTINIT_Msk (0xFUL << DWT_CTRL_POSTINIT_Pos)
- #define DWT_CTRL_POSTRESET_Pos 1
- #define DWT_CTRL_POSTRESET_Msk (0xFUL << DWT_CTRL_POSTRESET_Pos)
- #define DWT_CTRL_CYCCNTENA_Pos 0
- #define DWT_CTRL_CYCCNTENA_Msk (0x1UL /*<< DWT_CTRL_CYCCNTENA_Pos*/)
- #define DWT_CPLICNT_CPLICNT_Pos 0
- #define DWT_CPLICNT_CPLICNT_Msk (0xFFUL /*<< DWT_CPLICNT_CPLICNT_Pos*/)
- #define DWT_EXCCNT_EXCCNT_Pos 0
- #define DWT_EXCCNT_EXCCNT_Msk (0xFFUL /*<< DWT_EXCCNT_EXCCNT_Pos*/)
- #define DWT_SLEEPCNT_SLEEPCNT_Pos 0
- #define DWT_SLEEPCNT_SLEEPCNT_Msk (0xFFUL /*<< DWT_SLEEPCNT_SLEEPCNT_Pos*/)
- #define DWT_LSUCNT_LSUCNT_Pos 0
- #define DWT_LSUCNT_LSUCNT_Msk (0xFFUL /*<< DWT_LSUCNT_LSUCNT_Pos*/)
- #define DWT_FOLDCNT_FOLDCNT_Pos 0
- #define DWT_FOLDCNT_FOLDCNT_Msk (0xFFUL /*<< DWT_FOLDCNT_FOLDCNT_Pos*/)
- #define DWT_MASK_MASK_Pos 0
- #define DWT_MASK_MASK_Msk (0x1FUL /*<< DWT_MASK_MASK_Pos*/)
- #define DWT_FUNCTION_MATCHED_Pos 24
- #define DWT_FUNCTION_MATCHED_Msk (0x1UL << DWT_FUNCTION_MATCHED_Pos)
- #define DWT_FUNCTION_DATAVADDR1_Pos 16
- #define DWT_FUNCTION_DATAVADDR1_Msk (0xFUL << DWT_FUNCTION_DATAVADDR1_Pos)
- #define DWT_FUNCTION_DATAVADDR0_Pos 12
- #define DWT_FUNCTION_DATAVADDR0_Msk (0xFUL << DWT_FUNCTION_DATAVADDR0_Pos)
- #define DWT_FUNCTION_DATAVSIZE_Pos 10
- #define DWT_FUNCTION_DATAVSIZE_Msk (0x3UL << DWT_FUNCTION_DATAVSIZE_Pos)
- #define DWT_FUNCTION_LNK1ENA_Pos 9
- #define DWT_FUNCTION_LNK1ENA_Msk (0x1UL << DWT_FUNCTION_LNK1ENA_Pos)
- #define DWT_FUNCTION_DATAVMATCH_Pos 8
- #define DWT_FUNCTION_DATAVMATCH_Msk (0x1UL << DWT_FUNCTION_DATAVMATCH_Pos)
- #define DWT_FUNCTION_CYCMATCH_Pos 7
- #define DWT_FUNCTION_CYCMATCH_Msk (0x1UL << DWT_FUNCTION_CYCMATCH_Pos)
- #define DWT_FUNCTION_EMITRANGE_Pos 5
- #define DWT_FUNCTION_EMITRANGE_Msk (0x1UL << DWT_FUNCTION_EMITRANGE_Pos)
- #define DWT_FUNCTION_FUNCTION_Pos 0
- #define DWT_FUNCTION_FUNCTION_Msk (0xFUL /*<< DWT_FUNCTION_FUNCTION_Pos*/)

5.25.1 Detailed Description

Type definitions for the Data Watchpoint and Trace (DWT)

5.25.2 Macro Definition Documentation

5.25.2.1 DWT_CPICNT_CPICNT_Msk

```
#define DWT_CPICNT_CPICNT_Msk (0xFFUL /*<< DWT_CPICNT_CPICNT_Pos*/)
```

DWT CPICNT: CPICNT Mask

5.25.2.2 DWT_CPICNT_CPICNT_Pos

```
#define DWT_CPICNT_CPICNT_Pos 0
```

DWT CPICNT: CPICNT Position

5.25.2.3 DWT_CTRL_CPIEVTENA_Msk

```
#define DWT_CTRL_CPIEVTENA_Msk (0x1UL << DWT_CTRL_CPIEVTENA_Pos)
```

DWT CTRL: CPIEVTENA Mask

5.25.2.4 DWT_CTRL_CPIEVTENA_Pos

```
#define DWT_CTRL_CPIEVTENA_Pos 17
```

DWT CTRL: CPIEVTENA Position

5.25.2.5 DWT_CTRL_CYCCNTENA_Msk

```
#define DWT_CTRL_CYCCNTENA_Msk (0x1UL /*<< DWT_CTRL_CYCCNTENA_Pos*/)
```

DWT CTRL: CYCCNTENA Mask

5.25.2.6 DWT_CTRL_CYCCNTENA_Pos

```
#define DWT_CTRL_CYCCNTENA_Pos 0
```

DWT CTRL: CYCCNTENA Position

5.25.2.7 DWT_CTRL_CYCEVTENA_Msk

```
#define DWT_CTRL_CYCEVTENA_Msk (0x1UL << DWT_CTRL_CYCEVTENA_Pos)
```

DWT CTRL: CYCEVTENA Mask

5.25.2.8 DWT_CTRL_CYCEVTENA_Pos

```
#define DWT_CTRL_CYCEVTENA_Pos 22
```

DWT CTRL: CYCEVTENA Position

5.25.2.9 DWT_CTRL_CYCTAP_Msk

```
#define DWT_CTRL_CYCTAP_Msk (0x1UL << DWT_CTRL_CYCTAP_Pos)
```

DWT CTRL: CYCTAP Mask

5.25.2.10 DWT_CTRL_CYCTAP_Pos

```
#define DWT_CTRL_CYCTAP_Pos 9
```

DWT CTRL: CYCTAP Position

5.25.2.11 DWT_CTRL_EXCEVTENA_Msk

```
#define DWT_CTRL_EXCEVTENA_Msk (0x1UL << DWT_CTRL_EXCEVTENA_Pos)
```

DWT CTRL: EXCEVTENA Mask

5.25.2.12 DWT_CTRL_EXCEVTENA_Pos

```
#define DWT_CTRL_EXCEVTENA_Pos 18
```

DWT CTRL: EXCEVTENA Position

5.25.2.13 DWT_CTRL_EXCTRCENA_Msk

```
#define DWT_CTRL_EXCTRCENA_Msk (0x1UL << DWT_CTRL_EXCTRCENA_Pos)
```

DWT CTRL: EXCTRCENA Mask

5.25.2.14 DWT_CTRL_EXCTRCENA_Pos

```
#define DWT_CTRL_EXCTRCENA_Pos 16
```

DWT CTRL: EXCTRCENA Position

5.25.2.15 DWT_CTRL_FOLDEVTENA_Msk

```
#define DWT_CTRL_FOLDEVTENA_Msk (0x1UL << DWT_CTRL_FOLDEVTENA_Pos)
```

DWT CTRL: FOLDEVTENA Mask

5.25.2.16 DWT_CTRL_FOLDEVTENA_Pos

```
#define DWT_CTRL_FOLDEVTENA_Pos 21
```

DWT CTRL: FOLDEVTENA Position

5.25.2.17 DWT_CTRL_LSUEVTENA_Msk

```
#define DWT_CTRL_LSUEVTENA_Msk (0x1UL << DWT_CTRL_LSUEVTENA_Pos)
```

DWT CTRL: LSUEVTENA Mask

5.25.2.18 DWT_CTRL_LSUEVTENA_Pos

```
#define DWT_CTRL_LSUEVTENA_Pos 20
```

DWT CTRL: LSUEVTENA Position

5.25.2.19 DWT_CTRL_NOCYCCNT_Msk

```
#define DWT_CTRL_NOCYCCNT_Msk (0x1UL << DWT_CTRL_NOCYCCNT_Pos)
```

DWT CTRL: NOCYCCNT Mask

5.25.2.20 DWT_CTRL_NOCYCCNT_Pos

```
#define DWT_CTRL_NOCYCCNT_Pos 25
```

DWT CTRL: NOCYCCNT Position

5.25.2.21 DWT_CTRL_NOEXTTRIG_Msk

```
#define DWT_CTRL_NOEXTTRIG_Msk (0x1UL << DWT_CTRL_NOEXTTRIG_Pos)
```

DWT CTRL: NOEXTTRIG Mask

5.25.2.22 DWT_CTRL_NOEXTTRIG_Pos

```
#define DWT_CTRL_NOEXTTRIG_Pos 26
```

DWT CTRL: NOEXTTRIG Position

5.25.2.23 DWT_CTRL_NOPRFCNT_Msk

```
#define DWT_CTRL_NOPRFCNT_Msk (0x1UL << DWT_CTRL_NOPRFCNT_Pos)
```

DWT CTRL: NOPRFCNT Mask

5.25.2.24 DWT_CTRL_NOPRFCNT_Pos

```
#define DWT_CTRL_NOPRFCNT_Pos 24
```

DWT CTRL: NOPRFCNT Position

5.25.2.25 DWT_CTRL_NOTRCPKT_Msk

```
#define DWT_CTRL_NOTRCPKT_Msk (0x1UL << DWT_CTRL_NOTRCPKT_Pos)
```

DWT CTRL: NOTRCPKT Mask

5.25.2.26 DWT_CTRL_NOTRCPKT_Pos

```
#define DWT_CTRL_NOTRCPKT_Pos 27
```

DWT CTRL: NOTRCPKT Position

5.25.2.27 DWT_CTRL_NUMCOMP_Msk

```
#define DWT_CTRL_NUMCOMP_Msk (0xFUL << DWT_CTRL_NUMCOMP_Pos)
```

DWT CTRL: NUMCOMP Mask

5.25.2.28 DWT_CTRL_NUMCOMP_Pos

```
#define DWT_CTRL_NUMCOMP_Pos 28
```

DWT CTRL: NUMCOMP Position

5.25.2.29 DWT_CTRL_PCSAMPLENA_Msk

```
#define DWT_CTRL_PCSAMPLENA_Msk (0x1UL << DWT_CTRL_PCSAMPLENA_Pos)
```

DWT CTRL: PCSAMPLENA Mask

5.25.2.30 DWT_CTRL_PCSAMPLENA_Pos

```
#define DWT_CTRL_PCSAMPLENA_Pos 12
```

DWT CTRL: PCSAMPLENA Position

5.25.2.31 DWT_CTRL_POSTINIT_Msk

```
#define DWT_CTRL_POSTINIT_Msk (0xFUL << DWT_CTRL_POSTINIT_Pos)
```

DWT CTRL: POSTINIT Mask

5.25.2.32 DWT_CTRL_POSTINIT_Pos

```
#define DWT_CTRL_POSTINIT_Pos 5
```

DWT CTRL: POSTINIT Position

5.25.2.33 DWT_CTRL_POSTPRESET_Msk

```
#define DWT_CTRL_POSTPRESET_Msk (0xFUL << DWT_CTRL_POSTPRESET_Pos)
```

DWT CTRL: POSTPRESET Mask

5.25.2.34 DWT_CTRL_POSTPRESET_Pos

```
#define DWT_CTRL_POSTPRESET_Pos 1
```

DWT CTRL: POSTPRESET Position

5.25.2.35 DWT_CTRL_SLEEPEVTENA_Msk

```
#define DWT_CTRL_SLEEPEVTENA_Msk (0x1UL << DWT_CTRL_SLEEPEVTENA_Pos)
```

DWT CTRL: SLEEPEVTENA Mask

5.25.2.36 DWT_CTRL_SLEEPEVTENA_Pos

```
#define DWT_CTRL_SLEEPEVTENA_Pos 19
```

DWT CTRL: SLEEPEVTENA Position

5.25.2.37 DWT_CTRL_SYNCTAP_Msk

```
#define DWT_CTRL_SYNCTAP_Msk (0x3UL << DWT_CTRL_SYNCTAP_Pos)
```

DWT CTRL: SYNCTAP Mask

5.25.2.38 DWT_CTRL_SYNCTAP_Pos

```
#define DWT_CTRL_SYNCTAP_Pos 10
```

DWT CTRL: SYNCTAP Position

5.25.2.39 DWT_EXCCNT_EXCCNT_Msk

```
#define DWT_EXCCNT_EXCCNT_Msk (0xFFUL /*<< DWT_EXCCNT_EXCCNT_Pos*/)
```

DWT EXCCNT: EXCCNT Mask

5.25.2.40 DWT_EXCCNT_EXCCNT_Pos

```
#define DWT_EXCCNT_EXCCNT_Pos 0
```

DWT EXCCNT: EXCCNT Position

5.25.2.41 DWT_FOLDCNT_FOLDCNT_Msk

```
#define DWT_FOLDCNT_FOLDCNT_Msk (0xFFUL /*<< DWT_FOLDCNT_FOLDCNT_Pos*/)
```

DWT FOLDCNT: FOLDCNT Mask

5.25.2.42 DWT_FOLDCNT_FOLDCNT_Pos

```
#define DWT_FOLDCNT_FOLDCNT_Pos 0
```

DWT FOLDCNT: FOLDCNT Position

5.25.2.43 DWT_FUNCTION_CYCMATCH_Msk

```
#define DWT_FUNCTION_CYCMATCH_Msk (0x1UL << DWT_FUNCTION_CYCMATCH_Pos)
```

DWT FUNCTION: CYCMATCH Mask

5.25.2.44 DWT_FUNCTION_CYCMATCH_Pos

```
#define DWT_FUNCTION_CYCMATCH_Pos 7
```

DWT FUNCTION: CYCMATCH Position

5.25.2.45 DWT_FUNCTION_DATAVADDR0_Msk

```
#define DWT_FUNCTION_DATAVADDR0_Msk (0xFUL << DWT_FUNCTION_DATAVADDR0_Pos)
```

DWT FUNCTION: DATAVADDR0 Mask

5.25.2.46 DWT_FUNCTION_DATAVADDR0_Pos

```
#define DWT_FUNCTION_DATAVADDR0_Pos 12
```

DWT FUNCTION: DATAVADDR0 Position

5.25.2.47 DWT_FUNCTION_DATAVADDR1_Msk

```
#define DWT_FUNCTION_DATAVADDR1_Msk (0xFUL << DWT_FUNCTION_DATAVADDR1_Pos)
```

DWT FUNCTION: DATAVADDR1 Mask

5.25.2.48 DWT_FUNCTION_DATAVADDR1_Pos

```
#define DWT_FUNCTION_DATAVADDR1_Pos 16
```

DWT FUNCTION: DATAVADDR1 Position

5.25.2.49 DWT_FUNCTION_DATAVMATCH_Msk

```
#define DWT_FUNCTION_DATAVMATCH_Msk (0x1UL << DWT_FUNCTION_DATAVMATCH_Pos)
```

DWT FUNCTION: DATAVMATCH Mask

5.25.2.50 DWT_FUNCTION_DATAVMATCH_Pos

```
#define DWT_FUNCTION_DATAVMATCH_Pos 8
```

DWT FUNCTION: DATAVMATCH Position

5.25.2.51 DWT_FUNCTION_DATAVSIZE_Msk

```
#define DWT_FUNCTION_DATAVSIZE_Msk (0x3UL << DWT_FUNCTION_DATAVSIZE_Pos)
```

DWT FUNCTION: DATAVSIZE Mask

5.25.2.52 DWT_FUNCTION_DATAVSIZE_Pos

```
#define DWT_FUNCTION_DATAVSIZE_Pos 10
```

DWT FUNCTION: DATAVSIZE Position

5.25.2.53 DWT_FUNCTION_EMITRANGE_Msk

```
#define DWT_FUNCTION_EMITRANGE_Msk (0x1UL << DWT_FUNCTION_EMITRANGE_Pos)
```

DWT FUNCTION: EMITRANGE Mask

5.25.2.54 DWT_FUNCTION_EMITRANGE_Pos

```
#define DWT_FUNCTION_EMITRANGE_Pos 5
```

DWT FUNCTION: EMITRANGE Position

5.25.2.55 DWT_FUNCTION_FUNCTION_Msk

```
#define DWT_FUNCTION_FUNCTION_Msk (0xFUL /*<< DWT_FUNCTION_FUNCTION_Pos*/)
```

DWT FUNCTION: FUNCTION Mask

5.25.2.56 DWT_FUNCTION_FUNCTION_Pos

```
#define DWT_FUNCTION_FUNCTION_Pos 0
```

DWT FUNCTION: FUNCTION Position

5.25.2.57 DWT_FUNCTION_LNK1ENA_Msk

```
#define DWT_FUNCTION_LNK1ENA_Msk (0x1UL << DWT_FUNCTION_LNK1ENA_Pos)
```

DWT FUNCTION: LNK1ENA Mask

5.25.2.58 DWT_FUNCTION_LNK1ENA_Pos

```
#define DWT_FUNCTION_LNK1ENA_Pos 9
```

DWT FUNCTION: LNK1ENA Position

5.25.2.59 DWT_FUNCTION_MATCHED_Msk

```
#define DWT_FUNCTION_MATCHED_Msk (0x1UL << DWT_FUNCTION_MATCHED_Pos)
```

DWT FUNCTION: MATCHED Mask

5.25.2.60 DWT_FUNCTION_MATCHED_Pos

```
#define DWT_FUNCTION_MATCHED_Pos 24
```

DWT FUNCTION: MATCHED Position

5.25.2.61 DWT_LSUCNT_LSUCNT_Msk

```
#define DWT_LSUCNT_LSUCNT_Msk (0xFFUL /*<< DWT_LSUCNT_LSUCNT_Pos*/)
```

DWT LSUCNT: LSUCNT Mask

5.25.2.62 DWT_LSUCNT_LSUCNT_Pos

```
#define DWT_LSUCNT_LSUCNT_Pos 0
```

DWT LSUCNT: LSUCNT Position

5.25.2.63 DWT_MASK_MASK_Msk

```
#define DWT_MASK_MASK_Msk (0x1FUL /*<< DWT_MASK_MASK_Pos*/)
```

DWT MASK: MASK Mask

5.25.2.64 DWT_MASK_MASK_Pos

```
#define DWT_MASK_MASK_Pos 0
```

DWT MASK: MASK Position

5.25.2.65 DWT_SLEEPCNT_SLEEPCNT_Msk

```
#define DWT_SLEEPCNT_SLEEPCNT_Msk (0xFFUL /*<< DWT_SLEEPCNT_SLEEPCNT_Pos*/)
```

DWT SLEEPCNT: SLEEPCNT Mask

5.25.2.66 DWT_SLEEPCNT_SLEEPCNT_Pos

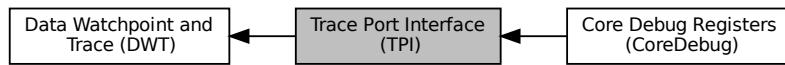
```
#define DWT_SLEEPCNT_SLEEPCNT_Pos 0
```

DWT SLEEPCNT: SLEEPCNT Position

5.26 Trace Port Interface (TPI)

Type definitions for the Trace Port Interface (TPI)

Collaboration diagram for Trace Port Interface (TPI):



Modules

- [Core Debug Registers \(CoreDebug\)](#)

Type definitions for the Core Debug Registers.

Classes

- struct [TPI_Type](#)

Structure type to access the Trace Port Interface Register (TPI).

- #define [TPI_ACPR_PRESCALER_Pos](#) 0
- #define [TPI_ACPR_PRESCALER_Msk](#) (0x1FFFUL /*<< [TPI_ACPR_PRESCALER_Pos](#)*/)
- #define [TPI_SPPR_TXMODE_Pos](#) 0
- #define [TPI_SPPR_TXMODE_Msk](#) (0x3UL /*<< [TPI_SPPR_TXMODE_Pos](#)*/)
- #define [TPI_FFSR_FtNonStop_Pos](#) 3
- #define [TPI_FFSR_FtNonStop_Msk](#) (0x1UL << [TPI_FFSR_FtNonStop_Pos](#))
- #define [TPI_FFSR_TCPresent_Pos](#) 2
- #define [TPI_FFSR_TCPresent_Msk](#) (0x1UL << [TPI_FFSR_TCPresent_Pos](#))
- #define [TPI_FFSR_FtStopped_Pos](#) 1
- #define [TPI_FFSR_FtStopped_Msk](#) (0x1UL << [TPI_FFSR_FtStopped_Pos](#))
- #define [TPI_FFSR_FInProg_Pos](#) 0
- #define [TPI_FFSR_FInProg_Msk](#) (0x1UL /*<< [TPI_FFSR_FInProg_Pos](#)*/)

- #define TPI_FFCR_TrigIn_Pos 8
- #define TPI_FFCR_TrigIn_Msk (0x1UL << TPI_FFCR_TrigIn_Pos)
- #define TPI_FFCR_EnFCont_Pos 1
- #define TPI_FFCR_EnFCont_Msk (0x1UL << TPI_FFCR_EnFCont_Pos)
- #define TPI_TRIGGER_TRIGGER_Pos 0
- #define TPI_TRIGGER_TRIGGER_Msk (0x1UL /*<< TPI_TRIGGER_TRIGGER_Pos*/)
- #define TPI_FIFO0_ITM_ATVALID_Pos 29
- #define TPI_FIFO0_ITM_ATVALID_Msk (0x3UL << TPI_FIFO0_ITM_ATVALID_Pos)
- #define TPI_FIFO0_ITM_bytectcount_Pos 27
- #define TPI_FIFO0_ITM_bytectcount_Msk (0x3UL << TPI_FIFO0_ITM_bytectcount_Pos)
- #define TPI_FIFO0_ETM_ATVALID_Pos 26
- #define TPI_FIFO0_ETM_ATVALID_Msk (0x3UL << TPI_FIFO0_ETM_ATVALID_Pos)
- #define TPI_FIFO0_ETM_bytectcount_Pos 24
- #define TPI_FIFO0_ETM_bytectcount_Msk (0x3UL << TPI_FIFO0_ETM_bytectcount_Pos)
- #define TPI_FIFO0_ETM2_Pos 16
- #define TPI_FIFO0_ETM2_Msk (0xFFUL << TPI_FIFO0_ETM2_Pos)
- #define TPI_FIFO0_ETM1_Pos 8
- #define TPI_FIFO0_ETM1_Msk (0xFFUL << TPI_FIFO0_ETM1_Pos)
- #define TPI_FIFO0_ETM0_Pos 0
- #define TPI_FIFO0_ETM0_Msk (0xFFUL /*<< TPI_FIFO0_ETM0_Pos*/)
- #define TPI_ITATBCTR2_ATREADY_Pos 0
- #define TPI_ITATBCTR2_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR2_ATREADY_Pos*/)
- #define TPI_FIFO1_ITM_ATVALID_Pos 29
- #define TPI_FIFO1_ITM_ATVALID_Msk (0x3UL << TPI_FIFO1_ITM_ATVALID_Pos)
- #define TPI_FIFO1_ITM_bytectcount_Pos 27
- #define TPI_FIFO1_ITM_bytectcount_Msk (0x3UL << TPI_FIFO1_ITM_bytectcount_Pos)
- #define TPI_FIFO1_ETM_ATVALID_Pos 26
- #define TPI_FIFO1_ETM_ATVALID_Msk (0x3UL << TPI_FIFO1_ETM_ATVALID_Pos)
- #define TPI_FIFO1_ETM_bytectcount_Pos 24
- #define TPI_FIFO1_ETM_bytectcount_Msk (0x3UL << TPI_FIFO1_ETM_bytectcount_Pos)
- #define TPI_FIFO1_ITM2_Pos 16
- #define TPI_FIFO1_ITM2_Msk (0xFFUL << TPI_FIFO1_ITM2_Pos)
- #define TPI_FIFO1_ITM1_Pos 8
- #define TPI_FIFO1_ITM1_Msk (0xFFUL << TPI_FIFO1_ITM1_Pos)
- #define TPI_FIFO1_ITM0_Pos 0
- #define TPI_FIFO1_ITM0_Msk (0xFFUL /*<< TPI_FIFO1_ITM0_Pos*/)
- #define TPI_ITATBCTR0_ATREADY_Pos 0
- #define TPI_ITATBCTR0_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR0_ATREADY_Pos*/)
- #define TPI_ITCTRL_Mode_Pos 0
- #define TPI_ITCTRL_Mode_Msk (0x1UL /*<< TPI_ITCTRL_Mode_Pos*/)
- #define TPI_DEVID_NRZVALID_Pos 11
- #define TPI_DEVID_NRZVALID_Msk (0x1UL << TPI_DEVID_NRZVALID_Pos)
- #define TPI_DEVID_MANCVALID_Pos 10
- #define TPI_DEVID_MANCVALID_Msk (0x1UL << TPI_DEVID_MANCVALID_Pos)
- #define TPI_DEVID_PTINVALID_Pos 9
- #define TPI_DEVID_PTINVALID_Msk (0x1UL << TPI_DEVID_PTINVALID_Pos)
- #define TPI_DEVID_MinBufSz_Pos 6
- #define TPI_DEVID_MinBufSz_Msk (0x7UL << TPI_DEVID_MinBufSz_Pos)
- #define TPI_DEVID_AsynClkln_Pos 5
- #define TPI_DEVID_AsynClkln_Msk (0x1UL << TPI_DEVID_AsynClkln_Pos)
- #define TPI_DEVID_NrTraceInput_Pos 0
- #define TPI_DEVID_NrTraceInput_Msk (0xFUL /*<< TPI_DEVID_NrTraceInput_Pos*/)
- #define TPI_DEVTYPE_MajorType_Pos 4
- #define TPI_DEVTYPE_MajorType_Msk (0xFUL << TPI_DEVTYPE_MajorType_Pos)
- #define TPI_DEVTYPE_SubType_Pos 0
- #define TPI_DEVTYPE_SubType_Msk (0xFUL /*<< TPI_DEVTYPE_SubType_Pos*/)

5.26.1 Detailed Description

Type definitions for the Trace Port Interface (TPI)

5.26.2 Macro Definition Documentation

5.26.2.1 TPI_ACPR_PRESCALER_Msk

```
#define TPI_ACPR_PRESCALER_Msk (0x1FFFUL /*<< TPI_ACPR_PRESCALER_Pos*/)
```

TPI ACPR: PRESCALER Mask

5.26.2.2 TPI_ACPR_PRESCALER_Pos

```
#define TPI_ACPR_PRESCALER_Pos 0
```

TPI ACPR: PRESCALER Position

5.26.2.3 TPI_DEVID_AsynClkIn_Msk

```
#define TPI_DEVID_AsynClkIn_Msk (0x1UL << TPI_DEVID_AsynClkIn_Pos)
```

TPI DEVID: AsynClkIn Mask

5.26.2.4 TPI_DEVID_AsynClkIn_Pos

```
#define TPI_DEVID_AsynClkIn_Pos 5
```

TPI DEVID: AsynClkIn Position

5.26.2.5 TPI_DEVID_MANCVALID_Msk

```
#define TPI_DEVID_MANCVALID_Msk (0x1UL << TPI_DEVID_MANCVALID_Pos)
```

TPI DEVID: MANCVALID Mask

5.26.2.6 TPI_DEVID_MANCVALID_Pos

```
#define TPI_DEVID_MANCVALID_Pos 10
```

TPI DEVID: MANCVALID Position

5.26.2.7 TPI_DEVID_MinBufSz_Msk

```
#define TPI_DEVID_MinBufSz_Msk (0x7UL << TPI_DEVID_MinBufSz_Pos)
```

TPI DEVID: MinBufSz Mask

5.26.2.8 TPI_DEVID_MinBufSz_Pos

```
#define TPI_DEVID_MinBufSz_Pos 6
```

TPI DEVID: MinBufSz Position

5.26.2.9 TPI_DEVID_NrTraceInput_Msk

```
#define TPI_DEVID_NrTraceInput_Msk (0x1FUL /*<< TPI_DEVID_NrTraceInput_Pos*/)
```

TPI DEVID: NrTraceInput Mask

5.26.2.10 TPI_DEVID_NrTraceInput_Pos

```
#define TPI_DEVID_NrTraceInput_Pos 0
```

TPI DEVID: NrTraceInput Position

5.26.2.11 TPI_DEVID_NRZVALID_Msk

```
#define TPI_DEVID_NRZVALID_Msk (0x1UL << TPI_DEVID_NRZVALID_Pos)
```

TPI DEVID: NRZVALID Mask

5.26.2.12 TPI_DEVID_NRZVALID_Pos

```
#define TPI_DEVID_NRZVALID_Pos 11
```

TPI DEVID: NRZVALID Position

5.26.2.13 TPI_DEVID_PTINVALID_Msk

```
#define TPI_DEVID_PTINVALID_Msk (0x1UL << TPI_DEVID_PTINVALID_Pos)
```

TPI DEVID: PTINVALID Mask

5.26.2.14 TPI_DEVID_PTINVALID_Pos

```
#define TPI_DEVID_PTINVALID_Pos 9
```

TPI DEVID: PTINVALID Position

5.26.2.15 TPI_DEVTYPE_MajorType_Msk

```
#define TPI_DEVTYPE_MajorType_Msk (0xFUL << TPI_DEVTYPE_MajorType_Pos)
```

TPI DEVTYPE: MajorType Mask

5.26.2.16 TPI_DEVTYPE_MajorType_Pos

```
#define TPI_DEVTYPE_MajorType_Pos 4
```

TPI DEVTYPE: MajorType Position

5.26.2.17 TPI_DEVTYPE_SubType_Msk

```
#define TPI_DEVTYPE_SubType_Msk (0xFUL /*<< TPI_DEVTYPE_SubType_Pos*/)
```

TPI DEVTYPE: SubType Mask

5.26.2.18 TPI_DEVTYPE_SubType_Pos

```
#define TPI_DEVTYPE_SubType_Pos 0
```

TPI DEVTYPE: SubType Position

5.26.2.19 TPI_FFCR_EnFCont_Msk

```
#define TPI_FFCR_EnFCont_Msk (0x1UL << TPI_FFCR_EnFCont_Pos)
```

TPI FFCR: EnFCont Mask

5.26.2.20 TPI_FFCR_EnFCont_Pos

```
#define TPI_FFCR_EnFCont_Pos 1
```

TPI FFCR: EnFCont Position

5.26.2.21 TPI_FFCR_TrigIn_Msk

```
#define TPI_FFCR_TrigIn_Msk (0x1UL << TPI_FFCR_TrigIn_Pos)
```

TPI FFCR: TrigIn Mask

5.26.2.22 TPI_FFCR_TrigIn_Pos

```
#define TPI_FFCR_TrigIn_Pos 8
```

TPI FFCR: TrigIn Position

5.26.2.23 TPI_FFSR_FInProg_Msk

```
#define TPI_FFSR_FInProg_Msk (0x1UL /*<< TPI_FFSR_FInProg_Pos*/)
```

TPI FFSR: FInProg Mask

5.26.2.24 TPI_FFSR_FInProg_Pos

```
#define TPI_FFSR_FInProg_Pos 0
```

TPI FFSR: FInProg Position

5.26.2.25 TPI_FFSR_FtNonStop_Msk

```
#define TPI_FFSR_FtNonStop_Msk (0x1UL << TPI_FFSR_FtNonStop_Pos)
```

TPI FFSR: FtNonStop Mask

5.26.2.26 TPI_FFSR_FtNonStop_Pos

```
#define TPI_FFSR_FtNonStop_Pos 3
```

TPI FFSR: FtNonStop Position

5.26.2.27 TPI_FFSR_FtStopped_Msk

```
#define TPI_FFSR_FtStopped_Msk (0x1UL << TPI_FFSR_FtStopped_Pos)
```

TPI FFSR: FtStopped Mask

5.26.2.28 TPI_FFSR_FtStopped_Pos

```
#define TPI_FFSR_FtStopped_Pos 1
```

TPI FFSR: FtStopped Position

5.26.2.29 TPI_FFSR_TCPresent_Msk

```
#define TPI_FFSR_TCPresent_Msk (0x1UL << TPI_FFSR_TCPresent_Pos)
```

TPI FFSR: TCPresent Mask

5.26.2.30 TPI_FFSR_TCPresent_Pos

```
#define TPI_FFSR_TCPresent_Pos 2
```

TPI FFSR: TCPresent Position

5.26.2.31 TPI_FIFO0_ETM0_Msk

```
#define TPI_FIFO0_ETM0_Msk (0xFFUL /*<< TPI_FIFO0_ETM0_Pos*/)
```

TPI FIFO0: ETM0 Mask

5.26.2.32 TPI_FIFO0_ETM0_Pos

```
#define TPI_FIFO0_ETM0_Pos 0
```

TPI FIFO0: ETM0 Position

5.26.2.33 TPI_FIFO0_ETM1_Msk

```
#define TPI_FIFO0_ETM1_Msk (0xFFUL << TPI_FIFO0_ETM1_Pos)
```

TPI FIFO0: ETM1 Mask

5.26.2.34 TPI_FIFO0_ETM1_Pos

```
#define TPI_FIFO0_ETM1_Pos 8
```

TPI FIFO0: ETM1 Position

5.26.2.35 TPI_FIFO0_ETM2_Msk

```
#define TPI_FIFO0_ETM2_Msk (0xFFUL << TPI_FIFO0_ETM2_Pos)
```

TPI FIFO0: ETM2 Mask

5.26.2.36 TPI_FIFO0_ETM2_Pos

```
#define TPI_FIFO0_ETM2_Pos 16
```

TPI FIFO0: ETM2 Position

5.26.2.37 TPI_FIFO0_ETM_ATVALID_Msk

```
#define TPI_FIFO0_ETM_ATVALID_Msk (0x3UL << TPI_FIFO0_ETM_ATVALID_Pos)
```

TPI FIFO0: ETM_ATVALID Mask

5.26.2.38 TPI_FIFO0_ETM_ATVALID_Pos

```
#define TPI_FIFO0_ETM_ATVALID_Pos 26
```

TPI FIFO0: ETM_ATVALID Position

5.26.2.39 TPI_FIFO0_ETM_bytecount_Msk

```
#define TPI_FIFO0_ETM_bytecount_Msk (0x3UL << TPI_FIFO0_ETM_bytecount_Pos)
```

TPI FIFO0: ETM_bytecount Mask

5.26.2.40 TPI_FIFO0_ETM_bytecount_Pos

```
#define TPI_FIFO0_ETM_bytecount_Pos 24
```

TPI FIFO0: ETM_bytecount Position

5.26.2.41 TPI_FIFO0_ITM_ATVALID_Msk

```
#define TPI_FIFO0_ITM_ATVALID_Msk (0x3UL << TPI_FIFO0_ITM_ATVALID_Pos)
```

TPI FIFO0: ITM_ATVALID Mask

5.26.2.42 TPI_FIFO0_ITM_ATVALID_Pos

```
#define TPI_FIFO0_ITM_ATVALID_Pos 29
```

TPI FIFO0: ITM_ATVALID Position

5.26.2.43 TPI_FIFO0_ITM_bytecount_Msk

```
#define TPI_FIFO0_ITM_bytecount_Msk (0x3UL << TPI_FIFO0_ITM_bytecount_Pos)
```

TPI FIFO0: ITM_bytecount Mask

5.26.2.44 TPI_FIFO0_ITM_bytecount_Pos

```
#define TPI_FIFO0_ITM_bytecount_Pos 27
```

TPI FIFO0: ITM_bytecount Position

5.26.2.45 TPI_FIFO1_ETM_ATVALID_Msk

```
#define TPI_FIFO1_ETM_ATVALID_Msk (0x3UL << TPI_FIFO1_ETM_ATVALID_Pos)
```

TPI FIFO1: ETM_ATVALID Mask

5.26.2.46 TPI_FIFO1_ETM_ATVALID_Pos

```
#define TPI_FIFO1_ETM_ATVALID_Pos 26
```

TPI FIFO1: ETM_ATVALID Position

5.26.2.47 TPI_FIFO1_ETM_bytecount_Msk

```
#define TPI_FIFO1_ETM_bytecount_Msk (0x3UL << TPI_FIFO1_ETM_bytecount_Pos)
```

TPI FIFO1: ETM_bytecount Mask

5.26.2.48 TPI_FIFO1_ETM_bytecount_Pos

```
#define TPI_FIFO1_ETM_bytecount_Pos 24
```

TPI FIFO1: ETM_bytecount Position

5.26.2.49 TPI_FIFO1_ITM0_Msk

```
#define TPI_FIFO1_ITM0_Msk (0xFFUL /*<< TPI_FIFO1_ITM0_Pos*/)
```

TPI FIFO1: ITM0 Mask

5.26.2.50 TPI_FIFO1_ITM0_Pos

```
#define TPI_FIFO1_ITM0_Pos 0
```

TPI FIFO1: ITM0 Position

5.26.2.51 TPI_FIFO1_ITM1_Msk

```
#define TPI_FIFO1_ITM1_Msk (0xFFUL << TPI_FIFO1_ITM1_Pos)
```

TPI FIFO1: ITM1 Mask

5.26.2.52 TPI_FIFO1_ITM1_Pos

```
#define TPI_FIFO1_ITM1_Pos 8
```

TPI FIFO1: ITM1 Position

5.26.2.53 TPI_FIFO1_ITM2_Msk

```
#define TPI_FIFO1_ITM2_Msk (0xFFUL << TPI_FIFO1_ITM2_Pos)
```

TPI FIFO1: ITM2 Mask

5.26.2.54 TPI_FIFO1_ITM2_Pos

```
#define TPI_FIFO1_ITM2_Pos 16
```

TPI FIFO1: ITM2 Position

5.26.2.55 TPI_FIFO1_ITM_ATVALID_Msk

```
#define TPI_FIFO1_ITM_ATVALID_Msk (0x3UL << TPI_FIFO1_ITM_ATVALID_Pos)
```

TPI FIFO1: ITM_ATVALID Mask

5.26.2.56 TPI_FIFO1_ITM_ATVALID_Pos

```
#define TPI_FIFO1_ITM_ATVALID_Pos 29
```

TPI FIFO1: ITM_ATVALID Position

5.26.2.57 TPI_FIFO1_ITM_bytecount_Msk

```
#define TPI_FIFO1_ITM_bytecount_Msk (0x3UL << TPI_FIFO1_ITM_bytecount_Pos)
```

TPI FIFO1: ITM_bytecount Mask

5.26.2.58 TPI_FIFO1_ITM_bytecount_Pos

```
#define TPI_FIFO1_ITM_bytecount_Pos 27
```

TPI FIFO1: ITM_bytecount Position

5.26.2.59 TPI_ITATBCTR0_ATREADY_Msk

```
#define TPI_ITATBCTR0_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR0_ATREADY_Pos*/)
```

TPI ITATBCTR0: ATREADY Mask

5.26.2.60 TPI_ITATBCTR0_ATREADY_Pos

```
#define TPI_ITATBCTR0_ATREADY_Pos 0
```

TPI ITATBCTR0: ATREADY Position

5.26.2.61 TPI_ITATBCTR2_ATREADY_Msk

```
#define TPI_ITATBCTR2_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR2_ATREADY_Pos*/)
```

TPI ITATBCTR2: ATREADY Mask

5.26.2.62 TPI_ITATBCTR2_ATREADY_Pos

```
#define TPI_ITATBCTR2_ATREADY_Pos 0
```

TPI ITATBCTR2: ATREADY Position

5.26.2.63 TPI_ITCTRL_Mode_Msk

```
#define TPI_ITCTRL_Mode_Msk (0x1UL /*<< TPI_ITCTRL_Mode_Pos*/)
```

TPI ITCTRL: Mode Mask

5.26.2.64 TPI_ITCTRL_Mode_Pos

```
#define TPI_ITCTRL_Mode_Pos 0
```

TPI ITCTRL: Mode Position

5.26.2.65 TPI_SPPR_TXMODE_Msk

```
#define TPI_SPPR_TXMODE_Msk (0x3UL /*<< TPI_SPPR_TXMODE_Pos*/)
```

TPI SPPR: TXMODE Mask

5.26.2.66 TPI_SPPR_TXMODE_Pos

```
#define TPI_SPPR_TXMODE_Pos 0
```

TPI SPPR: TXMODE Position

5.26.2.67 TPI_TRIGGER_TRIGGER_Msk

```
#define TPI_TRIGGER_TRIGGER_Msk (0x1UL /*<< TPI_TRIGGER_TRIGGER_Pos*/)
```

TPI TRIGGER: TRIGGER Mask

5.26.2.68 TPI_TRIGGER_TRIGGER_Pos

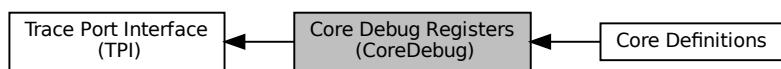
```
#define TPI_TRIGGER_TRIGGER_Pos 0
```

TPI TRIGGER: TRIGGER Position

5.27 Core Debug Registers (CoreDebug)

Type definitions for the Core Debug Registers.

Collaboration diagram for Core Debug Registers (CoreDebug):



Modules

- Core Definitions

Definitions for base addresses, unions, and structures.

Classes

- struct `CoreDebug_Type`

Structure type to access the Core Debug Register (CoreDebug).

- #define `CoreDebug_DHCSR_DBGKEY_Pos` 16
- #define `CoreDebug_DHCSR_DBGKEY_Msk` (0xFFFFUL << `CoreDebug_DHCSR_DBGKEY_Pos`)
- #define `CoreDebug_DHCSR_S_RESET_ST_Pos` 25
- #define `CoreDebug_DHCSR_S_RESET_ST_Msk` (1UL << `CoreDebug_DHCSR_S_RESET_ST_Pos`)
- #define `CoreDebug_DHCSR_S_RETIRE_ST_Pos` 24
- #define `CoreDebug_DHCSR_S_RETIRE_ST_Msk` (1UL << `CoreDebug_DHCSR_S_RETIRE_ST_Pos`)
- #define `CoreDebug_DHCSR_S_LOCKUP_Pos` 19
- #define `CoreDebug_DHCSR_S_LOCKUP_Msk` (1UL << `CoreDebug_DHCSR_S_LOCKUP_Pos`)
- #define `CoreDebug_DHCSR_S_SLEEP_Pos` 18
- #define `CoreDebug_DHCSR_S_SLEEP_Msk` (1UL << `CoreDebug_DHCSR_S_SLEEP_Pos`)
- #define `CoreDebug_DHCSR_S_HALT_Pos` 17
- #define `CoreDebug_DHCSR_S_HALT_Msk` (1UL << `CoreDebug_DHCSR_S_HALT_Pos`)
- #define `CoreDebug_DHCSR_S_REGRDY_Pos` 16
- #define `CoreDebug_DHCSR_S_REGRDY_Msk` (1UL << `CoreDebug_DHCSR_S_REGRDY_Pos`)
- #define `CoreDebug_DHCSR_C_SNAPSTALL_Pos` 5
- #define `CoreDebug_DHCSR_C_SNAPSTALL_Msk` (1UL << `CoreDebug_DHCSR_C_SNAPSTALL_Pos`)
- #define `CoreDebug_DHCSR_C_MASKINTS_Pos` 3
- #define `CoreDebug_DHCSR_C_MASKINTS_Msk` (1UL << `CoreDebug_DHCSR_C_MASKINTS_Pos`)
- #define `CoreDebug_DHCSR_C_STEP_Pos` 2
- #define `CoreDebug_DHCSR_C_STEP_Msk` (1UL << `CoreDebug_DHCSR_C_STEP_Pos`)
- #define `CoreDebug_DHCSR_C_HALT_Pos` 1
- #define `CoreDebug_DHCSR_C_HALT_Msk` (1UL << `CoreDebug_DHCSR_C_HALT_Pos`)
- #define `CoreDebug_DHCSR_C_DEBUGEN_Pos` 0
- #define `CoreDebug_DHCSR_C_DEBUGEN_Msk` (1UL /*<< `CoreDebug_DHCSR_C_DEBUGEN_Pos*/)`
- #define `CoreDebug_DCRSR_REGWnR_Pos` 16
- #define `CoreDebug_DCRSR_REGWnR_Msk` (1UL << `CoreDebug_DCRSR_REGWnR_Pos`)
- #define `CoreDebug_DCRSR_REGSEL_Pos` 0
- #define `CoreDebug_DCRSR_REGSEL_Msk` (0x1FUL /*<< `CoreDebug_DCRSR_REGSEL_Pos*/)`
- #define `CoreDebug_DEMCR_TRCENA_Pos` 24
- #define `CoreDebug_DEMCR_TRCENA_Msk` (1UL << `CoreDebug_DEMCR_TRCENA_Pos`)
- #define `CoreDebug_DEMCR_MON_REQ_Pos` 19
- #define `CoreDebug_DEMCR_MON_REQ_Msk` (1UL << `CoreDebug_DEMCR_MON_REQ_Pos`)
- #define `CoreDebug_DEMCR_MON_STEP_Pos` 18
- #define `CoreDebug_DEMCR_MON_STEP_Msk` (1UL << `CoreDebug_DEMCR_MON_STEP_Pos`)
- #define `CoreDebug_DEMCR_MON_PEND_Pos` 17
- #define `CoreDebug_DEMCR_MON_PEND_Msk` (1UL << `CoreDebug_DEMCR_MON_PEND_Pos`)
- #define `CoreDebug_DEMCR_MON_EN_Pos` 16
- #define `CoreDebug_DEMCR_MON_EN_Msk` (1UL << `CoreDebug_DEMCR_MON_EN_Pos`)
- #define `CoreDebug_DEMCR_VC_HARDERR_Pos` 10
- #define `CoreDebug_DEMCR_VC_HARDERR_Msk` (1UL << `CoreDebug_DEMCR_VC_HARDERR_Pos`)
- #define `CoreDebug_DEMCR_VC_INTERR_Pos` 9
- #define `CoreDebug_DEMCR_VC_INTERR_Msk` (1UL << `CoreDebug_DEMCR_VC_INTERR_Pos`)
- #define `CoreDebug_DEMCR_VC_BUSERR_Pos` 8

- #define CoreDebug_DEMCR_VC_BUSERR_Msk (1UL << CoreDebug_DEMCR_VC_BUSERR_Pos)
- #define CoreDebug_DEMCR_VC_STATERR_Pos 7
- #define CoreDebug_DEMCR_VC_STATERR_Msk (1UL << CoreDebug_DEMCR_VC_STATERR_Pos)
- #define CoreDebug_DEMCR_VC_CHKERR_Pos 6
- #define CoreDebug_DEMCR_VC_CHKERR_Msk (1UL << CoreDebug_DEMCR_VC_CHKERR_Pos)
- #define CoreDebug_DEMCR_VC_NOCPERR_Pos 5
- #define CoreDebug_DEMCR_VC_NOCPERR_Msk (1UL << CoreDebug_DEMCR_VC_NOCPERR_Pos)
- #define CoreDebug_DEMCR_VC_MMERR_Pos 4
- #define CoreDebug_DEMCR_VC_MMERR_Msk (1UL << CoreDebug_DEMCR_VC_MMERR_Pos)
- #define CoreDebug_DEMCR_VC_CORERESET_Pos 0
- #define CoreDebug_DEMCR_VC_CORERESET_Msk (1UL /*<< CoreDebug_DEMCR_VC_CORERESET_Pos*/)

5.27.1 Detailed Description

Type definitions for the Core Debug Registers.

5.27.2 Macro Definition Documentation

5.27.2.1 CoreDebug_DCRSR_REGSEL_Msk

```
#define CoreDebug_DCRSR_REGSEL_Msk (0x1FUL /*<< CoreDebug_DCRSR_REGSEL_Pos*/)
```

CoreDebug DCRSR: REGSEL Mask

5.27.2.2 CoreDebug_DCRSR_REGSEL_Pos

```
#define CoreDebug_DCRSR_REGSEL_Pos 0
```

CoreDebug DCRSR: REGSEL Position

5.27.2.3 CoreDebug_DCRSR_REGWnR_Msk

```
#define CoreDebug_DCRSR_REGWnR_Msk (1UL << CoreDebug_DCRSR_REGWnR_Pos)
```

CoreDebug DCRSR: REGWnR Mask

5.27.2.4 CoreDebug_DCRSR_REGWnR_Pos

```
#define CoreDebug_DCRSR_REGWnR_Pos 16
```

CoreDebug DCRSR: REGWnR Position

5.27.2.5 CoreDebug_DEMCR_MON_EN_Msk

```
#define CoreDebug_DEMCR_MON_EN_Msk (1UL << CoreDebug_DEMCR_MON_EN_Pos)
```

CoreDebug DEMCR: MON_EN Mask

5.27.2.6 CoreDebug_DEMCR_MON_EN_Pos

```
#define CoreDebug_DEMCR_MON_EN_Pos 16
```

CoreDebug DEMCR: MON_EN Position

5.27.2.7 CoreDebug_DEMCR_MON_PEND_Msk

```
#define CoreDebug_DEMCR_MON_PEND_Msk (1UL << CoreDebug_DEMCR_MON_PEND_Pos)
```

CoreDebug DEMCR: MON_PEND Mask

5.27.2.8 CoreDebug_DEMCR_MON_PEND_Pos

```
#define CoreDebug_DEMCR_MON_PEND_Pos 17
```

CoreDebug DEMCR: MON_PEND Position

5.27.2.9 CoreDebug_DEMCR_MON_REQ_Msk

```
#define CoreDebug_DEMCR_MON_REQ_Msk (1UL << CoreDebug_DEMCR_MON_REQ_Pos)
```

CoreDebug DEMCR: MON_REQ Mask

5.27.2.10 CoreDebug_DEMCR_MON_REQ_Pos

```
#define CoreDebug_DEMCR_MON_REQ_Pos 19
```

CoreDebug DEMCR: MON_REQ Position

5.27.2.11 CoreDebug_DEMCR_MON_STEP_Msk

```
#define CoreDebug_DEMCR_MON_STEP_Msk (1UL << CoreDebug_DEMCR_MON_STEP_Pos)
```

CoreDebug DEMCR: MON_STEP Mask

5.27.2.12 CoreDebug_DEMCR_MON_STEP_Pos

```
#define CoreDebug_DEMCR_MON_STEP_Pos 18
```

CoreDebug DEMCR: MON_STEP Position

5.27.2.13 CoreDebug_DEMCR_TRCENA_Msk

```
#define CoreDebug_DEMCR_TRCENA_Msk (1UL << CoreDebug_DEMCR_TRCENA_Pos)
```

CoreDebug DEMCR: TRCENA Mask

5.27.2.14 CoreDebug_DEMCR_TRCENA_Pos

```
#define CoreDebug_DEMCR_TRCENA_Pos 24
```

CoreDebug DEMCR: TRCENA Position

5.27.2.15 CoreDebug_DEMCR_VC_BUSERR_Msk

```
#define CoreDebug_DEMCR_VC_BUSERR_Msk (1UL << CoreDebug_DEMCR_VC_BUSERR_Pos)
```

CoreDebug DEMCR: VC_BUSERR Mask

5.27.2.16 CoreDebug_DEMCR_VC_BUSERR_Pos

```
#define CoreDebug_DEMCR_VC_BUSERR_Pos 8
```

CoreDebug DEMCR: VC_BUSERR Position

5.27.2.17 CoreDebug_DEMCR_VC_CHKERR_Msk

```
#define CoreDebug_DEMCR_VC_CHKERR_Msk (1UL << CoreDebug_DEMCR_VC_CHKERR_Pos)
```

CoreDebug DEMCR: VC_CHKERR Mask

5.27.2.18 CoreDebug_DEMCR_VC_CHKERR_Pos

```
#define CoreDebug_DEMCR_VC_CHKERR_Pos 6
```

CoreDebug DEMCR: VC_CHKERR Position

5.27.2.19 CoreDebug_DEMCR_VC_CORERESET_Msk

```
#define CoreDebug_DEMCR_VC_CORERESET_Msk (1UL /*<< CoreDebug_DEMCR_VC_CORERESET_Pos*/)
```

CoreDebug DEMCR: VC_CORERESET Mask

5.27.2.20 CoreDebug_DEMCR_VC_CORERESET_Pos

```
#define CoreDebug_DEMCR_VC_CORERESET_Pos 0
```

CoreDebug DEMCR: VC_CORERESET Position

5.27.2.21 CoreDebug_DEMCR_VC_HARDERR_Msk

```
#define CoreDebug_DEMCR_VC_HARDERR_Msk (1UL << CoreDebug_DEMCR_VC_HARDERR_Pos)
```

CoreDebug DEMCR: VC_HARDERR Mask

5.27.2.22 CoreDebug_DEMCR_VC_HARDERR_Pos

```
#define CoreDebug_DEMCR_VC_HARDERR_Pos 10
```

CoreDebug DEMCR: VC_HARDERR Position

5.27.2.23 CoreDebug_DEMCR_VC_INTERR_Msk

```
#define CoreDebug_DEMCR_VC_INTERR_Msk (1UL << CoreDebug_DEMCR_VC_INTERR_Pos)
```

CoreDebug DEMCR: VC_INTERR Mask

5.27.2.24 CoreDebug_DEMCR_VC_INTERR_Pos

```
#define CoreDebug_DEMCR_VC_INTERR_Pos 9
```

CoreDebug DEMCR: VC_INTERR Position

5.27.2.25 CoreDebug_DEMCR_VC_MMERR_Msk

```
#define CoreDebug_DEMCR_VC_MMERR_Msk (1UL << CoreDebug_DEMCR_VC_MMERR_Pos)
```

CoreDebug DEMCR: VC_MMERR Mask

5.27.2.26 CoreDebug_DEMCR_VC_MMERR_Pos

```
#define CoreDebug_DEMCR_VC_MMERR_Pos 4
```

CoreDebug DEMCR: VC_MMERR Position

5.27.2.27 CoreDebug_DEMCR_VC_NOCPERR_Msk

```
#define CoreDebug_DEMCR_VC_NOCPERR_Msk (1UL << CoreDebug_DEMCR_VC_NOCPERR_Pos)
```

CoreDebug DEMCR: VC_NOCPERR Mask

5.27.2.28 CoreDebug_DEMCR_VC_NOCPERR_Pos

```
#define CoreDebug_DEMCR_VC_NOCPERR_Pos 5
```

CoreDebug DEMCR: VC_NOCPERR Position

5.27.2.29 CoreDebug_DEMCR_VC_STATERR_Msk

```
#define CoreDebug_DEMCR_VC_STATERR_Msk (1UL << CoreDebug_DEMCR_VC_STATERR_Pos)
```

CoreDebug DEMCR: VC_STATERR Mask

5.27.2.30 CoreDebug_DEMCR_VC_STATERR_Pos

```
#define CoreDebug_DEMCR_VC_STATERR_Pos 7
```

CoreDebug DEMCR: VC_STATERR Position

5.27.2.31 CoreDebug_DHCSR_C_DEBUGEN_Msk

```
#define CoreDebug_DHCSR_C_DEBUGEN_Msk (1UL /*<< CoreDebug_DHCSR_C_DEBUGEN_Pos*/)
```

CoreDebug DHCSR: C_DEBUGEN Mask

5.27.2.32 CoreDebug_DHCSR_C_DEBUGEN_Pos

```
#define CoreDebug_DHCSR_C_DEBUGEN_Pos 0
```

CoreDebug DHCSR: C_DEBUGEN Position

5.27.2.33 CoreDebug_DHCSR_C_HALT_Msk

```
#define CoreDebug_DHCSR_C_HALT_Msk (1UL << CoreDebug_DHCSR_C_HALT_Pos)
```

CoreDebug DHCSR: C_HALT Mask

5.27.2.34 CoreDebug_DHCSR_C_HALT_Pos

```
#define CoreDebug_DHCSR_C_HALT_Pos 1
```

CoreDebug DHCSR: C_HALT Position

5.27.2.35 CoreDebug_DHCSR_C_MASKINTS_Msk

```
#define CoreDebug_DHCSR_C_MASKINTS_Msk (1UL << CoreDebug_DHCSR_C_MASKINTS_Pos)
```

CoreDebug DHCSR: C_MASKINTS Mask

5.27.2.36 CoreDebug_DHCSR_C_MASKINTS_Pos

```
#define CoreDebug_DHCSR_C_MASKINTS_Pos 3
```

CoreDebug DHCSR: C_MASKINTS Position

5.27.2.37 CoreDebug_DHCSR_C_SNAPSTALL_Msk

```
#define CoreDebug_DHCSR_C_SNAPSTALL_Msk (1UL << CoreDebug_DHCSR_C_SNAPSTALL_Pos)
```

CoreDebug DHCSR: C_SNAPSTALL Mask

5.27.2.38 CoreDebug_DHCSR_C_SNAPSTALL_Pos

```
#define CoreDebug_DHCSR_C_SNAPSTALL_Pos 5
```

CoreDebug DHCSR: C_SNAPSTALL Position

5.27.2.39 CoreDebug_DHCSR_C_STEP_Msk

```
#define CoreDebug_DHCSR_C_STEP_Msk (1UL << CoreDebug_DHCSR_C_STEP_Pos)
```

CoreDebug DHCSR: C_STEP Mask

5.27.2.40 CoreDebug_DHCSR_C_STEP_Pos

```
#define CoreDebug_DHCSR_C_STEP_Pos 2
```

CoreDebug DHCSR: C_STEP Position

5.27.2.41 CoreDebug_DHCSR_DBGKEY_Msk

```
#define CoreDebug_DHCSR_DBGKEY_Msk (0xFFFFFUL << CoreDebug_DHCSR_DBGKEY_Pos)
```

CoreDebug DHCSR: DBGKEY Mask

5.27.2.42 CoreDebug_DHCSR_DBGKEY_Pos

```
#define CoreDebug_DHCSR_DBGKEY_Pos 16
```

CoreDebug DHCSR: DBGKEY Position

5.27.2.43 CoreDebug_DHCSR_S_HALT_Msk

```
#define CoreDebug_DHCSR_S_HALT_Msk (1UL << CoreDebug_DHCSR_S_HALT_Pos)
```

CoreDebug DHCSR: S_HALT Mask

5.27.2.44 CoreDebug_DHCSR_S_HALT_Pos

```
#define CoreDebug_DHCSR_S_HALT_Pos 17
```

CoreDebug DHCSR: S_HALT Position

5.27.2.45 CoreDebug_DHCSR_S_LOCKUP_Msk

```
#define CoreDebug_DHCSR_S_LOCKUP_Msk (1UL << CoreDebug_DHCSR_S_LOCKUP_Pos)
```

CoreDebug DHCSR: S_LOCKUP Mask

5.27.2.46 CoreDebug_DHCSR_S_LOCKUP_Pos

```
#define CoreDebug_DHCSR_S_LOCKUP_Pos 19
```

CoreDebug DHCSR: S_LOCKUP Position

5.27.2.47 CoreDebug_DHCSR_S_REGRDY_Msk

```
#define CoreDebug_DHCSR_S_REGRDY_Msk (1UL << CoreDebug_DHCSR_S_REGRDY_Pos)
```

CoreDebug DHCSR: S_REGRDY Mask

5.27.2.48 CoreDebug_DHCSR_S_REGRDY_Pos

```
#define CoreDebug_DHCSR_S_REGRDY_Pos 16
```

CoreDebug DHCSR: S_REGRDY Position

5.27.2.49 CoreDebug_DHCSR_S_RESET_ST_Msk

```
#define CoreDebug_DHCSR_S_RESET_ST_Msk (1UL << CoreDebug_DHCSR_S_RESET_ST_Pos)
```

CoreDebug DHCSR: S_RESET_ST Mask

5.27.2.50 CoreDebug_DHCSR_S_RESET_ST_Pos

```
#define CoreDebug_DHCSR_S_RESET_ST_Pos 25
```

CoreDebug DHCSR: S_RESET_ST Position

5.27.2.51 CoreDebug_DHCSR_S_RETIRE_ST_Msk

```
#define CoreDebug_DHCSR_S_RETIRE_ST_Msk (1UL << CoreDebug_DHCSR_S_RETIRE_ST_Pos)
```

CoreDebug DHCSR: S_RETIRE_ST Mask

5.27.2.52 CoreDebug_DHCSR_S_RETIRE_ST_Pos

```
#define CoreDebug_DHCSR_S_RETIRE_ST_Pos 24
```

CoreDebug DHCSR: S_RETIRE_ST Position

5.27.2.53 CoreDebug_DHCSR_S_SLEEP_Msk

```
#define CoreDebug_DHCSR_S_SLEEP_Msk (1UL << CoreDebug_DHCSR_S_SLEEP_Pos)
```

CoreDebug DHCSR: S_SLEEP Mask

5.27.2.54 CoreDebug_DHCSR_S_SLEEP_Pos

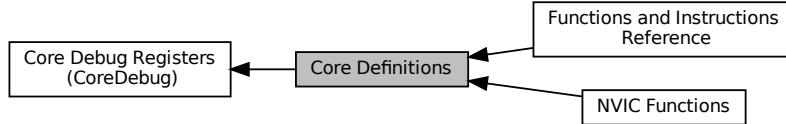
```
#define CoreDebug_DHCSR_S_SLEEP_Pos 18
```

CoreDebug DHCSR: S_SLEEP Position

5.28 Core Definitions

Definitions for base addresses, unions, and structures.

Collaboration diagram for Core Definitions:



Modules

- [Functions and Instructions Reference](#)
- [NVIC Functions](#)

Functions that manage interrupts and exceptions via the NVIC.

- #define SCS_BASE (0xE000E000UL)
- #define ITM_BASE (0xE0000000UL)
- #define DWT_BASE (0xE0001000UL)
- #define TPI_BASE (0xE0040000UL)
- #define CoreDebug_BASE (0xE000EDF0UL)
- #define SysTick_BASE (SCS_BASE + 0x0010UL)
- #define NVIC_BASE (SCS_BASE + 0x0100UL)
- #define SCB_BASE (SCS_BASE + 0x0D00UL)
- #define SCnSCB ((SCnSCB_Type *) SCS_BASE)
- #define SCB ((SCB_Type *) SCB_BASE)
- #define SysTick ((SysTick_Type *) SysTick_BASE)
- #define NVIC ((NVIC_Type *) NVIC_BASE)
- #define ITM ((ITM_Type *) ITM_BASE)
- #define DWT ((DWT_Type *) DWT_BASE)
- #define TPI ((TPI_Type *) TPI_BASE)
- #define CoreDebug ((CoreDebug_Type *) CoreDebug_BASE)

5.28.1 Detailed Description

Definitions for base addresses, unions, and structures.

5.28.2 Macro Definition Documentation

5.28.2.1 CoreDebug

```
#define CoreDebug ((CoreDebug_Type *) CoreDebug_BASE)
```

Core Debug configuration struct

5.28.2.2 CoreDebug_BASE

```
#define CoreDebug_BASE (0xE000EDF0UL)
```

Core Debug Base Address

5.28.2.3 DWT

```
#define DWT ((DWT_Type *) DWT_BASE )
```

DWT configuration struct

5.28.2.4 DWT_BASE

```
#define DWT_BASE (0xE0001000UL)
```

DWT Base Address

5.28.2.5 ITM

```
#define ITM ((ITM_Type *) ITM_BASE )
```

ITM configuration struct

5.28.2.6 ITM_BASE

```
#define ITM_BASE (0xE0000000UL)
```

ITM Base Address

5.28.2.7 NVIC

```
#define NVIC ((NVIC_Type *) NVIC_BASE )
```

NVIC configuration struct

5.28.2.8 NVIC_BASE

```
#define NVIC_BASE (SCS_BASE + 0x0100UL)
```

NVIC Base Address

5.28.2.9 SCB

```
#define SCB ((SCB_Type *) SCB_BASE )
```

SCB configuration struct

5.28.2.10 SCB_BASE

```
#define SCB_BASE (SCS_BASE + 0x0D00UL)
```

System Control Block Base Address

5.28.2.11 SCnSCB

```
#define SCnSCB ((SCnSCB_Type *) SCS_BASE )
```

System control Register not in SCB

5.28.2.12 SCS_BASE

```
#define SCS_BASE (0xE000E000UL)
```

System Control Space Base Address

5.28.2.13 SysTick

```
#define SysTick ((SysTick_Type *) SysTick_BASE )
```

SysTick configuration struct

5.28.2.14 SysTick_BASE

```
#define SysTick_BASE (SCS_BASE + 0x0010UL)
```

SysTick Base Address

5.28.2.15 TPI

```
#define TPI ((TPI_Type *) TPI_BASE )
```

TPI configuration struct

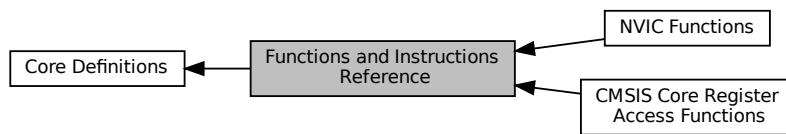
5.28.2.16 TPI_BASE

```
#define TPI_BASE (0xE0040000UL)
```

TPI Base Address

5.29 Functions and Instructions Reference

Collaboration diagram for Functions and Instructions Reference:



Modules

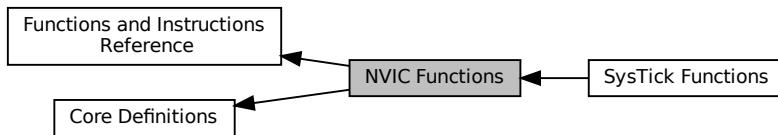
- [NVIC Functions](#)
Functions that manage interrupts and exceptions via the NVIC.
- [CMSIS Core Register Access Functions](#)

5.29.1 Detailed Description

5.30 NVIC Functions

Functions that manage interrupts and exceptions via the NVIC.

Collaboration diagram for NVIC Functions:



Modules

- [SysTick Functions](#)

Functions that configure the System.

- `__STATIC_INLINE void NVIC_SetPriorityGrouping (uint32_t PriorityGroup)`
Set Priority Grouping.
- `__STATIC_INLINE uint32_t NVIC_GetPriorityGrouping (void)`
Get Priority Grouping.
- `__STATIC_INLINE void NVIC_EnableIRQ (IRQn_Type IRQn)`
Enable External Interrupt.
- `__STATIC_INLINE void NVIC_DisableIRQ (IRQn_Type IRQn)`
Disable External Interrupt.
- `__STATIC_INLINE uint32_t NVIC_GetPendingIRQ (IRQn_Type IRQn)`
Get Pending Interrupt.
- `__STATIC_INLINE void NVIC_SetPendingIRQ (IRQn_Type IRQn)`
Set Pending Interrupt.
- `__STATIC_INLINE void NVIC_ClearPendingIRQ (IRQn_Type IRQn)`
Clear Pending Interrupt.
- `__STATIC_INLINE uint32_t NVIC_GetActive (IRQn_Type IRQn)`
Get Active Interrupt.
- `__STATIC_INLINE void NVIC_SetPriority (IRQn_Type IRQn, uint32_t priority)`
Set Interrupt Priority.
- `__STATIC_INLINE uint32_t NVIC_GetPriority (IRQn_Type IRQn)`
Get Interrupt Priority.
- `__STATIC_INLINE uint32_t NVIC_EncodePriority (uint32_t PriorityGroup, uint32_t PreemptPriority, uint32_t SubPriority)`
Encode Priority.
- `__STATIC_INLINE void NVIC_DecodePriority (uint32_t Priority, uint32_t PriorityGroup, uint32_t *pPreemptPriority, uint32_t *pSubPriority)`
Decode Priority.
- `__STATIC_INLINE void NVIC_SystemReset (void)`
System Reset.

5.30.1 Detailed Description

Functions that manage interrupts and exceptions via the NVIC.

5.30.2 Function Documentation

5.30.2.1 NVIC_ClearPendingIRQ()

```
__STATIC_INLINE void NVIC_ClearPendingIRQ (
    IRQn_Type IRQn )
```

Clear Pending Interrupt.

The function clears the pending bit of an external interrupt.

Parameters

in	<i>IRQn</i>	External interrupt number. Value cannot be negative.
----	-------------	--

5.30.2.2 NVIC_DecodePriority()

```
__STATIC_INLINE void NVIC_DecodePriority (
    uint32_t Priority,
    uint32_t PriorityGroup,
    uint32_t * pPreemptPriority,
    uint32_t * pSubPriority )
```

Decode Priority.

The function decodes an interrupt priority value with a given priority group to preemptive priority value and subpriority value. In case of a conflict between priority grouping and available priority bits (`__NVIC_PRIO_BITS`) the smallest possible priority group is set.

Parameters

in	<i>Priority</i>	Priority value, which can be retrieved with the function NVIC_GetPriority() .
in	<i>PriorityGroup</i>	Used priority group.
out	<i>pPreemptPriority</i>	Preemptive priority value (starting from 0).
out	<i>pSubPriority</i>	Subpriority value (starting from 0).

5.30.2.3 NVIC_DisableIRQ()

```
__STATIC_INLINE void NVIC_DisableIRQ (
    IRQn_Type IRQn )
```

Disable External Interrupt.

The function disables a device-specific interrupt in the NVIC interrupt controller.

Parameters

in	<i>IRQn</i>	External interrupt number. Value cannot be negative.
----	-------------	--

5.30.2.4 NVIC_EnableIRQ()

```
__STATIC_INLINE void NVIC_EnableIRQ (
    IRQn_Type IRQn )
```

Enable External Interrupt.

The function enables a device-specific interrupt in the NVIC interrupt controller.

Parameters

in	<i>IRQn</i>	External interrupt number. Value cannot be negative.
----	-------------	--

5.30.2.5 NVIC_EncodePriority()

```
__STATIC_INLINE uint32_t NVIC_EncodePriority (
    uint32_t PriorityGroup,
    uint32_t PreemptPriority,
    uint32_t SubPriority )
```

Encode Priority.

The function encodes the priority for an interrupt with the given priority group, preemptive priority value, and subpriority value. In case of a conflict between priority grouping and available priority bits (`__NVIC_PRIO_BITS`), the smallest possible priority group is set.

Parameters

in	<i>PriorityGroup</i>	Used priority group.
in	<i>PreemptPriority</i>	Preemptive priority value (starting from 0).
in	<i>SubPriority</i>	Subpriority value (starting from 0).

Returns

Encoded priority. Value can be used in the function [NVIC_SetPriority\(\)](#).

5.30.2.6 NVIC_GetActive()

```
__STATIC_INLINE uint32_t NVIC_GetActive (
    IRQn_Type IRQn )
```

Get Active Interrupt.

The function reads the active register in NVIC and returns the active bit.

Parameters

in	IRQn	Interrupt number.
----	------	-------------------

Returns

0 Interrupt status is not active.

1 Interrupt status is active.

5.30.2.7 NVIC_GetPendingIRQ()

```
__STATIC_INLINE uint32_t NVIC_GetPendingIRQ (
    IRQn_Type IRQn )
```

Get Pending Interrupt.

The function reads the pending register in the NVIC and returns the pending bit for the specified interrupt.

Parameters

in	IRQn	Interrupt number.
----	------	-------------------

Returns

0 Interrupt status is not pending.

1 Interrupt status is pending.

5.30.2.8 NVIC_GetPriority()

```
__STATIC_INLINE uint32_t NVIC_GetPriority (
    IRQn_Type IRQn )
```

Get Interrupt Priority.

The function reads the priority of an interrupt. The interrupt number can be positive to specify an external (device specific) interrupt, or negative to specify an internal (core) interrupt.

Parameters

in	<i>IRQn</i>	Interrupt number.
----	-------------	-------------------

Returns

Interrupt Priority. Value is aligned automatically to the implemented priority bits of the microcontroller.

5.30.2.9 NVIC_GetPriorityGrouping()

```
__STATIC_INLINE uint32_t NVIC_GetPriorityGrouping (
    void )
```

Get Priority Grouping.

The function reads the priority grouping field from the NVIC Interrupt Controller.

Returns

Priority grouping field (SCB->AIRCR [10:8] PRIGROUP field).

5.30.2.10 NVIC_SetPendingIRQ()

```
__STATIC_INLINE void NVIC_SetPendingIRQ (
    IRQn_Type IRQn )
```

Set Pending Interrupt.

The function sets the pending bit of an external interrupt.

Parameters

in	<i>IRQn</i>	Interrupt number. Value cannot be negative.
----	-------------	---

5.30.2.11 NVIC_SetPriority()

```
__STATIC_INLINE void NVIC_SetPriority (
```

```
IRQn_Type IRQn,
uint32_t priority )
```

Set Interrupt Priority.

The function sets the priority of an interrupt.

Note

The priority cannot be set for every core interrupt.

Parameters

in	<i>IRQn</i>	Interrupt number.
in	<i>priority</i>	Priority to set.

5.30.2.12 NVIC_SetPriorityGrouping()

```
__STATIC_INLINE void NVIC_SetPriorityGrouping (
    uint32_t PriorityGroup )
```

Set Priority Grouping.

The function sets the priority grouping field using the required unlock sequence. The parameter PriorityGroup is assigned to the field SCB->AIRCR [10:8] PRIGROUP field. Only values from 0..7 are used. In case of a conflict between priority grouping and available priority bits (`__NVIC_PRIO_BITS`), the smallest possible priority group is set.

Parameters

in	<i>PriorityGroup</i>	Priority grouping field.
----	----------------------	--------------------------

5.30.2.13 NVIC_SystemReset()

```
__STATIC_INLINE void NVIC_SystemReset (
    void )
```

System Reset.

The function initiates a system reset request to reset the MCU.

5.31 SysTick Functions

Functions that configure the System.

Collaboration diagram for SysTick Functions:



Modules

- [ITM Functions](#)
Functions that access the ITM debug interface.
- `__STATIC_INLINE uint32_t SysTick_Config (uint32_t ticks)`
System Tick Configuration.

5.31.1 Detailed Description

Functions that configure the System.

5.31.2 Function Documentation

5.31.2.1 [SysTick_Config\(\)](#)

```
__STATIC_INLINE uint32_t SysTick_Config (
    uint32_t ticks )
```

System Tick Configuration.

The function initializes the System Timer and its interrupt, and starts the System Tick Timer. Counter is in free running mode to generate periodic interrupts.

Parameters

in	<i>ticks</i>	Number of ticks between two interrupts.
----	--------------	---

Returns

- 0 Function succeeded.
- 1 Function failed.

Note

When the variable `__Vendor_SysTickConfig` is set to 1, then the function `SysTick_Config` is not included. In this case, the file `device.h` must contain a vendor-specific implementation of this function.

5.32 ITM Functions

Functions that access the ITM debug interface.

Collaboration diagram for ITM Functions:



Variables

- `uint32_t APSR_Type::_reserved0:16`
- `uint32_t APSR_Type::GE:4`
- `uint32_t APSR_Type::_reserved1:7`
- `uint32_t APSR_Type::Q:1`
- `uint32_t APSR_Type::V:1`
- `uint32_t APSR_Type::C:1`
- `uint32_t APSR_Type::Z:1`
- `uint32_t APSR_Type::N:1`
- struct {
 - `uint32_t APSR_Type::_reserved0:16`
 - `uint32_t APSR_Type::GE:4`
 - `uint32_t APSR_Type::_reserved1:7`
 - `uint32_t APSR_Type::Q:1`
 - `uint32_t APSR_Type::V:1`
 - `uint32_t APSR_Type::C:1`
 - `uint32_t APSR_Type::Z:1`
 - `uint32_t APSR_Type::N:1`}
- `} APSR_Type::b`
- `uint32_t APSR_Type::w`
- `uint32_t IPSR_Type::ISR:9`
- `uint32_t IPSR_Type::_reserved0:23`

- struct {
 - uint32_t **IPSR_Type::ISR**:9
 - uint32_t **IPSR_Type::_reserved0**:23
} **IPSR_Type::b**

- uint32_t **IPSR_Type::w**
- uint32_t **xPSR_Type::ISR**:9
- uint32_t **xPSR_Type::_reserved0**:7
- uint32_t **xPSR_Type::GE**:4
- uint32_t **xPSR_Type::_reserved1**:4
- uint32_t **xPSR_Type::T**:1
- uint32_t **xPSR_Type::IT**:2
- uint32_t **xPSR_Type::Q**:1
- uint32_t **xPSR_Type::V**:1
- uint32_t **xPSR_Type::C**:1
- uint32_t **xPSR_Type::Z**:1
- uint32_t **xPSR_Type::N**:1
- struct {
 - uint32_t **xPSR_Type::ISR**:9
 - uint32_t **xPSR_Type::_reserved0**:7
 - uint32_t **xPSR_Type::GE**:4
 - uint32_t **xPSR_Type::_reserved1**:4
 - uint32_t **xPSR_Type::T**:1
 - uint32_t **xPSR_Type::IT**:2
 - uint32_t **xPSR_Type::Q**:1
 - uint32_t **xPSR_Type::V**:1
 - uint32_t **xPSR_Type::C**:1
 - uint32_t **xPSR_Type::Z**:1
 - uint32_t **xPSR_Type::N**:1
} **xPSR_Type::b**

- uint32_t **xPSR_Type::w**
- uint32_t **CONTROL_Type::nPRIV**:1
- uint32_t **CONTROL_Type::SPSEL**:1
- uint32_t **CONTROL_Type::FPCA**:1
- uint32_t **CONTROL_Type::_reserved0**:29
- struct {
 - uint32_t **CONTROL_Type::nPRIV**:1
 - uint32_t **CONTROL_Type::SPSEL**:1
 - uint32_t **CONTROL_Type::FPCA**:1
 - uint32_t **CONTROL_Type::_reserved0**:29
} **CONTROL_Type::b**

- uint32_t **CONTROL_Type::w**
- **_IO** uint32_t **NVIC_Type::ISER** [8]
- uint32_t **NVIC_Type::RESERVED0** [24]
- **_IO** uint32_t **NVIC_Type::ICER** [8]
- uint32_t **NVIC_Type::RSERVED1** [24]
- **_IO** uint32_t **NVIC_Type::ISPR** [8]
- uint32_t **NVIC_Type::RESERVED2** [24]
- **_IO** uint32_t **NVIC_Type::ICPR** [8]
- uint32_t **NVIC_Type::RESERVED3** [24]
- **_IO** uint32_t **NVIC_Type::IABR** [8]
- uint32_t **NVIC_Type::RESERVED4** [56]
- **_IO** uint8_t **NVIC_Type::IP** [240]
- uint32_t **NVIC_Type::RESERVED5** [644]

- `_O uint32_t NVIC_Type::STIR`
- `_I uint32_t SCB_Type::CPUID`
- `_IO uint32_t SCB_Type::ICSR`
- `_IO uint32_t SCB_Type::VTOR`
- `_IO uint32_t SCB_Type::AIRCR`
- `_IO uint32_t SCB_Type::SCR`
- `_IO uint32_t SCB_Type::CCR`
- `_IO uint8_t SCB_Type::SHP [12]`
- `_IO uint32_t SCB_Type::SHCSR`
- `_IO uint32_t SCB_Type::CSR`
- `_IO uint32_t SCB_Type::HFSR`
- `_IO uint32_t SCB_Type::DFSR`
- `_IO uint32_t SCB_Type::MMFAR`
- `_IO uint32_t SCB_Type::BFAR`
- `_IO uint32_t SCB_Type::AFSR`
- `_I uint32_t SCB_Type::PFR [2]`
- `_I uint32_t SCB_Type::DFR`
- `_I uint32_t SCB_Type::ADR`
- `_I uint32_t SCB_Type::MMFR [4]`
- `_I uint32_t SCB_Type::ISAR [5]`
- `uint32_t SCB_Type::RESERVED0 [5]`
- `_IO uint32_t SCB_Type::CPACR`
- `uint32_t SCnSCB_Type::RESERVED0 [1]`
- `_I uint32_t SCnSCB_Type::ICTR`
- `_IO uint32_t SCnSCB_Type::ACTLR`
- `_IO uint32_t SysTick_Type::CTRL`
- `_IO uint32_t SysTick_Type::LOAD`
- `_IO uint32_t SysTick_Type::VAL`
- `_I uint32_t SysTick_Type::CALIB`
- `_O uint8_t ITM_Type::u8`
- `_O uint16_t ITM_Type::u16`
- `_O uint32_t ITM_Type::u32`
- union {
 - `_O uint8_t ITM_Type::u8`
 - `_O uint16_t ITM_Type::u16`
 - `_O uint32_t ITM_Type::u32`}
- `ITM_Type::PORT [32]`
- `uint32_t ITM_Type::RESERVED0 [864]`
- `_IO uint32_t ITM_Type::TER`
- `uint32_t ITM_Type::RESERVED1 [15]`
- `_IO uint32_t ITM_Type::TPR`
- `uint32_t ITM_Type::RESERVED2 [15]`
- `_IO uint32_t ITM_Type::TCR`
- `uint32_t ITM_Type::RESERVED3 [29]`
- `_O uint32_t ITM_Type::IWR`
- `_I uint32_t ITM_Type::IRR`
- `_IO uint32_t ITM_Type::IMCR`
- `uint32_t ITM_Type::RESERVED4 [43]`
- `_O uint32_t ITM_Type::LAR`
- `_I uint32_t ITM_Type::LSR`
- `uint32_t ITM_Type::RESERVED5 [6]`
- `_I uint32_t ITM_Type::PID4`
- `_I uint32_t ITM_Type::PID5`
- `_I uint32_t ITM_Type::PID6`

- `__I uint32_t ITM_Type::PID7`
- `__I uint32_t ITM_Type::PID0`
- `__I uint32_t ITM_Type::PID1`
- `__I uint32_t ITM_Type::PID2`
- `__I uint32_t ITM_Type::PID3`
- `__I uint32_t ITM_Type::CID0`
- `__I uint32_t ITM_Type::CID1`
- `__I uint32_t ITM_Type::CID2`
- `__I uint32_t ITM_Type::CID3`
- `__IO uint32_t DWT_Type::CTRL`
- `__IO uint32_t DWT_Type::CYCCNT`
- `__IO uint32_t DWT_Type::CPICNT`
- `__IO uint32_t DWT_Type::EXCCNT`
- `__IO uint32_t DWT_Type::SLEEPCNT`
- `__IO uint32_t DWT_Type::LSUCNT`
- `__IO uint32_t DWT_Type::FOLDCNT`
- `__I uint32_t DWT_Type::PCSR`
- `__IO uint32_t DWT_Type::COMP0`
- `__IO uint32_t DWT_Type::MASK0`
- `__IO uint32_t DWT_Type::FUNCTION0`
- `uint32_t DWT_Type::RESERVED0 [1]`
- `__IO uint32_t DWT_Type::COMP1`
- `__IO uint32_t DWT_Type::MASK1`
- `__IO uint32_t DWT_Type::FUNCTION1`
- `uint32_t DWT_Type::RESERVED1 [1]`
- `__IO uint32_t DWT_Type::COMP2`
- `__IO uint32_t DWT_Type::MASK2`
- `__IO uint32_t DWT_Type::FUNCTION2`
- `uint32_t DWT_Type::RESERVED2 [1]`
- `__IO uint32_t DWT_Type::COMP3`
- `__IO uint32_t DWT_Type::MASK3`
- `__IO uint32_t DWT_Type::FUNCTION3`
- `__IO uint32_t TPI_Type::SSPSR`
- `__IO uint32_t TPI_Type::CSPSR`
- `uint32_t TPI_Type::RESERVED0 [2]`
- `__IO uint32_t TPI_Type::ACPR`
- `uint32_t TPI_Type::RESERVED1 [55]`
- `__IO uint32_t TPI_Type::SPPR`
- `uint32_t TPI_Type::RESERVED2 [131]`
- `__I uint32_t TPI_Type::FFSR`
- `__IO uint32_t TPI_Type::FFCR`
- `__I uint32_t TPI_Type::FSCR`
- `uint32_t TPI_Type::RESERVED3 [759]`
- `__I uint32_t TPI_Type::TRIGGER`
- `__I uint32_t TPI_Type::FIFO0`
- `__I uint32_t TPI_Type::ITATBCTR2`
- `uint32_t TPI_Type::RESERVED4 [1]`
- `__I uint32_t TPI_Type::ITATBCTR0`
- `__I uint32_t TPI_Type::FIFO1`
- `__IO uint32_t TPI_Type::ITCTRL`
- `uint32_t TPI_Type::RESERVED5 [39]`
- `__IO uint32_t TPI_Type::CLAIMSET`
- `__IO uint32_t TPI_Type::CLAIMCLR`
- `uint32_t TPI_Type::RESERVED7 [8]`
- `__I uint32_t TPI_Type::DEVID`

- `__I uint32_t TPI_Type::DEVTYPE`
- `__IO uint32_t CoreDebug_Type::DHCSR`
- `__O uint32_t CoreDebug_Type::DCRSR`
- `__IO uint32_t CoreDebug_Type::DCRDR`
- `__IO uint32_t CoreDebug_Type::DEMCR`

- volatile int32_t `ITM_RxBuffer`
- `__STATIC_INLINE uint32_t ITM_SendChar (uint32_t ch)`
ITM Send Character.
- `__STATIC_INLINE int32_t ITM_ReceiveChar (void)`
ITM Receive Character.
- `__STATIC_INLINE int32_t ITM_CheckChar (void)`
ITM Check Character.
- `#define ITM_RXBUFFER_EMPTY 0x5AA55AA5`

5.32.1 Detailed Description

Functions that access the ITM debug interface.

5.32.2 Macro Definition Documentation

5.32.2.1 ITM_RXBUFFER_EMPTY

```
#define ITM_RXBUFFER_EMPTY 0x5AA55AA5
```

Value identifying `ITM_RxBuffer` is ready for next character.

5.32.3 Function Documentation

5.32.3.1 ITM_CheckChar()

```
__STATIC_INLINE int32_t ITM_CheckChar (
    void )
```

ITM Check Character.

The function checks whether a character is pending for reading in the variable `ITM_RxBuffer`.

Returns

0 No character available.

1 Character available.

5.32.3.2 ITM_ReceiveChar()

```
__STATIC_INLINE int32_t ITM_ReceiveChar (
    void )
```

ITM Receive Character.

The function inputs a character via the external variable [ITM_RxBuffer](#).

Returns

Received character.
-1 No character pending.

5.32.3.3 ITM_SendChar()

```
__STATIC_INLINE uint32_t ITM_SendChar (
    uint32_t ch )
```

ITM Send Character.

The function transmits a character via the ITM channel 0, and

- Just returns when no debugger is connected that has booked the output.
- Is blocking when a debugger is connected, but the previous character sent has not been transmitted.

Parameters

in	ch	Character to transmit.
----	----	------------------------

Returns

Character to transmit.

5.32.4 Variable Documentation

5.32.4.1 _reserved0 [1/8]

```
uint32_t APSR_Type::_reserved0
```

bit: 0..15 Reserved

5.32.4.2 [2/8]

```
uint32_t { ... } ::_reserved0
```

bit: 0..15 Reserved

5.32.4.3 _reserved0 [3/8]

```
uint32_t IPSR_Type::_reserved0
```

bit: 9..31 Reserved

5.32.4.4 [4/8]

```
uint32_t { ... } ::_reserved0
```

bit: 9..31 Reserved

5.32.4.5 _reserved0 [5/8]

```
uint32_t xPSR_Type::_reserved0
```

bit: 9..15 Reserved

5.32.4.6 [6/8]

```
uint32_t { ... } ::_reserved0
```

bit: 9..15 Reserved

5.32.4.7 _reserved0 [7/8]

```
uint32_t CONTROL_Type::_reserved0
```

bit: 3..31 Reserved

5.32.4.8 [8/8]

```
uint32_t { ... } ::_reserved0
```

bit: 3..31 Reserved

5.32.4.9 _reserved1 [1/4]

```
uint32_t APSR_Type::_reserved1
```

bit: 20..26 Reserved

5.32.4.10 [2/4]

```
uint32_t { ... } ::_reserved1
```

bit: 20..26 Reserved

5.32.4.11 [3/4]

```
uint32_t { ... } ::_reserved1
```

bit: 20..23 Reserved

5.32.4.12 _reserved1 [4/4]

```
uint32_t xPSR_Type::_reserved1
```

bit: 20..23 Reserved

5.32.4.13 ACPR

`__IO uint32_t TPI_Type::ACPR`

Offset: 0x010 (R/W) Asynchronous Clock Prescaler Register

5.32.4.14 ACTLR

`__IO uint32_t SCnSCB_Type::ACTLR`

Offset: 0x008 (R/W) Auxiliary Control Register

5.32.4.15 ADR

`__I uint32_t SCB_Type::ADR`

Offset: 0x04C (R/) Auxiliary Feature Register

5.32.4.16 AFSR

```
__IO uint32_t SCB_Type::AFSR
```

Offset: 0x03C (R/W) Auxiliary Fault Status Register

5.32.4.17 AIRCR

```
__IO uint32_t SCB_Type::AIRCR
```

Offset: 0x00C (R/W) Application Interrupt and Reset Control Register

5.32.4.18 [1/4]

```
struct { ... } APSR_Type::b
```

Structure used for bit access

5.32.4.19 [2/4]

```
struct { ... } IPSR_Type::b
```

Structure used for bit access

5.32.4.20 [3/4]

```
struct { ... } xPSR_Type::b
```

Structure used for bit access

5.32.4.21 [4/4]

```
struct { ... } CONTROL_Type::b
```

Structure used for bit access

5.32.4.22 BFAR

```
__IO uint32_t SCB_Type::BFAR
```

Offset: 0x038 (R/W) BusFault Address Register

5.32.4.23 C [1/4]

```
uint32_t APSR_Type::C
```

bit: 29 Carry condition code flag

5.32.4.24 [2/4]

```
uint32_t { ... } ::C
```

bit: 29 Carry condition code flag

5.32.4.25 C [3/4]

```
uint32_t xPSR_Type::C
```

bit: 29 Carry condition code flag

5.32.4.26 [4/4]

```
uint32_t { ... } ::C
```

bit: 29 Carry condition code flag

5.32.4.27 CALIB

```
__I uint32_t SysTick_Type::CALIB
```

Offset: 0x00C (R/) SysTick Calibration Register

5.32.4.28 CCR

```
__IO uint32_t SCB_Type::CCR
```

Offset: 0x014 (R/W) Configuration Control Register

5.32.4.29 CFSR

```
__IO uint32_t SCB_Type::CFSR
```

Offset: 0x028 (R/W) Configurable Fault Status Register

5.32.4.30 CID0

```
__I uint32_t ITM_Type::CID0
```

Offset: 0xFF0 (R/) ITM Component Identification Register #0

5.32.4.31 CID1

```
__I uint32_t ITM_Type::CID1
```

Offset: 0xFF4 (R/) ITM Component Identification Register #1

5.32.4.32 CID2

```
__I uint32_t ITM_Type::CID2
```

Offset: 0xFF8 (R/) ITM Component Identification Register #2

5.32.4.33 CID3

```
__I uint32_t ITM_Type::CID3
```

Offset: 0xFFC (R/) ITM Component Identification Register #3

5.32.4.34 CLAIMCLR

```
__IO uint32_t TPI_Type::CLAIMCLR
```

Offset: 0xFA4 (R/W) Claim tag clear

5.32.4.35 CLAIMSET

```
__IO uint32_t TPI_Type::CLAIMSET
```

Offset: 0xFA0 (R/W) Claim tag set

5.32.4.36 COMP0

```
__IO uint32_t DWT_Type::COMP0
```

Offset: 0x020 (R/W) Comparator Register 0

5.32.4.37 COMP1

`__IO uint32_t DWT_Type::COMP1`

Offset: 0x030 (R/W) Comparator Register 1

5.32.4.38 COMP2

`__IO uint32_t DWT_Type::COMP2`

Offset: 0x040 (R/W) Comparator Register 2

5.32.4.39 COMP3

`__IO uint32_t DWT_Type::COMP3`

Offset: 0x050 (R/W) Comparator Register 3

5.32.4.40 CPACR

`__IO uint32_t SCB_Type::CPACR`

Offset: 0x088 (R/W) Coprocessor Access Control Register

5.32.4.41 CPICNT

`__IO uint32_t DWT_Type::CPICNT`

Offset: 0x008 (R/W) CPI Count Register

5.32.4.42 CPUID

`__I uint32_t SCB_Type::CPUID`

Offset: 0x000 (R/) CPUID Base Register

5.32.4.43 CSPSR

`__IO uint32_t TPI_Type::CSPSR`

Offset: 0x004 (R/W) Current Parallel Port Size Register

5.32.4.44 CTRL [1/2]

```
__IO uint32_t SysTick_Type::CTRL
```

Offset: 0x000 (R/W) SysTick Control and Status Register

5.32.4.45 CTRL [2/2]

```
__IO uint32_t DWT_Type::CTRL
```

Offset: 0x000 (R/W) Control Register

5.32.4.46 CYCCNT

```
__IO uint32_t DWT_Type::CYCCNT
```

Offset: 0x004 (R/W) Cycle Count Register

5.32.4.47 DCRDR

```
__IO uint32_t CoreDebug_Type::DCRDR
```

Offset: 0x008 (R/W) Debug Core Register Data Register

5.32.4.48 DCRSR

```
__IO uint32_t CoreDebug_Type::DCRSR
```

Offset: 0x004 (/W) Debug Core Register Selector Register

5.32.4.49 DEMCR

```
__IO uint32_t CoreDebug_Type::DEMCR
```

Offset: 0x00C (R/W) Debug Exception and Monitor Control Register

5.32.4.50 DEVID

```
__I uint32_t TPI_Type::DEVID
```

Offset: 0xFC8 (R/) TPIU_DEVID

5.32.4.51 DEVTYPE

`__I uint32_t TPI_Type::DEVTYPE`

Offset: 0xFCC (R/) TPIU_DEVTYPE

5.32.4.52 DFR

`__I uint32_t SCB_Type::DFR`

Offset: 0x048 (R/) Debug Feature Register

5.32.4.53 DFSR

`__IO uint32_t SCB_Type::DFSR`

Offset: 0x030 (R/W) Debug Fault Status Register

5.32.4.54 DHCSR

`__IO uint32_t CoreDebug_Type::DHCSR`

Offset: 0x000 (R/W) Debug Halting Control and Status Register

5.32.4.55 EXCCNT

`__IO uint32_t DWT_Type::EXCCNT`

Offset: 0x00C (R/W) Exception Overhead Count Register

5.32.4.56 FFCR

`__IO uint32_t TPI_Type::FFCR`

Offset: 0x304 (R/W) Formatter and Flush Control Register

5.32.4.57 FFSR

`__I uint32_t TPI_Type::FFSR`

Offset: 0x300 (R/) Formatter and Flush Status Register

5.32.4.58 FIFO0

```
__I uint32_t TPI_Type::FIFO0
```

Offset: 0xEEC (R/) Integration ETM Data

5.32.4.59 FIFO1

```
__I uint32_t TPI_Type::FIFO1
```

Offset: 0xEFC (R/) Integration ITM Data

5.32.4.60 FOLDCNT

```
__IO uint32_t DWT_Type::FOLDCNT
```

Offset: 0x018 (R/W) Folded-instruction Count Register

5.32.4.61 FPCA [1/2]

```
uint32_t CONTROL_Type::FPCA
```

bit: 2 FP extension active flag

5.32.4.62 [2/2]

```
uint32_t { ... } ::FPCA
```

bit: 2 FP extension active flag

5.32.4.63 FSCR

```
__I uint32_t TPI_Type::FSCR
```

Offset: 0x308 (R/) Formatter Synchronization Counter Register

5.32.4.64 FUNCTION0

```
__IO uint32_t DWT_Type::FUNCTION0
```

Offset: 0x028 (R/W) Function Register 0

5.32.4.65 FUNCTION1

`__IO uint32_t DWT_Type::FUNCTION1`

Offset: 0x038 (R/W) Function Register 1

5.32.4.66 FUNCTION2

`__IO uint32_t DWT_Type::FUNCTION2`

Offset: 0x048 (R/W) Function Register 2

5.32.4.67 FUNCTION3

`__IO uint32_t DWT_Type::FUNCTION3`

Offset: 0x058 (R/W) Function Register 3

5.32.4.68 [1/4]

`uint32_t { ... } ::GE`

bit: 16..19 Greater than or Equal flags

5.32.4.69 GE [2/4]

`uint32_t APSR_Type::GE`

bit: 16..19 Greater than or Equal flags

5.32.4.70 GE [3/4]

`uint32_t xPSR_Type::GE`

bit: 16..19 Greater than or Equal flags

5.32.4.71 [4/4]

`uint32_t { ... } ::GE`

bit: 16..19 Greater than or Equal flags

5.32.4.72 HFSR

`__IO uint32_t SCB_Type::HFSR`

Offset: 0x02C (R/W) HardFault Status Register

5.32.4.73 IABR

`__IO uint32_t NVIC_Type::IABR[8]`

Offset: 0x200 (R/W) Interrupt Active bit Register

5.32.4.74 ICER

`__IO uint32_t NVIC_Type::ICER[8]`

Offset: 0x080 (R/W) Interrupt Clear Enable Register

5.32.4.75 ICPR

`__IO uint32_t NVIC_Type::ICPR[8]`

Offset: 0x180 (R/W) Interrupt Clear Pending Register

5.32.4.76 ICSR

`__IO uint32_t SCB_Type::ICSR`

Offset: 0x004 (R/W) Interrupt Control and State Register

5.32.4.77 ICTR

`__I uint32_t SCnSCB_Type::ICTR`

Offset: 0x004 (R/) Interrupt Controller Type Register

5.32.4.78 IMCR

`__IO uint32_t ITM_Type::IMCR`

Offset: 0xF00 (R/W) ITM Integration Mode Control Register

5.32.4.79 IP

```
__IO uint8_t NVIC_Type::IP[240]
```

Offset: 0x300 (R/W) Interrupt Priority Register (8Bit wide)

5.32.4.80 IRR

```
__I uint32_t ITM_Type::IRR
```

Offset: 0xEFC (R/) ITM Integration Read Register

5.32.4.81 ISAR

```
__I uint32_t SCB_Type::ISAR[5]
```

Offset: 0x060 (R/) Instruction Set Attributes Register

5.32.4.82 ISER

```
__IO uint32_t NVIC_Type::ISER[8]
```

Offset: 0x000 (R/W) Interrupt Set Enable Register

5.32.4.83 ISPR

```
__IO uint32_t NVIC_Type::ISPR[8]
```

Offset: 0x100 (R/W) Interrupt Set Pending Register

5.32.4.84 ISR [1/4]

```
uint32_t IPSR_Type::ISR
```

bit: 0.. 8 Exception number

5.32.4.85 [2/4]

```
uint32_t { ... } ::ISR
```

bit: 0.. 8 Exception number

5.32.4.86 ISR [3/4]

```
uint32_t xPSR_Type::ISR
```

bit: 0.. 8 Exception number

5.32.4.87 [4/4]

```
uint32_t { ... } ::ISR
```

bit: 0.. 8 Exception number

5.32.4.88 IT [1/2]

```
uint32_t xPSR_Type::IT
```

bit: 25..26 saved IT state (read 0)

5.32.4.89 [2/2]

```
uint32_t { ... } ::IT
```

bit: 25..26 saved IT state (read 0)

5.32.4.90 ITATBCTR0

```
I uint32_t TPI_Type::ITATBCTR0
```

Offset: 0xEF8 (R/) ITATBCTR0

5.32.4.91 ITATBCTR2

```
I uint32_t TPI_Type::ITATBCTR2
```

Offset: 0xEF0 (R/) ITATBCTR2

5.32.4.92 ITCTRL

```
IO uint32_t TPI_Type::ITCTRL
```

Offset: 0xF00 (R/W) Integration Mode Control

5.32.4.93 ITM_RxBuffer

```
volatile int32_t ITM_RxBuffer [extern]
```

External variable to receive characters.

5.32.4.94 IWR

```
__IO uint32_t ITM_Type::IWR
```

Offset: 0xEF8 (/W) ITM Integration Write Register

5.32.4.95 LAR

```
__IO uint32_t ITM_Type::LAR
```

Offset: 0xFB0 (/W) ITM Lock Access Register

5.32.4.96 LOAD

```
__IO uint32_t SysTick_Type::LOAD
```

Offset: 0x004 (R/W) SysTick Reload Value Register

5.32.4.97 LSR

```
__I uint32_t ITM_Type::LSR
```

Offset: 0xFB4 (R/) ITM Lock Status Register

5.32.4.98 LSUCNT

```
__IO uint32_t DWT_Type::LSUCNT
```

Offset: 0x014 (R/W) LSU Count Register

5.32.4.99 MASK0

```
__IO uint32_t DWT_Type::MASK0
```

Offset: 0x024 (R/W) Mask Register 0

5.32.4.100 MASK1

```
__IO uint32_t DWT_Type::MASK1
```

Offset: 0x034 (R/W) Mask Register 1

5.32.4.101 MASK2

```
__IO uint32_t DWT_Type::MASK2
```

Offset: 0x044 (R/W) Mask Register 2

5.32.4.102 MASK3

```
__IO uint32_t DWT_Type::MASK3
```

Offset: 0x054 (R/W) Mask Register 3

5.32.4.103 MMFAR

```
__IO uint32_t SCB_Type::MMFAR
```

Offset: 0x034 (R/W) MemManage Fault Address Register

5.32.4.104 MMFR

```
__I uint32_t SCB_Type::MMFR[4]
```

Offset: 0x050 (R/) Memory Model Feature Register

5.32.4.105 [1/4]

```
uint32_t { ... } ::N
```

bit: 31 Negative condition code flag

5.32.4.106 N [2/4]

```
uint32_t APSR_Type::N
```

bit: 31 Negative condition code flag

5.32.4.107 [3/4]

```
uint32_t { ... } ::N
```

bit: 31 Negative condition code flag

5.32.4.108 N [4/4]

```
uint32_t xPSR_Type::N
```

bit: 31 Negative condition code flag

5.32.4.109 nPRIV [1/2]

```
uint32_t CONTROL_Type::nPRIV
```

bit: 0 Execution privilege in Thread mode

5.32.4.110 [2/2]

```
uint32_t { ... } ::nPRIV
```

bit: 0 Execution privilege in Thread mode

5.32.4.111 PCSR

[__I](#) uint32_t DWT_Type::PCSR

Offset: 0x01C (R/) Program Counter Sample Register

5.32.4.112 PFR

[__I](#) uint32_t SCB_Type::PFR[2]

Offset: 0x040 (R/) Processor Feature Register

5.32.4.113 PID0

[__I](#) uint32_t ITM_Type::PID0

Offset: 0xFE0 (R/) ITM Peripheral Identification Register #0

5.32.4.114 PID1

`__I uint32_t ITM_Type::PID1`

Offset: 0xFE4 (R/) ITM Peripheral Identification Register #1

5.32.4.115 PID2

`__I uint32_t ITM_Type::PID2`

Offset: 0xFE8 (R/) ITM Peripheral Identification Register #2

5.32.4.116 PID3

`__I uint32_t ITM_Type::PID3`

Offset: 0xFEC (R/) ITM Peripheral Identification Register #3

5.32.4.117 PID4

`__I uint32_t ITM_Type::PID4`

Offset: 0xFD0 (R/) ITM Peripheral Identification Register #4

5.32.4.118 PID5

`__I uint32_t ITM_Type::PID5`

Offset: 0xFD4 (R/) ITM Peripheral Identification Register #5

5.32.4.119 PID6

`__I uint32_t ITM_Type::PID6`

Offset: 0xFD8 (R/) ITM Peripheral Identification Register #6

5.32.4.120 PID7

`__I uint32_t ITM_Type::PID7`

Offset: 0xFDC (R/) ITM Peripheral Identification Register #7

5.32.4.121

`__O { ... } ITM_Type::PORT[32]`

Offset: 0x000 (/W) ITM Stimulus Port Registers

5.32.4.122 [1/4]

```
uint32_t { ... } ::Q
```

bit: 27 Saturation condition flag

5.32.4.123 Q [2/4]

```
uint32_t APSR_Type::Q
```

bit: 27 Saturation condition flag

5.32.4.124 [3/4]

```
uint32_t { ... } ::Q
```

bit: 27 Saturation condition flag

5.32.4.125 Q [4/4]

```
uint32_t xPSR_Type::Q
```

bit: 27 Saturation condition flag

5.32.4.126 SCR

```
IO uint32_t SCB_Type::SCR
```

Offset: 0x010 (R/W) System Control Register

5.32.4.127 SHCSR

```
IO uint32_t SCB_Type::SHCSR
```

Offset: 0x024 (R/W) System Handler Control and State Register

5.32.4.128 SHP

```
IO uint8_t SCB_Type::SHP[12]
```

Offset: 0x018 (R/W) System Handlers Priority Registers (4-7, 8-11, 12-15)

5.32.4.129 SLEEPCNT

```
__IO uint32_t DWT_Type::SLEEPCNT
```

Offset: 0x010 (R/W) Sleep Count Register

5.32.4.130 SPPR

```
__IO uint32_t TPI_Type::SPPR
```

Offset: 0x0F0 (R/W) Selected Pin Protocol Register

5.32.4.131 SPSEL [1/2]

```
uint32_t CONTROL_Type::SPSEL
```

bit: 1 Stack to be used

5.32.4.132 [2/2]

```
uint32_t { ... } ::SPSEL
```

bit: 1 Stack to be used

5.32.4.133 SSPSR

```
__IO uint32_t TPI_Type::SSPSR
```

Offset: 0x000 (R/) Supported Parallel Port Size Register

5.32.4.134 STIR

```
__O uint32_t NVIC_Type::STIR
```

Offset: 0xE00 (/W) Software Trigger Interrupt Register

5.32.4.135 [1/2]

```
uint32_t { ... } ::T
```

bit: 24 Thumb bit (read 0)

5.32.4.136 T [2/2]

```
uint32_t xPSR_Type::T
```

bit: 24 Thumb bit (read 0)

5.32.4.137 TCR

```
__IO uint32_t ITM_Type::TCR
```

Offset: 0xE80 (R/W) ITM Trace Control Register

5.32.4.138 TER

```
__IO uint32_t ITM_Type::TER
```

Offset: 0xE00 (R/W) ITM Trace Enable Register

5.32.4.139 TPR

```
__IO uint32_t ITM_Type::TPR
```

Offset: 0xE40 (R/W) ITM Trace Privilege Register

5.32.4.140 TRIGGER

```
__I uint32_t TPI_Type::TRIGGER
```

Offset: 0xEE8 (R/) TRIGGER

5.32.4.141 u16 [1/2]

```
__O uint16_t ITM_Type::u16
```

Offset: 0x000 (/W) ITM Stimulus Port 16-bit

5.32.4.142 [2/2]

```
__O { ... } ::u16
```

Offset: 0x000 (/W) ITM Stimulus Port 16-bit

5.32.4.143 u32 [1/2]

`__O uint32_t ITM_Type::u32`

Offset: 0x000 (/W) ITM Stimulus Port 32-bit

5.32.4.144 [2/2]

`__O { ... } ::u32`

Offset: 0x000 (/W) ITM Stimulus Port 32-bit

5.32.4.145 [1/2]

`__O { ... } ::u8`

Offset: 0x000 (/W) ITM Stimulus Port 8-bit

5.32.4.146 u8 [2/2]

`__O uint8_t ITM_Type::u8`

Offset: 0x000 (/W) ITM Stimulus Port 8-bit

5.32.4.147 [1/4]

`uint32_t { ... } ::V`

bit: 28 Overflow condition code flag

5.32.4.148 V [2/4]

`uint32_t APSR_Type::V`

bit: 28 Overflow condition code flag

5.32.4.149 [3/4]

`uint32_t { ... } ::V`

bit: 28 Overflow condition code flag

5.32.4.150 V [4/4]

```
uint32_t xPSR_Type::V
```

bit: 28 Overflow condition code flag

5.32.4.151 VAL

```
__IO uint32_t SysTick_Type::VAL
```

Offset: 0x008 (R/W) SysTick Current Value Register

5.32.4.152 VTOR

```
__IO uint32_t SCB_Type::VTOR
```

Offset: 0x008 (R/W) Vector Table Offset Register

5.32.4.153 w [1/4]

```
uint32_t APSR_Type::w
```

Type used for word access

5.32.4.154 w [2/4]

```
uint32_t IPSR_Type::w
```

Type used for word access

5.32.4.155 w [3/4]

```
uint32_t xPSR_Type::w
```

Type used for word access

5.32.4.156 w [4/4]

```
uint32_t CONTROL_Type::w
```

Type used for word access

5.32.4.157 Z [1/4]

```
uint32_t APSR_Type::Z
```

bit: 30 Zero condition code flag

5.32.4.158 [2/4]

```
uint32_t { ... } ::Z
```

bit: 30 Zero condition code flag

5.32.4.159 [3/4]

```
uint32_t { ... } ::Z
```

bit: 30 Zero condition code flag

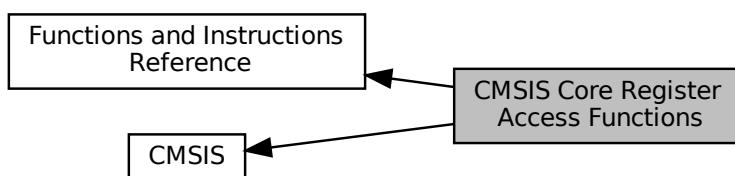
5.32.4.160 Z [4/4]

```
uint32_t xPSR_Type::Z
```

bit: 30 Zero condition code flag

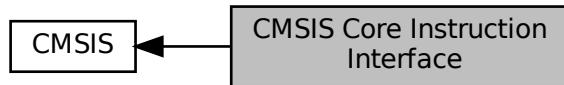
5.33 CMSIS Core Register Access Functions

Collaboration diagram for CMSIS Core Register Access Functions:



5.34 CMSIS Core Instruction Interface

Collaboration diagram for CMSIS Core Instruction Interface:



Access to dedicated instructions

5.35 CMSIS SIMD Intrinsics

Collaboration diagram for CMSIS SIMD Intrinsics:



Access to dedicated SIMD instructions

5.36 Platform Standard Types

Defines standard platform types for STM32F4xx microcontrollers.

Macros

- #define `CPU_TYPE_8` 8
CPU Type Definitions.
- #define `CPU_TYPE_16` 16
- #define `CPU_TYPE_32` 32
- #define `CPU_TYPE_64` 64
- #define `MSB_FIRST` 0
Bit Order Definitions.
- #define `LSB_FIRST` 1

- #define HIGH_BYTE_FIRST 0
 - Byte Order Definitions.*
- #define LOW_BYTE_FIRST 1
- #define CPU_TYPE CPU_TYPE_32
 - Selected CPU Characteristics.*
- #define CPU_BIT_ORDER LSB_FIRST
- #define CPU_BYTE_ORDER LOW_BYTE_FIRST
- #define TRUE 1
 - Boolean Constants.*
- #define FALSE 0
- #define SET TRUE
- #define RESET FALSE
- #define FLAG_SET SET
- #define FLAG_RESET RESET

Typedefs

- typedef unsigned char boolean
 - Standard Data Types.*
- typedef signed char sint8
- typedef unsigned char uint8
- typedef signed short sint16
- typedef unsigned short uint16
- typedef signed long sint32
- typedef signed long long sint64
- typedef unsigned long uint32
- typedef unsigned long long uint64
- typedef unsigned long uint8_least
 - Least Type Definitions.*
- typedef unsigned long uint16_least
- typedef unsigned long uint32_least
- typedef signed long sint8_least
- typedef signed long sint16_least
- typedef signed long sint32_least
- typedef float float32
 - Floating-Point Types.*
- typedef double float64
- typedef void * VoidPtr
 - Pointer Types.*
- typedef const void * ConstVoidPtr
- typedef volatile unsigned char vuint8_t
 - Volatile Types.*
- typedef volatile unsigned short vuint16_t
- typedef volatile unsigned long vuint32_t

5.36.1 Detailed Description

Defines standard platform types for STM32F4xx microcontrollers.

5.36.2 Macro Definition Documentation

5.36.2.1 CPU_BIT_ORDER

```
#define CPU_BIT_ORDER LSB_FIRST
```

Bit order: Least Significant Bit first

5.36.2.2 CPU_BYTE_ORDER

```
#define CPU_BYTE_ORDER LOW_BYTE_FIRST
```

Byte order: Low byte first

5.36.2.3 CPU_TYPE

```
#define CPU_TYPE CPU_TYPE_32
```

Selected CPU Characteristics.

Defines the CPU characteristics used in this file. 32-bit CPU

5.36.2.4 CPU_TYPE_16

```
#define CPU_TYPE_16 16
```

16-bit CPU

5.36.2.5 CPU_TYPE_32

```
#define CPU_TYPE_32 32
```

32-bit CPU

5.36.2.6 CPU_TYPE_64

```
#define CPU_TYPE_64 64
```

64-bit CPU

5.36.2.7 CPU_TYPE_8

```
#define CPU_TYPE_8 8
```

CPU Type Definitions.

Defines the different CPU types supported. 8-bit CPU

5.36.2.8 FALSE

```
#define FALSE 0
```

Boolean false

5.36.2.9 FLAG_RESET

```
#define FLAG_RESET RESET
```

Flag reset macro

5.36.2.10 FLAG_SET

```
#define FLAG_SET SET
```

Flag set macro

5.36.2.11 HIGH_BYTE_FIRST

```
#define HIGH_BYTE_FIRST 0
```

Byte Order Definitions.

Defines the byte order conventions. High byte first

5.36.2.12 LOW_BYTE_FIRST

```
#define LOW_BYTE_FIRST 1
```

Low byte first

5.36.2.13 LSB_FIRST

```
#define LSB_FIRST 1
```

Least Significant Bit first

5.36.2.14 MSB_FIRST

```
#define MSB_FIRST 0
```

Bit Order Definitions.

Defines the bit order conventions. Most Significant Bit first

5.36.2.15 RESET

```
#define RESET FALSE
```

Reset macro

5.36.2.16 SET

```
#define SET TRUE
```

Set macro

5.36.2.17 TRUE

```
#define TRUE 1
```

Boolean Constants.

Defines standard boolean constants. Boolean true

5.36.3 Typedef Documentation

5.36.3.1 boolean

```
typedef unsigned char boolean
```

Standard Data Types.

Defines standard data types used across the platform. Boolean type, 1 byte

5.36.3.2 ConstVoidPtr

```
typedef const void* ConstVoidPtr
```

Constant generic pointer type

5.36.3.3 float32

```
typedef float float32
```

Floating-Point Types.

Defines floating-point data types. 32-bit floating point

5.36.3.4 float64

```
typedef double float64
```

64-bit floating point

5.36.3.5 sint16

```
typedef signed short sint16
```

Signed 16-bit integer

5.36.3.6 sint16_least

```
typedef signed long sint16_least
```

Signed minimum 16-bit integer

5.36.3.7 sint32

```
typedef signed long sint32
```

Signed 32-bit integer

5.36.3.8 sint32_least

```
typedef signed long sint32_least
```

Signed minimum 32-bit integer

5.36.3.9 sint64

```
typedef signed long long sint64
```

Signed 64-bit integer

5.36.3.10 sint8

```
typedef signed char sint8
```

Signed 8-bit integer

5.36.3.11 sint8_least

```
typedef signed long sint8_least
```

Signed minimum 8-bit integer

5.36.3.12 uint16

```
typedef unsigned short uint16
```

Unsigned 16-bit integer

5.36.3.13 uint16_least

```
typedef unsigned long uint16_least
```

Unsigned minimum 16-bit integer

5.36.3.14 uint32

```
typedef unsigned long uint32
```

Unsigned 32-bit integer

5.36.3.15 uint32_least

```
typedef unsigned long uint32_least
```

Unsigned minimum 32-bit integer

5.36.3.16 uint64

```
typedef unsigned long long uint64
```

Unsigned 64-bit integer

5.36.3.17 uint8

```
typedef unsigned char uint8
```

Unsigned 8-bit integer

5.36.3.18 uint8_least

```
typedef unsigned long uint8_least
```

Least Type Definitions.

Defines types with the minimum size constraints useful for memory-limited systems. Unsigned minimum 8-bit integer

5.36.3.19 VoidPtr

```
typedef void* VoidPtr
```

Pointer Types.

Defines generic pointer types. Generic pointer type

5.36.3.20 vuint16_t

```
typedef volatile unsigned short vuint16_t
```

Volatile 16-bit unsigned integer

5.36.3.21 vuint32_t

```
typedef volatile unsigned long vuint32_t
```

Volatile 32-bit unsigned integer

5.36.3.22 vuint8_t

```
typedef volatile unsigned char vuint8_t
```

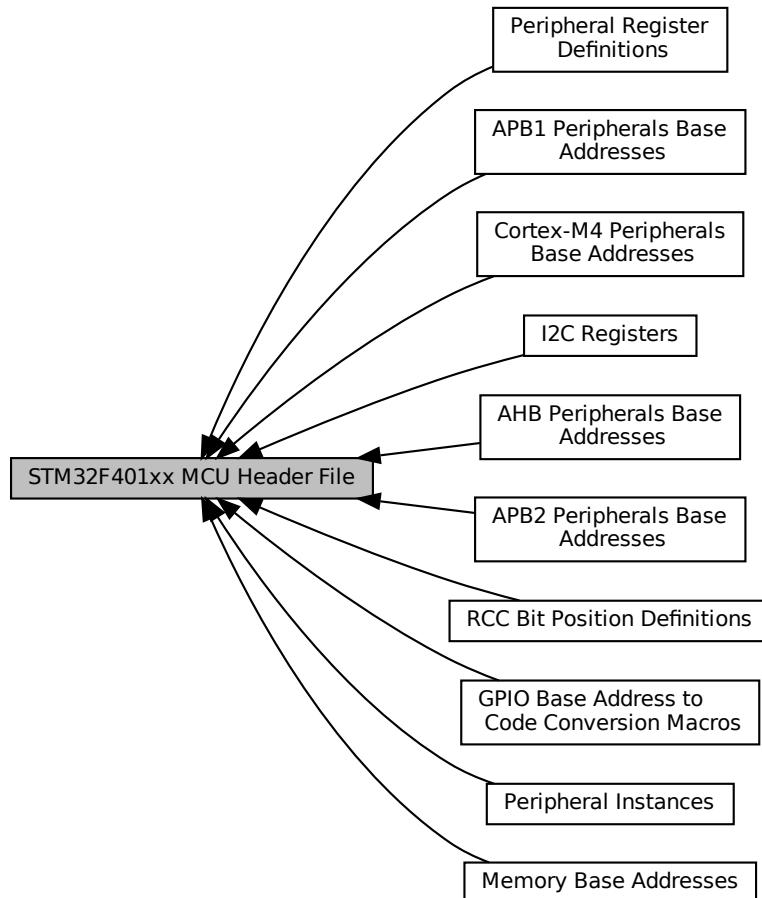
Volatile Types.

Defines volatile data types used for hardware registers. Volatile 8-bit unsigned integer

5.37 STM32F401xx MCU Header File

Header file containing all the necessary information about the STM32F401xx MCU.

Collaboration diagram for STM32F401xx MCU Header File:



Modules

- [Memory Base Addresses](#)
- [Cortex-M4 Peripherals Base Addresses](#)
- [AHB Peripherals Base Addresses](#)
- [APB2 Peripherals Base Addresses](#)
- [APB1 Peripherals Base Addresses](#)
- [Peripheral Register Definitions](#)

Structures defining the register layouts for various peripherals.

- [Peripheral Instances](#)

Peripheral instances for various hardware modules in the STM32 microcontroller.

- [I2C Registers](#)

Bit definitions for the I2C peripheral registers.

- [RCC Bit Position Definitions](#)

Bit position definitions for various registers in the RCC peripheral.

- [GPIO Base Address to Code Conversion Macros](#)

Macros for converting GPIO base addresses to corresponding port codes.

5.37.1 Detailed Description

Header file containing all the necessary information about the STM32F401xx MCU.

5.38 Memory Base Addresses

Collaboration diagram for Memory Base Addresses:



Macros

- `#define FLASH_MEMORY_BASE 0x08000000UL`
Base address of Flash memory.
- `#define SYSTEM_MEMORY_BASE 0x1FFFF000UL`
Base address of System memory.
- `#define SRAM_MEMORY_BASE 0x20000000UL`
Base address of SRAM memory.
- `#define PERIPHERALS_BASE 0x40000000UL`
Base address of Peripheral registers.
- `#define CORTEX_M4_INTERNAL_BASE 0xE0000000UL`
Base address of Cortex-M4 internal peripherals.

5.38.1 Detailed Description

5.39 Cortex-M4 Peripherals Base Addresses

Collaboration diagram for Cortex-M4 Peripherals Base Addresses:



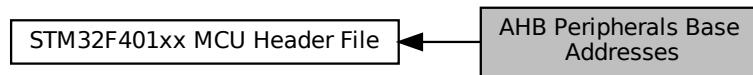
Macros

- `#define NVIC_BASE 0xE000E100UL`
Base address of NVIC (Nested Vectored Interrupt Controller)
- `#define SCB_BASE 0xE000ED00UL`
Base address of System Control Block (SCB)
- `#define STK_BASE 0xE000E010UL`
Base address of SysTick Timer.

5.39.1 Detailed Description

5.40 AHB Peripherals Base Addresses

Collaboration diagram for AHB Peripherals Base Addresses:



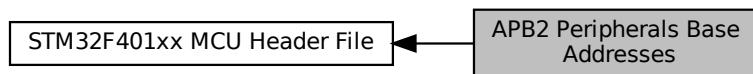
Macros

- `#define RCC_BASE 0x40023800UL`
Base address of RCC (Reset and Clock Control)
- `#define CRC_BASE 0x40023000UL`
Base address of CRC (Cyclic Redundancy Check)

5.40.1 Detailed Description

5.41 APB2 Peripherals Base Addresses

Collaboration diagram for APB2 Peripherals Base Addresses:



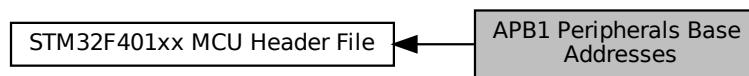
Macros

- `#define GPIOA_BASE 0x40020000UL`
Base address of GPIOA (General Purpose I/O port A)
- `#define GPIOB_BASE 0x40020400UL`
Base address of GPIOB (General Purpose I/O port B)
- `#define GPIOC_BASE 0x40020800UL`
Base address of GPIOC (General Purpose I/O port C)
- `#define GPIOD_BASE 0x40020C00UL`
Base address of GPIOD (General Purpose I/O port D)
- `#define GPIOE_BASE 0x40021000UL`
Base address of GPIOE (General Purpose I/O port E)
- `#define EXTI_BASE 0x40013C00UL`
Base address of EXTI (External Interrupt/Event Controller)
- `#define TIM1_BASE 0x40010000UL`
Base address of TIM1 (Timer 1)
- `#define TIM2_BASE 0x40000000UL`
Base address of TIM2 (Timer 2)
- `#define USART1_BASE 0x40011000UL`
Base address of USART1 (Universal Synchronous/Asynchronous Receiver Transmitter 1)
- `#define USART6_BASE 0x40011400UL`
Base address of USART6 (Universal Synchronous/Asynchronous Receiver Transmitter 6)
- `#define SPI1_BASE 0x40013000UL`
Base address of SPI1 (Serial Peripheral Interface 1)
- `#define SYSCFG_BASE 0x40013800UL`
Base address of SYSCFG (System Configuration Controller)

5.41.1 Detailed Description

5.42 APB1 Peripherals Base Addresses

Collaboration diagram for APB1 Peripherals Base Addresses:



Macros

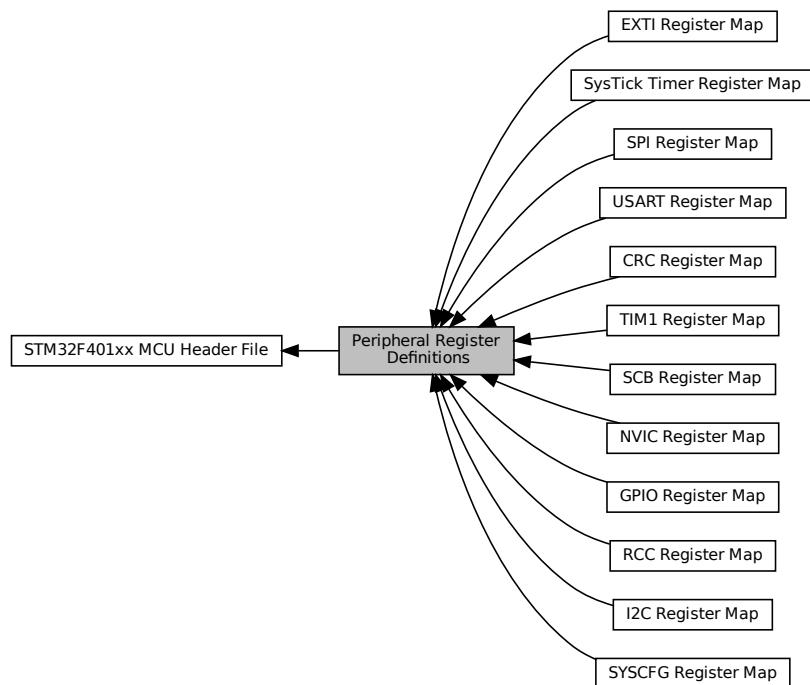
- `#define USART2_BASE 0x40004400UL`
Base address of USART2 (Universal Synchronous/Asynchronous Receiver Transmitter 2)
- `#define SPI2_BASE 0x40003800UL`
Base address of SPI2 (Serial Peripheral Interface 2)
- `#define I2C1_BASE 0x40005400UL`
Base address of I2C1 (Inter-Integrated Circuit 1)
- `#define I2C2_BASE 0x40005800UL`
Base address of I2C2 (Inter-Integrated Circuit 2)

5.42.1 Detailed Description

5.43 Peripheral Register Definitions

Structures defining the register layouts for various peripherals.

Collaboration diagram for Peripheral Register Definitions:



Modules

- [NVIC Register Map](#)
NVIC (Nested Vectored Interrupt Controller) Register Map.
- [SCB Register Map](#)
SCB (System Control Block) Register Map.
- [SysTick Timer Register Map](#)
SysTick Timer Register Map.
- [GPIO Register Map](#)
GPIO (General-Purpose Input/Output) Register Map.
- [RCC Register Map](#)
RCC (Reset and Clock Control) Register Map.
- [EXTI Register Map](#)
EXTI (External Interrupts) Register Map.
- [SYSCFG Register Map](#)
SYSCFG (System Configuration) Register Map.
- [USART Register Map](#)

USART (Universal Synchronous Asynchronous Receiver Transmitter) Register Map.

- [SPI Register Map](#)

SPI (Serial Peripheral Interface) Register Map.

- [I2C Register Map](#)

I2C (Inter-Integrated Circuit) Register Map.

- [CRC Register Map](#)

CRC (Cyclic Redundancy Check) Register Map.

- [TIM1 Register Map](#)

TIM1 (Timer 1) Register Map.

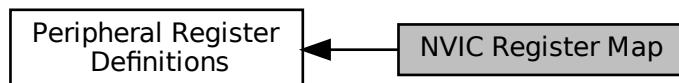
5.43.1 Detailed Description

Structures defining the register layouts for various peripherals.

5.44 NVIC Register Map

NVIC (Nested Vectored Interrupt Controller) Register Map.

Collaboration diagram for NVIC Register Map:



Classes

- struct [NVIC_TypeDef](#)

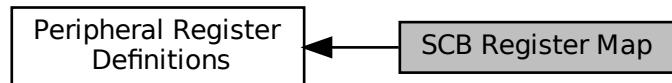
5.44.1 Detailed Description

NVIC (Nested Vectored Interrupt Controller) Register Map.

5.45 SCB Register Map

SCB (System Control Block) Register Map.

Collaboration diagram for SCB Register Map:



Classes

- struct [SCB_TypeDef](#)

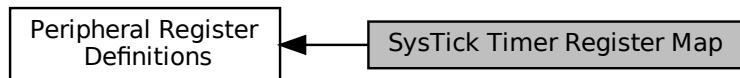
5.45.1 Detailed Description

SCB (System Control Block) Register Map.

5.46 SysTick Timer Register Map

SysTick Timer Register Map.

Collaboration diagram for SysTick Timer Register Map:



Classes

- struct [STK_TypeDef](#)

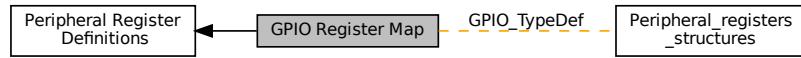
5.46.1 Detailed Description

SysTick Timer Register Map.

5.47 GPIO Register Map

GPIO (General-Purpose Input/Output) Register Map.

Collaboration diagram for GPIO Register Map:



Classes

- struct [GPIO_TypeDef](#)

General Purpose I/O.

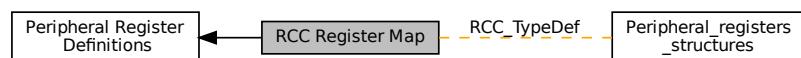
5.47.1 Detailed Description

GPIO (General-Purpose Input/Output) Register Map.

5.48 RCC Register Map

RCC (Reset and Clock Control) Register Map.

Collaboration diagram for RCC Register Map:



Classes

- struct [RCC_TypeDef](#)

Reset and Clock Control.

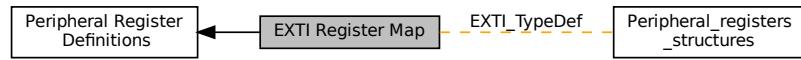
5.48.1 Detailed Description

RCC (Reset and Clock Control) Register Map.

5.49 EXTI Register Map

EXTI (External Interrupts) Register Map.

Collaboration diagram for EXTI Register Map:



Classes

- struct [EXTI_TypeDef](#)
External Interrupt/Event Controller.

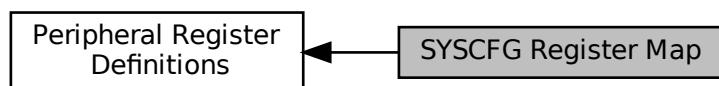
5.49.1 Detailed Description

EXTI (External Interrupts) Register Map.

5.50 SYSCFG Register Map

SYSCFG (System Configuration) Register Map.

Collaboration diagram for SYSCFG Register Map:



Classes

- struct [SYSCFG_RegDef_t](#)

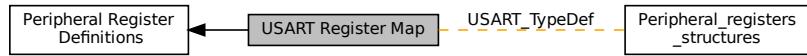
5.50.1 Detailed Description

SYSCFG (System Configuration) Register Map.

5.51 USART Register Map

USART (Universal Synchronous Asynchronous Receiver Transmitter) Register Map.

Collaboration diagram for USART Register Map:



Classes

- struct [USART_TypeDef](#)
Universal Synchronous Asynchronous Receiver Transmitter.

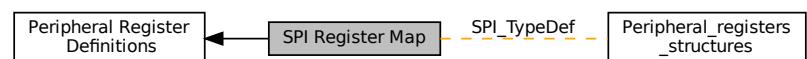
5.51.1 Detailed Description

USART (Universal Synchronous Asynchronous Receiver Transmitter) Register Map.

5.52 SPI Register Map

SPI (Serial Peripheral Interface) Register Map.

Collaboration diagram for SPI Register Map:



Classes

- struct [SPI_TypeDef](#)
Serial Peripheral Interface.

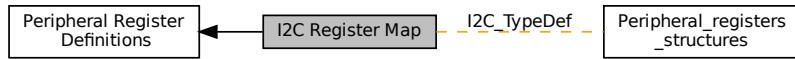
5.52.1 Detailed Description

SPI (Serial Peripheral Interface) Register Map.

5.53 I2C Register Map

I2C (Inter-Integrated Circuit) Register Map.

Collaboration diagram for I2C Register Map:



Classes

- struct [I2C_TypeDef](#)
Inter-integrated Circuit Interface.

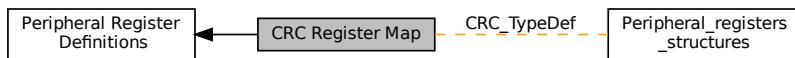
5.53.1 Detailed Description

I2C (Inter-Integrated Circuit) Register Map.

5.54 CRC Register Map

CRC (Cyclic Redundancy Check) Register Map.

Collaboration diagram for CRC Register Map:



Classes

- struct [CRC_TypeDef](#)
CRC calculation unit.

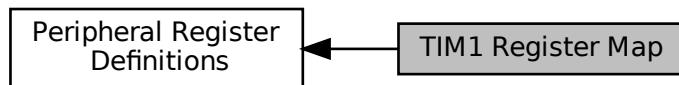
5.54.1 Detailed Description

CRC (Cyclic Redundancy Check) Register Map.

5.55 TIM1 Register Map

TIM1 (Timer 1) Register Map.

Collaboration diagram for TIM1 Register Map:



Classes

- struct [TIM1_TypeDef](#)

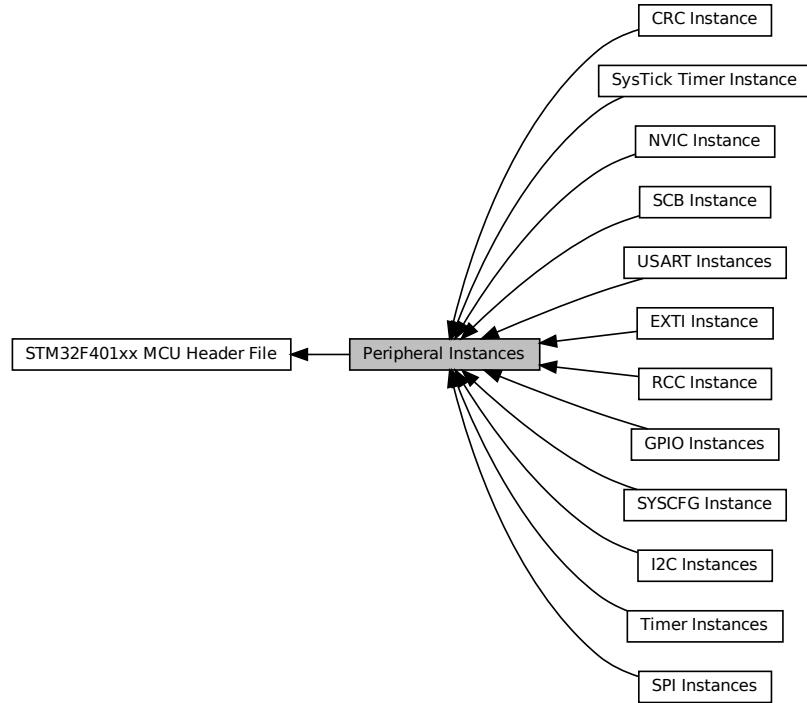
5.55.1 Detailed Description

TIM1 (Timer 1) Register Map.

5.56 Peripheral Instances

Peripheral instances for various hardware modules in the STM32 microcontroller.

Collaboration diagram for Peripheral Instances:



Modules

- **NVIC Instance**
Instance of the Nested Vectored Interrupt Controller (NVIC).
- **SCB Instance**
Instance of the System Control Block (SCB).
- **SysTick Timer Instance**
Instance of the SysTick Timer.
- **GPIO Instances**
Instances of the General-Purpose Input/Output (GPIO) ports.
- **RCC Instance**
Instance of the Reset and Clock Control (RCC).
- **EXTI Instance**
Instance of the External Interrupts (EXTI).
- **SYSCFG Instance**
Instance of the System Configuration (SYSCFG).
- **Timer Instances**
Instances of the Timers (TIM1 and TIM2).
- **USART Instances**
Instances of the Universal Synchronous Asynchronous Receiver Transmitters (USART1, USART2, and USART6).
- **SPI Instances**
Instances of the Serial Peripheral Interfaces (SPI1 and SPI2).
- **I2C Instances**
Instances of the Inter-Integrated Circuits (I2C1 and I2C2).
- **CRC Instance**
Instance of the Cyclic Redundancy Check (CRC).

5.56.1 Detailed Description

Peripheral instances for various hardware modules in the STM32 microcontroller.

5.57 NVIC Instance

Instance of the Nested Vectored Interrupt Controller (NVIC).

Collaboration diagram for NVIC Instance:



Macros

- `#define NVIC ((NVIC_TypeDef*)NVIC_BASE)`

5.57.1 Detailed Description

Instance of the Nested Vectored Interrupt Controller (NVIC).

5.57.2 Macro Definition Documentation

5.57.2.1 NVIC

```
#define NVIC ((NVIC_TypeDef*)NVIC_BASE)
```

Pointer to the NVIC registers.

5.58 SCB Instance

Instance of the System Control Block (SCB).

Collaboration diagram for SCB Instance:



Macros

- `#define SCB ((SCB_TypeDef*)SCB_BASE)`

5.58.1 Detailed Description

Instance of the System Control Block (SCB).

5.58.2 Macro Definition Documentation

5.58.2.1 SCB

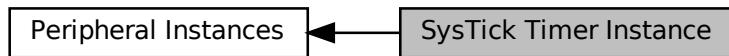
```
#define SCB ((SCB_TypeDef*) SCB_BASE)
```

Pointer to the SCB registers.

5.59 SysTick Timer Instance

Instance of the SysTick Timer.

Collaboration diagram for SysTick Timer Instance:



Macros

- `#define STK ((STK_TypeDef*)STK_BASE)`

5.59.1 Detailed Description

Instance of the SysTick Timer.

5.59.2 Macro Definition Documentation

5.59.2.1 STK

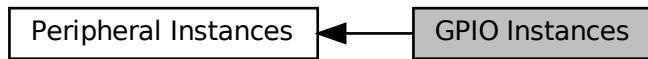
```
#define STK ((STK_TypeDef*)STK_BASE)
```

Pointer to the SysTick Timer registers.

5.60 GPIO Instances

Instances of the General-Purpose Input/Output (GPIO) ports.

Collaboration diagram for GPIO Instances:



Macros

- `#define GPIOA ((GPIO_TypeDef*)GPIOA_BASE)`
- `#define GPIOB ((GPIO_TypeDef*)GPIOB_BASE)`
- `#define GPIOC ((GPIO_TypeDef*)GPIOC_BASE)`
- `#define GPIOD ((GPIO_TypeDef*)GPIOD_BASE)`
- `#define GPIOE ((GPIO_TypeDef*)GPIOE_BASE)`

5.60.1 Detailed Description

Instances of the General-Purpose Input/Output (GPIO) ports.

5.60.2 Macro Definition Documentation

5.60.2.1 GPIOA

```
#define GPIOA ((GPIO_TypeDef*)GPIOA_BASE)
```

Pointer to GPIOA registers.

5.60.2.2 GPIOB

```
#define GPIOB ((GPIO_TypeDef*)GPIOB_BASE)
```

Pointer to GPIOB registers.

5.60.2.3 GPIOC

```
#define GPIOC ((GPIO_TypeDef*)GPIOC_BASE)
```

Pointer to GPIOC registers.

5.60.2.4 GPIOD

```
#define GPIOD ((GPIO_TypeDef*)GPIOD_BASE)
```

Pointer to GPIOD registers.

5.60.2.5 GPIOE

```
#define GPIOE ((GPIO_TypeDef*)GPIOE_BASE)
```

Pointer to GPIOE registers.

5.61 RCC Instance

Instance of the Reset and Clock Control (RCC).

Collaboration diagram for RCC Instance:



Macros

- `#define RCC ((RCC_TypeDef*)RCC_BASE)`

5.61.1 Detailed Description

Instance of the Reset and Clock Control (RCC).

5.61.2 Macro Definition Documentation

5.61.2.1 RCC

```
#define RCC ((RCC_TypeDef*) RCC_BASE)
```

Pointer to the RCC registers.

5.62 EXTI Instance

Instance of the External Interrupts (EXTI).

Collaboration diagram for EXTI Instance:



Macros

- `#define EXTI ((EXTI_TypeDef*)EXTI_BASE)`

5.62.1 Detailed Description

Instance of the External Interrupts (EXTI).

5.62.2 Macro Definition Documentation

5.62.2.1 EXTI

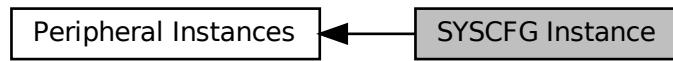
```
#define EXTI ((EXTI_TypeDef*)EXTI_BASE)
```

Pointer to the EXTI registers.

5.63 SYSCFG Instance

Instance of the System Configuration (SYSCFG).

Collaboration diagram for SYSCFG Instance:



Macros

- `#define SYSCFG ((SYSCFG_RegDef_t*)SYSCFG_BASE)`

5.63.1 Detailed Description

Instance of the System Configuration (SYSCFG).

5.63.2 Macro Definition Documentation

5.63.2.1 SYSCFG

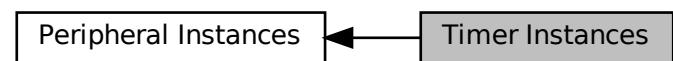
```
#define SYSCFG ((SYSCFG_RegDef_t*)SYSCFG_BASE)
```

Pointer to the SYSCFG registers.

5.64 Timer Instances

Instances of the Timers (TIM1 and TIM2).

Collaboration diagram for Timer Instances:



Macros

- `#define TIM1 ((TIM1_TypeDef*)TIM1_BASE)`
- `#define TIM2 ((TIM1_TypeDef*)TIM2_BASE)`

5.64.1 Detailed Description

Instances of the Timers (TIM1 and TIM2).

5.64.2 Macro Definition Documentation

5.64.2.1 TIM1

```
#define TIM1 ((TIM1_TypeDef*)TIM1_BASE)
```

Pointer to TIM1 registers.

5.64.2.2 TIM2

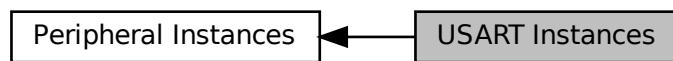
```
#define TIM2 ((TIM1_TypeDef*)TIM2_BASE)
```

Pointer to TIM2 registers.

5.65 USART Instances

Instances of the Universal Synchronous Asynchronous Receiver Transmitters (USART1, USART2, and USART6).

Collaboration diagram for USART Instances:



Macros

- `#define USART1 ((USART_TypeDef*)USART1_BASE)`
- `#define USART2 ((USART_TypeDef*)USART2_BASE)`
- `#define USART6 ((USART_TypeDef*)USART6_BASE)`

5.65.1 Detailed Description

Instances of the Universal Synchronous Asynchronous Receiver Transmitters (USART1, USART2, and USART6).

5.65.2 Macro Definition Documentation

5.65.2.1 USART1

```
#define USART1 ((USART_TypeDef*)USART1_BASE)
```

Pointer to USART1 registers.

5.65.2.2 USART2

```
#define USART2 ((USART_TypeDef*)USART2_BASE)
```

Pointer to USART2 registers.

5.65.2.3 USART6

```
#define USART6 ((USART_TypeDef*)USART6_BASE)
```

Pointer to USART6 registers.

5.66 SPI Instances

Instances of the Serial Peripheral Interfaces (SPI1 and SPI2).

Collaboration diagram for SPI Instances:



Macros

- #define SPI1 ((SPI_TypeDef*)SPI1_BASE)
- #define SPI2 ((SPI_TypeDef*)SPI2_BASE)

5.66.1 Detailed Description

Instances of the Serial Peripheral Interfaces (SPI1 and SPI2).

5.66.2 Macro Definition Documentation

5.66.2.1 SPI1

```
#define SPI1 ((SPI_TypeDef*) SPI1_BASE)
```

Pointer to SPI1 registers.

5.66.2.2 SPI2

```
#define SPI2 ((SPI_TypeDef*) SPI2_BASE)
```

Pointer to SPI2 registers.

5.67 I2C Instances

Instances of the Inter-Integrated Circuits (I2C1 and I2C2).

Collaboration diagram for I2C Instances:



Macros

- #define I2C1 ((I2C_TypeDef*)I2C1_BASE)
- #define I2C2 ((I2C_TypeDef*)I2C2_BASE)

5.67.1 Detailed Description

Instances of the Inter-Integrated Circuits (I2C1 and I2C2).

5.67.2 Macro Definition Documentation

5.67.2.1 I2C1

```
#define I2C1 ((I2C_TypeDef*)I2C1_BASE)
```

Pointer to I2C1 registers.

5.67.2.2 I2C2

```
#define I2C2 ((I2C_TypeDef*)I2C2_BASE)
```

Pointer to I2C2 registers.

5.68 CRC Instance

Instance of the Cyclic Redundancy Check (CRC).

Collaboration diagram for CRC Instance:



Macros

- #define CRC ((CRC_TypeDef*)CRC_BASE)

5.68.1 Detailed Description

Instance of the Cyclic Redundancy Check (CRC).

5.68.2 Macro Definition Documentation

5.68.2.1 CRC

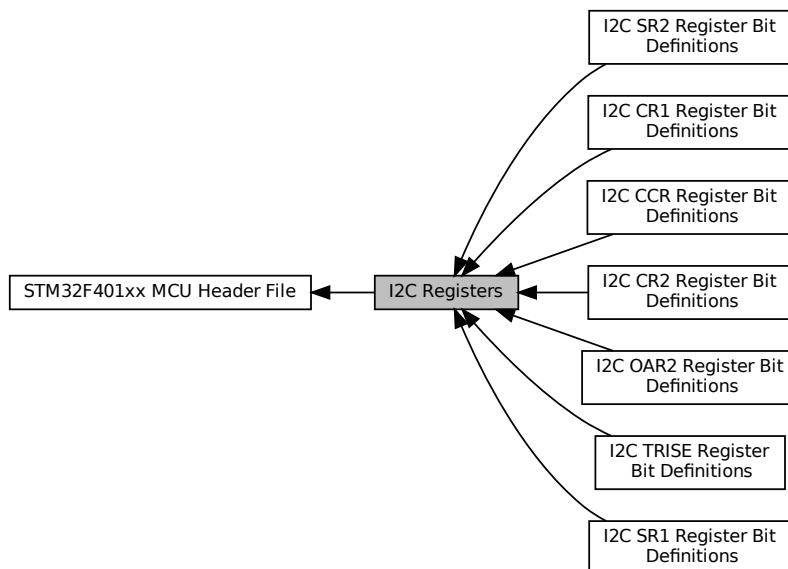
```
#define CRC ((CRC_TypeDef*) CRC_BASE)
```

Pointer to the CRC registers.

5.69 I2C Registers

Bit definitions for the I2C peripheral registers.

Collaboration diagram for I2C Registers:



Modules

- [I2C CR1 Register Bit Definitions](#)
Bit definitions for the I2C Control Register 1 (CR1).
- [I2C CR2 Register Bit Definitions](#)
Bit definitions for the I2C Control Register 2 (CR2).
- [I2C OAR2 Register Bit Definitions](#)
- [I2C SR1 Register Bit Definitions](#)
- [I2C SR2 Register Bit Definitions](#)
- [I2C CCR Register Bit Definitions](#)
- [I2C TRISE Register Bit Definitions](#)

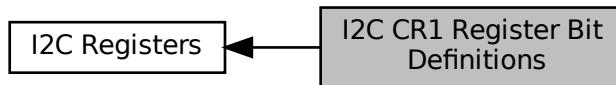
5.69.1 Detailed Description

Bit definitions for the I2C peripheral registers.

5.70 I2C CR1 Register Bit Definitions

Bit definitions for the I2C Control Register 1 (CR1).

Collaboration diagram for I2C CR1 Register Bit Definitions:



Macros

- #define `I2C_CR1_PE_Pos` (0U)

Position and mask for the Peripheral Enable bit in the I2C_CR1 register.
- #define `I2C_CR1_PE_Msk` (0x1UL << `I2C_CR1_PE_Pos`)

`I2C_CR1_PE` I2C_CR1_PE_Msk
- #define `I2C_CR1_SMBUS_Pos` (1U)

Position and mask for the SMBus Mode bit in the I2C_CR1 register.
- #define `I2C_CR1_SMBUS_Msk` (0x1UL << `I2C_CR1_SMBUS_Pos`)

`I2C_CR1_SMBUS` I2C_CR1_SMBUS_Msk
- #define `I2C_CR1_SMBTYPE_Pos` (3U)

Position and mask for the SMBus Type bit in the I2C_CR1 register.
- #define `I2C_CR1_SMBTYPE_Msk` (0x1UL << `I2C_CR1_SMBTYPE_Pos`)

`I2C_CR1_SMBTYPE` I2C_CR1_SMBTYPE_Msk
- #define `I2C_CR1_ENARP_Pos` (4U)

Position and mask for the ARP Enable bit in the I2C_CR1 register.
- #define `I2C_CR1_ENARP_Msk` (0x1UL << `I2C_CR1_ENARP_Pos`)

`I2C_CR1_ENARP` I2C_CR1_ENARP_Msk
- #define `I2C_CR1_ENPEC_Pos` (5U)

Position and mask for the PEC Enable bit in the I2C_CR1 register.
- #define `I2C_CR1_ENPEC_Msk` (0x1UL << `I2C_CR1_ENPEC_Pos`)

`I2C_CR1_ENPEC` I2C_CR1_ENPEC_Msk
- #define `I2C_CR1_ENGC_Pos` (6U)

Position and mask for the General Call Enable bit in the I2C_CR1 register.
- #define `I2C_CR1_ENGC_Msk` (0x1UL << `I2C_CR1_ENGC_Pos`)

`I2C_CR1_ENGC` I2C_CR1_ENGC_Msk
- #define `I2C_CR1_NOSTRETCH_Pos` (7U)

Position and mask for the Clock Stretching Disable bit in the I2C_CR1 register.
- #define `I2C_CR1_NOSTRETCH_Msk` (0x1UL << `I2C_CR1_NOSTRETCH_Pos`)

`I2C_CR1_NOSTRETCH` I2C_CR1_NOSTRETCH_Msk
- #define `I2C_CR1_START_Pos` (8U)

Position and mask for the Start Generation bit in the I2C_CR1 register.
- #define `I2C_CR1_START_Msk` (0x1UL << `I2C_CR1_START_Pos`)

`I2C_CR1_START` I2C_CR1_START_Msk
- #define `I2C_CR1_STOP_Pos` (9U)

Position and mask for the Stop Generation bit in the I2C_CR1 register.

- #define I2C_CR1_STOP_Msk (0x1UL << I2C_CR1_STOP_Pos)
- #define I2C_CR1_STOP I2C_CR1_STOP_Msk
- #define I2C_CR1_ACK_Pos (10U)

Position and mask for the Acknowledge Enable bit in the I2C_CR1 register.

- #define I2C_CR1_ACK_Msk (0x1UL << I2C_CR1_ACK_Pos)
- #define I2C_CR1_ACK I2C_CR1_ACK_Msk
- #define I2C_CR1_POS_Pos (11U)

Position and mask for the Acknowledge/PEC Position bit in the I2C_CR1 register.

- #define I2C_CR1_POS_Msk (0x1UL << I2C_CR1_POS_Pos)
- #define I2C_CR1_POS I2C_CR1_POS_Msk
- #define I2C_CR1_PEC_Pos (12U)

Position and mask for the Packet Error Checking bit in the I2C_CR1 register.

- #define I2C_CR1_PEC_Msk (0x1UL << I2C_CR1_PEC_Pos)
- #define I2C_CR1_PEC I2C_CR1_PEC_Msk
- #define I2C_CR1_ALERT_Pos (13U)

Position and mask for the SMBus Alert bit in the I2C_CR1 register.

- #define I2C_CR1_ALERT_Msk (0x1UL << I2C_CR1_ALERT_Pos)
- #define I2C_CR1_ALERT I2C_CR1_ALERT_Msk
- #define I2C_CR1_SWRST_Pos (15U)

Position and mask for the Software Reset bit in the I2C_CR1 register.

- #define I2C_CR1_SWRST_Msk (0x1UL << I2C_CR1_SWRST_Pos)
- #define I2C_CR1_SWRST I2C_CR1_SWRST_Msk

5.70.1 Detailed Description

Bit definitions for the I2C Control Register 1 (CR1).

5.70.2 Macro Definition Documentation

5.70.2.1 I2C_CR1_ACK

```
#define I2C_CR1_ACK I2C_CR1_ACK_Msk
```

Acknowledge Enable

5.70.2.2 I2C_CR1_ACK_Msk

```
#define I2C_CR1_ACK_Msk (0x1UL << I2C_CR1_ACK_Pos)
```

Bit mask for Acknowledge Enable

5.70.2.3 I2C_CR1_ACK_Pos

```
#define I2C_CR1_ACK_Pos (10U)
```

Position and mask for the Acknowledge Enable bit in the I2C_CR1 register.

Position of Acknowledge Enable bit

5.70.2.4 I2C_CR1_ALERT

```
#define I2C_CR1_ALERT I2C_CR1_ALERT_Msk
```

SMBus Alert

5.70.2.5 I2C_CR1_ALERT_Msk

```
#define I2C_CR1_ALERT_Msk (0x1UL << I2C_CR1_ALERT_Pos)
```

Bit mask for SMBus Alert

5.70.2.6 I2C_CR1_ALERT_Pos

```
#define I2C_CR1_ALERT_Pos (13U)
```

Position and mask for the SMBus Alert bit in the I2C_CR1 register.

Position of SMBus Alert bit

5.70.2.7 I2C_CR1_ENARP

```
#define I2C_CR1_ENARP I2C_CR1_ENARP_Msk
```

ARP Enable

5.70.2.8 I2C_CR1_ENARP_Msk

```
#define I2C_CR1_ENARP_Msk (0x1UL << I2C_CR1_ENARP_Pos)
```

Bit mask for ARP Enable

5.70.2.9 I2C_CR1_ENARP_Pos

```
#define I2C_CR1_ENARP_Pos (4U)
```

Position and mask for the ARP Enable bit in the I2C_CR1 register.

Position of ARP Enable bit

5.70.2.10 I2C_CR1_ENGC

```
#define I2C_CR1_ENGC I2C_CR1_ENGC_Msk
```

General Call Enable

5.70.2.11 I2C_CR1_ENGC_Msk

```
#define I2C_CR1_ENGC_Msk (0x1UL << I2C_CR1_ENGC_Pos)
```

Bit mask for General Call Enable

5.70.2.12 I2C_CR1_ENGC_Pos

```
#define I2C_CR1_ENGC_Pos (6U)
```

Position and mask for the General Call Enable bit in the I2C_CR1 register.

Position of General Call Enable bit

5.70.2.13 I2C_CR1_ENPEC

```
#define I2C_CR1_ENPEC I2C_CR1_ENPEC_Msk
```

PEC Enable

5.70.2.14 I2C_CR1_ENPEC_Msk

```
#define I2C_CR1_ENPEC_Msk (0x1UL << I2C_CR1_ENPEC_Pos)
```

Bit mask for PEC Enable

5.70.2.15 I2C_CR1_ENPEC_Pos

```
#define I2C_CR1_ENPEC_Pos (5U)
```

Position and mask for the PEC Enable bit in the I2C_CR1 register.

Position of PEC Enable bit

5.70.2.16 I2C_CR1_NOSTRETCH

```
#define I2C_CR1_NOSTRETCH I2C_CR1_NOSTRETCH_Msk
```

Clock Stretching Disable (Slave mode)

5.70.2.17 I2C_CR1_NOSTRETCH_Msk

```
#define I2C_CR1_NOSTRETCH_Msk (0x1UL << I2C_CR1_NOSTRETCH_Pos)
```

Bit mask for Clock Stretching Disable

5.70.2.18 I2C_CR1_NOSTRETCH_Pos

```
#define I2C_CR1_NOSTRETCH_Pos (7U)
```

Position and mask for the Clock Stretching Disable bit in the I2C_CR1 register.

Position of Clock Stretching Disable bit

5.70.2.19 I2C_CR1_PE

```
#define I2C_CR1_PE I2C_CR1_PE_Msk
```

Peripheral Enable

5.70.2.20 I2C_CR1_PE_Msk

```
#define I2C_CR1_PE_Msk (0x1UL << I2C_CR1_PE_Pos)
```

Bit mask for Peripheral Enable

5.70.2.21 I2C_CR1_PE_Pos

```
#define I2C_CR1_PE_Pos (0U)
```

Position and mask for the Peripheral Enable bit in the I2C_CR1 register.

Position of Peripheral Enable bit

5.70.2.22 I2C_CR1_PEC

```
#define I2C_CR1_PEC I2C_CR1_PEC_Msk
```

Packet Error Checking

5.70.2.23 I2C_CR1_PEC_Msk

```
#define I2C_CR1_PEC_Msk (0x1UL << I2C_CR1_PEC_Pos)
```

Bit mask for Packet Error Checking

5.70.2.24 I2C_CR1_PEC_Pos

```
#define I2C_CR1_PEC_Pos (12U)
```

Position and mask for the Packet Error Checking bit in the I2C_CR1 register.

Position of Packet Error Checking bit

5.70.2.25 I2C_CR1_POS

```
#define I2C_CR1_POS I2C_CR1_POS_Msk
```

Acknowledge/PEC Position (for data reception)

5.70.2.26 I2C_CR1_POS_Msk

```
#define I2C_CR1_POS_Msk (0x1UL << I2C_CR1_POS_Pos)
```

Bit mask for Acknowledge/PEC Position

5.70.2.27 I2C_CR1_POS_Pos

```
#define I2C_CR1_POS_Pos (11U)
```

Position and mask for the Acknowledge/PEC Position bit in the I2C_CR1 register.

Position of Acknowledge/PEC Position bit

5.70.2.28 I2C_CR1_SMBTYPE

```
#define I2C_CR1_SMBTYPE I2C_CR1_SMBTYPE_Msk
```

SMBus Type

5.70.2.29 I2C_CR1_SMBTYPE_Msk

```
#define I2C_CR1_SMBTYPE_Msk (0x1UL << I2C_CR1_SMBTYPE_Pos)
```

Bit mask for SMBus Type

5.70.2.30 I2C_CR1_SMBTYPE_Pos

```
#define I2C_CR1_SMBTYPE_Pos (3U)
```

Position and mask for the SMBus Type bit in the I2C_CR1 register.

Position of SMBus Type bit

5.70.2.31 I2C_CR1_SMBUS

```
#define I2C_CR1_SMBUS I2C_CR1_SMBUS_Msk
```

SMBus Mode

5.70.2.32 I2C_CR1_SMBUS_Msk

```
#define I2C_CR1_SMBUS_Msk (0x1UL << I2C_CR1_SMBUS_Pos)
```

Bit mask for SMBus Mode

5.70.2.33 I2C_CR1_SMBUS_Pos

```
#define I2C_CR1_SMBUS_Pos (1U)
```

Position and mask for the SMBus Mode bit in the I2C_CR1 register.

Position of SMBus Mode bit

5.70.2.34 I2C_CR1_START

```
#define I2C_CR1_START I2C_CR1_START_Msk
```

Start Generation

5.70.2.35 I2C_CR1_START_Msk

```
#define I2C_CR1_START_Msk (0x1UL << I2C_CR1_START_Pos)
```

Bit mask for Start Generation

5.70.2.36 I2C_CR1_START_Pos

```
#define I2C_CR1_START_Pos (8U)
```

Position and mask for the Start Generation bit in the I2C_CR1 register.

Position of Start Generation bit

5.70.2.37 I2C_CR1_STOP

```
#define I2C_CR1_STOP I2C_CR1_STOP_Msk
```

Stop Generation

5.70.2.38 I2C_CR1_STOP_Msk

```
#define I2C_CR1_STOP_Msk (0x1UL << I2C_CR1_STOP_Pos)
```

Bit mask for Stop Generation

5.70.2.39 I2C_CR1_STOP_Pos

```
#define I2C_CR1_STOP_Pos (9U)
```

Position and mask for the Stop Generation bit in the I2C_CR1 register.

Position of Stop Generation bit

5.70.2.40 I2C_CR1_SWRST

```
#define I2C_CR1_SWRST I2C_CR1_SWRST_Msk
```

Software Reset

5.70.2.41 I2C_CR1_SWRST_Msk

```
#define I2C_CR1_SWRST_Msk (0x1UL << I2C_CR1_SWRST_Pos)
```

Bit mask for Software Reset

5.70.2.42 I2C_CR1_SWRST_Pos

```
#define I2C_CR1_SWRST_Pos (15U)
```

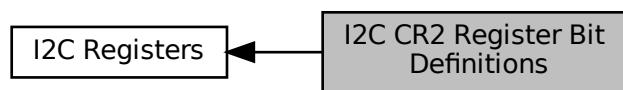
Position and mask for the Software Reset bit in the I2C_CR1 register.

Position of Software Reset bit

5.71 I2C CR2 Register Bit Definitions

Bit definitions for the I2C Control Register 2 (CR2).

Collaboration diagram for I2C CR2 Register Bit Definitions:



Macros

- `#define I2C_CR2_FREQ_Pos (0U)`
Position and mask for the Peripheral Clock Frequency bits in the I2C_CR2 register.
- `#define I2C_CR2_FREQ_Msk (0x3FUL << I2C_CR2_FREQ_Pos)`
- `#define I2C_CR2_FREQ I2C_CR2_FREQ_Msk`
- `#define I2C_CR2_ITERREN_Pos (8U)`
Position and mask for the Error Interrupt Enable bit in the I2C_CR2 register.
- `#define I2C_CR2_ITERREN_Msk (0x1UL << I2C_CR2_ITERREN_Pos)`
- `#define I2C_CR2_ITERREN I2C_CR2_ITERREN_Msk`
- `#define I2C_CR2_IYTEVTEN_Pos (9U)`
Position and mask for the Event Interrupt Enable bit in the I2C_CR2 register.
- `#define I2C_CR2_IYTEVTEN_Msk (0x1UL << I2C_CR2_IYTEVTEN_Pos)`
- `#define I2C_CR2_IYTEVTEN I2C_CR2_IYTEVTEN_Msk`
- `#define I2C_CR2_ITBUFEN_Pos (10U)`
Position and mask for the Buffer Interrupt Enable bit in the I2C_CR2 register.
- `#define I2C_CR2_ITBUFEN_Msk (0x1UL << I2C_CR2_ITBUFEN_Pos)`
- `#define I2C_CR2_ITBUFEN I2C_CR2_ITBUFEN_Msk`
- `#define I2C_CR2_DMAEN_Pos (11U)`
Position and mask for the DMA Requests Enable bit in the I2C_CR2 register.
- `#define I2C_CR2_DMAEN_Msk (0x1UL << I2C_CR2_DMAEN_Pos)`
- `#define I2C_CR2_DMAEN I2C_CR2_DMAEN_Msk`
- `#define I2C_CR2_LAST_Pos (12U)`
Position and mask for the Last DMA Transfer bit in the I2C_CR2 register.
- `#define I2C_CR2_LAST_Msk (0x1UL << I2C_CR2_LAST_Pos)`
- `#define I2C_CR2_LAST I2C_CR2_LAST_Msk`

5.71.1 Detailed Description

Bit definitions for the I2C Control Register 2 (CR2).

5.71.2 Macro Definition Documentation

5.71.2.1 I2C_CR2_DMAEN

```
#define I2C_CR2_DMAEN I2C_CR2_DMAEN_Msk
```

DMA Requests Enable

5.71.2.2 I2C_CR2_DMAEN_Msk

```
#define I2C_CR2_DMAEN_Msk (0x1UL << I2C_CR2_DMAEN_Pos)
```

Bit mask for DMA Requests Enable

5.71.2.3 I2C_CR2_DMAEN_Pos

```
#define I2C_CR2_DMAEN_Pos (11U)
```

Position and mask for the DMA Requests Enable bit in the I2C_CR2 register.

Position of DMA Requests Enable bit

5.71.2.4 I2C_CR2_FREQ

```
#define I2C_CR2_FREQ I2C_CR2_FREQ_Msk
```

Peripheral Clock Frequency

5.71.2.5 I2C_CR2_FREQ_Msk

```
#define I2C_CR2_FREQ_Msk (0x3FUL << I2C_CR2_FREQ_Pos)
```

Bit mask for Peripheral Clock Frequency

5.71.2.6 I2C_CR2_FREQ_Pos

```
#define I2C_CR2_FREQ_Pos (0U)
```

Position and mask for the Peripheral Clock Frequency bits in the I2C_CR2 register.

Position of Peripheral Clock Frequency bits

5.71.2.7 I2C_CR2_ITBUFEN

```
#define I2C_CR2_ITBUFEN I2C_CR2_ITBUFEN_Msk
```

Buffer Interrupt Enable

5.71.2.8 I2C_CR2_ITBUFEN_Msk

```
#define I2C_CR2_ITBUFEN_Msk (0x1UL << I2C_CR2_ITBUFEN_Pos)
```

Bit mask for Buffer Interrupt Enable

5.71.2.9 I2C_CR2_ITBUFEN_Pos

```
#define I2C_CR2_ITBUFEN_Pos (10U)
```

Position and mask for the Buffer Interrupt Enable bit in the I2C_CR2 register.

Position of Buffer Interrupt Enable bit

5.71.2.10 I2C_CR2_ITERREN

```
#define I2C_CR2_ITERREN I2C_CR2_ITERREN_Msk
```

Error Interrupt Enable

5.71.2.11 I2C_CR2_ITERREN_Msk

```
#define I2C_CR2_ITERREN_Msk (0x1UL << I2C_CR2_ITERREN_Pos)
```

Bit mask for Error Interrupt Enable

5.71.2.12 I2C_CR2_ITERREN_Pos

```
#define I2C_CR2_ITERREN_Pos (8U)
```

Position and mask for the Error Interrupt Enable bit in the I2C_CR2 register.

Position of Error Interrupt Enable bit

5.71.2.13 I2C_CR2_Itevtten

```
#define I2C_CR2_Itevtten I2C_CR2_Itevtten_Msk
```

Event Interrupt Enable

5.71.2.14 I2C_CR2_Itevtten_Msk

```
#define I2C_CR2_Itevtten_Msk (0x1UL << I2C_CR2_Itevtten_Pos)
```

Bit mask for Event Interrupt Enable

5.71.2.15 I2C_CR2_Itevtten_Pos

```
#define I2C_CR2_Itevtten_Pos (9U)
```

Position and mask for the Event Interrupt Enable bit in the I2C_CR2 register.

Position of Event Interrupt Enable bit

5.71.2.16 I2C_CR2_LAST

```
#define I2C_CR2_LAST I2C_CR2_LAST_Msk
```

Last DMA Transfer

5.71.2.17 I2C_CR2_LAST_Msk

```
#define I2C_CR2_LAST_Msk (0x1UL << I2C_CR2_LAST_Pos)
```

Bit mask for Last DMA Transfer

5.71.2.18 I2C_CR2_LAST_Pos

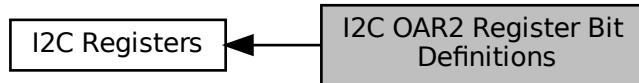
```
#define I2C_CR2_LAST_Pos (12U)
```

Position and mask for the Last DMA Transfer bit in the I2C_CR2 register.

Position of Last DMA Transfer bit

5.72 I2C OAR2 Register Bit Definitions

Collaboration diagram for I2C OAR2 Register Bit Definitions:



Macros

- #define I2C_OAR2_ENDUAL_Pos (0U)
Bit definitions for the I2C_OAR2 register.
- #define I2C_OAR2_ENDUAL_Msk (0x1UL << I2C_OAR2_ENDUAL_Pos)
- #define I2C_OAR2_ENDUAL_I2C_OAR2_ENDUAL_Msk
- #define I2C_OAR2_ADD2_Pos (1U)
- #define I2C_OAR2_ADD2_Msk (0x1UL << I2C_OAR2_ADD2_Pos)
- #define I2C_OAR2_ADD2_I2C_OAR2_ADD2_Msk

5.72.1 Detailed Description

5.72.2 Macro Definition Documentation

5.72.2.1 I2C_OAR2_ADD2

```
#define I2C_OAR2_ADD2 I2C_OAR2_ADD2_Msk
```

Second Address bit

5.72.2.2 I2C_OAR2_ADD2_Msk

```
#define I2C_OAR2_ADD2_Msk (0x1UL << I2C_OAR2_ADD2_Pos)
```

Bit mask for Second Address bit

5.72.2.3 I2C_OAR2_ADD2_Pos

```
#define I2C_OAR2_ADD2_Pos (1U)
```

Position of Second Address bit

5.72.2.4 I2C_OAR2_ENDUAL

```
#define I2C_OAR2_ENDUAL I2C_OAR2_ENDUAL_Msk
```

Dual Addressing Mode Enable

5.72.2.5 I2C_OAR2_ENDUAL_Msk

```
#define I2C_OAR2_ENDUAL_Msk (0x1UL << I2C_OAR2_ENDUAL_Pos)
```

Bit mask for Dual Addressing Mode Enable

5.72.2.6 I2C_OAR2_ENDUAL_Pos

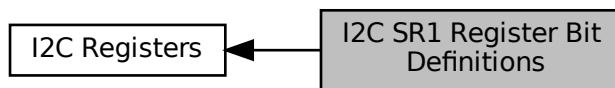
```
#define I2C_OAR2_ENDUAL_Pos (0U)
```

Bit definitions for the I2C_OAR2 register.

Position of Dual Addressing Mode Enable bit

5.73 I2C SR1 Register Bit Definitions

Collaboration diagram for I2C SR1 Register Bit Definitions:



Macros

- `#define I2C_SR1_SB_Pos (0U)`
Bit definitions for the I2C_SR1 register.
- `#define I2C_SR1_SB_Msk (0x1UL << I2C_SR1_SB_Pos)`
- `#define I2C_SR1_SB I2C_SR1_SB_Msk`
- `#define I2C_SR1_ADDR_Pos (1U)`
- `#define I2C_SR1_ADDR_Msk (0x1UL << I2C_SR1_ADDR_Pos)`
- `#define I2C_SR1_ADDR I2C_SR1_ADDR_Msk`
- `#define I2C_SR1_BTF_Pos (2U)`
- `#define I2C_SR1_BTF_Msk (0x1UL << I2C_SR1_BTF_Pos)`
- `#define I2C_SR1_BTF I2C_SR1_BTF_Msk`
- `#define I2C_SR1_ADD10_Pos (3U)`
- `#define I2C_SR1_ADD10_Msk (0x1UL << I2C_SR1_ADD10_Pos)`
- `#define I2C_SR1_ADD10 I2C_SR1_ADD10_Msk`
- `#define I2C_SR1_STOPF_Pos (4U)`
- `#define I2C_SR1_STOPF_Msk (0x1UL << I2C_SR1_STOPF_Pos)`
- `#define I2C_SR1_STOPF I2C_SR1_STOPF_Msk`
- `#define I2C_SR1_RXNE_Pos (6U)`
- `#define I2C_SR1_RXNE_Msk (0x1UL << I2C_SR1_RXNE_Pos)`
- `#define I2C_SR1_RXNE I2C_SR1_RXNE_Msk`
- `#define I2C_SR1_TXE_Pos (7U)`
- `#define I2C_SR1_TXE_Msk (0x1UL << I2C_SR1_TXE_Pos)`
- `#define I2C_SR1_TXE I2C_SR1_TXE_Msk`
- `#define I2C_SR1_BERR_Pos (8U)`
- `#define I2C_SR1_BERR_Msk (0x1UL << I2C_SR1_BERR_Pos)`
- `#define I2C_SR1_BERR I2C_SR1_BERR_Msk`
- `#define I2C_SR1_ARLO_Pos (9U)`
- `#define I2C_SR1_ARLO_Msk (0x1UL << I2C_SR1_ARLO_Pos)`
- `#define I2C_SR1_ARLO I2C_SR1_ARLO_Msk`
- `#define I2C_SR1_AF_Pos (10U)`
- `#define I2C_SR1_AF_Msk (0x1UL << I2C_SR1_AF_Pos)`
- `#define I2C_SR1_AF I2C_SR1_AF_Msk`
- `#define I2C_SR1_OVR_Pos (11U)`
- `#define I2C_SR1_OVR_Msk (0x1UL << I2C_SR1_OVR_Pos)`
- `#define I2C_SR1_OVR I2C_SR1_OVR_Msk`
- `#define I2C_SR1_PECERR_Pos (12U)`
- `#define I2C_SR1_PECERR_Msk (0x1UL << I2C_SR1_PECERR_Pos)`
- `#define I2C_SR1_PECERR I2C_SR1_PECERR_Msk`
- `#define I2C_SR1_TIMEOUT_Pos (14U)`
- `#define I2C_SR1_TIMEOUT_Msk (0x1UL << I2C_SR1_TIMEOUT_Pos)`
- `#define I2C_SR1_TIMEOUT I2C_SR1_TIMEOUT_Msk`
- `#define I2C_SR1_SMBALERT_Pos (15U)`
- `#define I2C_SR1_SMBALERT_Msk (0x1UL << I2C_SR1_SMBALERT_Pos)`
- `#define I2C_SR1_SMBALERT I2C_SR1_SMBALERT_Msk`

5.73.1 Detailed Description

5.73.2 Macro Definition Documentation

5.73.2.1 I2C_SR1_ADD10

```
#define I2C_SR1_ADD10 I2C_SR1_ADD10_Msk
```

10-bit Address Header Sent (Master mode)

5.73.2.2 I2C_SR1_ADD10_Msk

```
#define I2C_SR1_ADD10_Msk (0x1UL << I2C_SR1_ADD10_Pos)
```

Bit mask for 10-bit Address Header Sent

5.73.2.3 I2C_SR1_ADD10_Pos

```
#define I2C_SR1_ADD10_Pos (3U)
```

Position of 10-bit Address Header Sent (Master mode)

5.73.2.4 I2C_SR1_ADDR

```
#define I2C_SR1_ADDR I2C_SR1_ADDR_Msk
```

Address Sent (Master mode) or Matched (Slave mode)

5.73.2.5 I2C_SR1_ADDR_Msk

```
#define I2C_SR1_ADDR_Msk (0x1UL << I2C_SR1_ADDR_Pos)
```

Bit mask for Address Sent or Matched

5.73.2.6 I2C_SR1_ADDR_Pos

```
#define I2C_SR1_ADDR_Pos (1U)
```

Position of Address Sent (Master mode) or Matched (Slave mode)

5.73.2.7 I2C_SR1_AF

```
#define I2C_SR1_AF I2C_SR1_AF_Msk
```

Acknowledge Failure

5.73.2.8 I2C_SR1_AF_Msk

```
#define I2C_SR1_AF_Msk (0x1UL << I2C_SR1_AF_Pos)
```

Bit mask for Acknowledge Failure

5.73.2.9 I2C_SR1_AF_Pos

```
#define I2C_SR1_AF_Pos (10U)
```

Position of Acknowledge Failure bit

5.73.2.10 I2C_SR1_ARLO

```
#define I2C_SR1_ARLO I2C_SR1_ARLO_Msk
```

Arbitration Lost (Master mode)

5.73.2.11 I2C_SR1_ARLO_Msk

```
#define I2C_SR1_ARLO_Msk (0x1UL << I2C_SR1_ARLO_Pos)
```

Bit mask for Arbitration Lost (Master mode)

5.73.2.12 I2C_SR1_ARLO_Pos

```
#define I2C_SR1_ARLO_Pos (9U)
```

Position of Arbitration Lost (Master mode)

5.73.2.13 I2C_SR1_BERR

```
#define I2C_SR1_BERR I2C_SR1_BERR_Msk
```

Bus Error

5.73.2.14 I2C_SR1_BERR_Msk

```
#define I2C_SR1_BERR_Msk (0x1UL << I2C_SR1_BERR_Pos)
```

Bit mask for Bus Error

5.73.2.15 I2C_SR1_BERR_Pos

```
#define I2C_SR1_BERR_Pos (8U)
```

Position of Bus Error bit

5.73.2.16 I2C_SR1_BTF

```
#define I2C_SR1_BTF I2C_SR1_BTF_Msk
```

Byte Transfer Finished

5.73.2.17 I2C_SR1_BTF_Msk

```
#define I2C_SR1_BTF_Msk (0x1UL << I2C_SR1_BTF_Pos)
```

Bit mask for Byte Transfer Finished

5.73.2.18 I2C_SR1_BTF_Pos

```
#define I2C_SR1_BTF_Pos (2U)
```

Position of Byte Transfer Finished bit

5.73.2.19 I2C_SR1_OVR

```
#define I2C_SR1_OVR I2C_SR1_OVR_Msk
```

Overrun/Underrun

5.73.2.20 I2C_SR1_OVR_Msk

```
#define I2C_SR1_OVR_Msk (0x1UL << I2C_SR1_OVR_Pos)
```

Bit mask for Overrun/Underrun

5.73.2.21 I2C_SR1_OVR_Pos

```
#define I2C_SR1_OVR_Pos (11U)
```

Position of Overrun/Underrun bit

5.73.2.22 I2C_SR1_PECERR

```
#define I2C_SR1_PECERR I2C_SR1_PECERR_Msk
```

PEC Error in Reception

5.73.2.23 I2C_SR1_PECERR_Msk

```
#define I2C_SR1_PECERR_Msk (0x1UL << I2C_SR1_PECERR_Pos)
```

Bit mask for PEC Error in Reception

5.73.2.24 I2C_SR1_PECERR_Pos

```
#define I2C_SR1_PECERR_Pos (12U)
```

Position of PEC Error in Reception bit

5.73.2.25 I2C_SR1_RXNE

```
#define I2C_SR1_RXNE I2C_SR1_RXNE_Msk
```

Data Register Not Empty (Receiver)

5.73.2.26 I2C_SR1_RXNE_Msk

```
#define I2C_SR1_RXNE_Msk (0x1UL << I2C_SR1_RXNE_Pos)
```

Bit mask for Data Register Not Empty (Receiver)

5.73.2.27 I2C_SR1_RXNE_Pos

```
#define I2C_SR1_RXNE_Pos (6U)
```

Position of Data Register Not Empty (Receiver)

5.73.2.28 I2C_SR1_SB

```
#define I2C_SR1_SB I2C_SR1_SB_Msk
```

Start Bit (Master mode)

5.73.2.29 I2C_SR1_SB_Msk

```
#define I2C_SR1_SB_Msk (0x1UL << I2C_SR1_SB_Pos)
```

Bit mask for Start Bit (Master mode)

5.73.2.30 I2C_SR1_SB_Pos

```
#define I2C_SR1_SB_Pos (0U)
```

Bit definitions for the I2C_SR1 register.

Position of Start Bit (Master mode)

5.73.2.31 I2C_SR1_SMBALERT

```
#define I2C_SR1_SMBALERT I2C_SR1_SMBALERT_Msk
```

SMBus Alert

5.73.2.32 I2C_SR1_SMBALERT_Msk

```
#define I2C_SR1_SMBALERT_Msk (0x1UL << I2C_SR1_SMBALERT_Pos)
```

Bit mask for SMBus Alert

5.73.2.33 I2C_SR1_SMBALERT_Pos

```
#define I2C_SR1_SMBALERT_Pos (15U)
```

Position of SMBus Alert bit

5.73.2.34 I2C_SR1_STOPF

```
#define I2C_SR1_STOPF I2C_SR1_STOPF_Msk
```

Stop Detection (Slave mode)

5.73.2.35 I2C_SR1_STOPF_Msk

```
#define I2C_SR1_STOPF_Msk (0x1UL << I2C_SR1_STOPF_Pos)
```

Bit mask for Stop Detection (Slave mode)

5.73.2.36 I2C_SR1_STOPF_Pos

```
#define I2C_SR1_STOPF_Pos (4U)
```

Position of Stop Detection (Slave mode)

5.73.2.37 I2C_SR1_TIMEOUT

```
#define I2C_SR1_TIMEOUT I2C_SR1_TIMEOUT_Msk
```

Timeout or Tlow Error

5.73.2.38 I2C_SR1_TIMEOUT_Msk

```
#define I2C_SR1_TIMEOUT_Msk (0x1UL << I2C_SR1_TIMEOUT_Pos)
```

Bit mask for Timeout or Tlow Error

5.73.2.39 I2C_SR1_TIMEOUT_Pos

```
#define I2C_SR1_TIMEOUT_Pos (14U)
```

Position of Timeout or Tlow Error bit

5.73.2.40 I2C_SR1_TXE

```
#define I2C_SR1_TXE I2C_SR1_TXE_Msk
```

Data Register Empty (Transmitter)

5.73.2.41 I2C_SR1_TXE_Msk

```
#define I2C_SR1_TXE_Msk (0x1UL << I2C_SR1_TXE_Pos)
```

Bit mask for Data Register Empty (Transmitter)

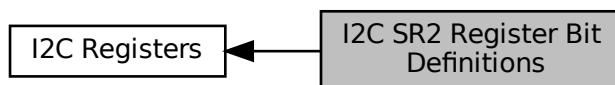
5.73.2.42 I2C_SR1_TXE_Pos

```
#define I2C_SR1_TXE_Pos (7U)
```

Position of Data Register Empty (Transmitter)

5.74 I2C SR2 Register Bit Definitions

Collaboration diagram for I2C SR2 Register Bit Definitions:



Macros

- #define I2C_SR2_MSL_Pos (0U)

Bit definitions for the I2C_SR2 register.
- #define I2C_SR2_MSL_Msk (0x1UL << I2C_SR2_MSL_Pos)
- #define I2C_SR2_MSL I2C_SR2_MSL_Msk
- #define I2C_SR2_BUSY_Pos (1U)
- #define I2C_SR2_BUSY_Msk (0x1UL << I2C_SR2_BUSY_Pos)
- #define I2C_SR2_BUSY I2C_SR2_BUSY_Msk
- #define I2C_SR2_TRA_Pos (2U)
- #define I2C_SR2_TRA_Msk (0x1UL << I2C_SR2_TRA_Pos)
- #define I2C_SR2_TRA I2C_SR2_TRA_Msk
- #define I2C_SR2_GENCALL_Pos (4U)
- #define I2C_SR2_GENCALL_Msk (0x1UL << I2C_SR2_GENCALL_Pos)
- #define I2C_SR2_GENCALL I2C_SR2_GENCALL_Msk
- #define I2C_SR2_SMBDEFAULT_Pos (5U)

- #define I2C_SR2_SMBDEFAULT_Msk (0x1UL << I2C_SR2_SMBDEFAULT_Pos)
- #define I2C_SR2_SMBDEFAULT I2C_SR2_SMBDEFAULT_Msk
- #define I2C_SR2_SMBHOST_Pos (6U)
- #define I2C_SR2_SMBHOST_Msk (0x1UL << I2C_SR2_SMBHOST_Pos)
- #define I2C_SR2_SMBHOST I2C_SR2_SMBHOST_Msk
- #define I2C_SR2_DUALF_Pos (7U)
- #define I2C_SR2_DUALF_Msk (0x1UL << I2C_SR2_DUALF_Pos)
- #define I2C_SR2_DUALF I2C_SR2_DUALF_Msk
- #define I2C_SR2_PEC_Pos (8U)
- #define I2C_SR2_PEC_Msk (0xFFUL << I2C_SR2_PEC_Pos)
- #define I2C_SR2_PEC I2C_SR2_PEC_Msk

5.74.1 Detailed Description

5.74.2 Macro Definition Documentation

5.74.2.1 I2C_SR2_BUSY

```
#define I2C_SR2_BUSY I2C_SR2_BUSY_Msk
```

Bus Busy

5.74.2.2 I2C_SR2_BUSY_Msk

```
#define I2C_SR2_BUSY_Msk (0x1UL << I2C_SR2_BUSY_Pos)
```

Bit mask for Bus Busy

5.74.2.3 I2C_SR2_BUSY_Pos

```
#define I2C_SR2_BUSY_Pos (1U)
```

Position of Bus Busy bit

5.74.2.4 I2C_SR2_DUALF

```
#define I2C_SR2_DUALF I2C_SR2_DUALF_Msk
```

Dual Flag (Slave mode)

5.74.2.5 I2C_SR2_DUALF_Msk

```
#define I2C_SR2_DUALF_Msk (0x1UL << I2C_SR2_DUALF_Pos)
```

Bit mask for Dual Flag (Slave mode)

5.74.2.6 I2C_SR2_DUALF_Pos

```
#define I2C_SR2_DUALF_Pos (7U)
```

Position of Dual Flag (Slave mode) bit

5.74.2.7 I2C_SR2_GENCALL

```
#define I2C_SR2_GENCALL I2C_SR2_GENCALL_Msk
```

General Call Address (Slave mode)

5.74.2.8 I2C_SR2_GENCALL_Msk

```
#define I2C_SR2_GENCALL_Msk (0x1UL << I2C_SR2_GENCALL_Pos)
```

Bit mask for General Call Address (Slave mode)

5.74.2.9 I2C_SR2_GENCALL_Pos

```
#define I2C_SR2_GENCALL_Pos (4U)
```

Position of General Call Address (Slave mode) bit

5.74.2.10 I2C_SR2_MSL

```
#define I2C_SR2_MSL I2C_SR2_MSL_Msk
```

Master/Slave

5.74.2.11 I2C_SR2_MSL_Msk

```
#define I2C_SR2_MSL_Msk (0x1UL << I2C_SR2_MSL_Pos)
```

Bit mask for Master/Slave

5.74.2.12 I2C_SR2_MSL_Pos

```
#define I2C_SR2_MSL_Pos (0U)
```

Bit definitions for the I2C_SR2 register.

Position of Master/Slave bit

5.74.2.13 I2C_SR2_PEC

```
#define I2C_SR2_PEC I2C_SR2_PEC_Msk
```

PEC Register (Slave mode)

5.74.2.14 I2C_SR2_PEC_Msk

```
#define I2C_SR2_PEC_Msk (0xFFUL << I2C_SR2_PEC_Pos)
```

Bit mask for PEC Register (Slave mode)

5.74.2.15 I2C_SR2_PEC_Pos

```
#define I2C_SR2_PEC_Pos (8U)
```

Position of PEC Register (Slave mode) bit

5.74.2.16 I2C_SR2_SMBDEFAULT

```
#define I2C_SR2_SMBDEFAULT I2C_SR2_SMBDEFAULT_Msk
```

SMBus Device Default Address (Slave mode)

5.74.2.17 I2C_SR2_SMBDEFAULT_Msk

```
#define I2C_SR2_SMBDEFAULT_Msk (0x1UL << I2C_SR2_SMBDEFAULT_Pos)
```

Bit mask for SMBus Device Default Address (Slave mode)

5.74.2.18 I2C_SR2_SMBDEFAULT_Pos

```
#define I2C_SR2_SMBDEFAULT_Pos (5U)
```

Position of SMBus Device Default Address (Slave mode) bit

5.74.2.19 I2C_SR2_SMBHOST

```
#define I2C_SR2_SMBHOST I2C_SR2_SMBHOST_Msk
```

SMBus Host Header (Slave mode)

5.74.2.20 I2C_SR2_SMBHOST_Msk

```
#define I2C_SR2_SMBHOST_Msk (0x1UL << I2C_SR2_SMBHOST_Pos)
```

Bit mask for SMBus Host Header (Slave mode)

5.74.2.21 I2C_SR2_SMBHOST_Pos

```
#define I2C_SR2_SMBHOST_Pos (6U)
```

Position of SMBus Host Header (Slave mode) bit

5.74.2.22 I2C_SR2_TRA

```
#define I2C_SR2_TRA I2C_SR2_TRA_Msk
```

Transmitter/Receiver

5.74.2.23 I2C_SR2_TRA_Msk

```
#define I2C_SR2_TRA_Msk (0x1UL << I2C_SR2_TRA_Pos)
```

Bit mask for Transmitter/Receiver

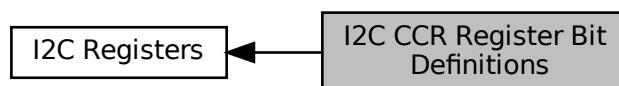
5.74.2.24 I2C_SR2_TRA_Pos

```
#define I2C_SR2_TRA_Pos (2U)
```

Position of Transmitter/Receiver bit

5.75 I2C CCR Register Bit Definitions

Collaboration diagram for I2C CCR Register Bit Definitions:



Macros

- #define I2C_CCR_CCR_Pos (0U)
Bit definitions for the I2C_CCR register.
- #define I2C_CCR_CCR_Msk (0xFFFFUL << I2C_CCR_CCR_Pos)
- #define I2C_CCR_CCR I2C_CCR_CCR_Msk
- #define I2C_CCR_DUTY_Pos (14U)
- #define I2C_CCR_DUTY_Msk (0x1UL << I2C_CCR_DUTY_Pos)
- #define I2C_CCR_DUTY I2C_CCR_DUTY_Msk

5.75.1 Detailed Description

5.75.2 Macro Definition Documentation

5.75.2.1 I2C_CCR_CCR

```
#define I2C_CCR_CCR I2C_CCR_CCR_Msk
```

Clock Control Register

5.75.2.2 I2C_CCR_CCR_Msk

```
#define I2C_CCR_CCR_Msk (0xFFFFUL << I2C_CCR_CCR_Pos)
```

Bit mask for Clock Control Register

5.75.2.3 I2C_CCR_CCR_Pos

```
#define I2C_CCR_CCR_Pos (0U)
```

Bit definitions for the I2C_CCR register.

Position of Clock Control Register

5.75.2.4 I2C_CCR_DUTY

```
#define I2C_CCR_DUTY I2C_CCR_DUTY_Msk
```

Duty Cycle bit

5.75.2.5 I2C_CCR_DUTY_Msk

```
#define I2C_CCR_DUTY_Msk (0x1UL << I2C_CCR_DUTY_Pos)
```

Bit mask for Duty Cycle bit

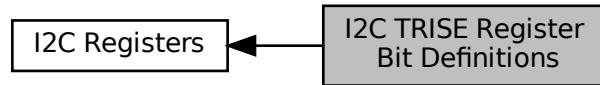
5.75.2.6 I2C_CCR_DUTY_Pos

```
#define I2C_CCR_DUTY_Pos (14U)
```

Position of Duty Cycle bit

5.76 I2C TRISE Register Bit Definitions

Collaboration diagram for I2C TRISE Register Bit Definitions:



Macros

- `#define I2C_TRISE_TRISE_Pos (0U)`
Bit definitions for the I2C_TRISE register.
- `#define I2C_TRISE_TRISE_Msk (0x3FUL << I2C_TRISE_TRISE_Pos)`
- `#define I2C_TRISE_TRISE I2C_TRISE_TRISE_Msk`

5.76.1 Detailed Description

5.76.2 Macro Definition Documentation

5.76.2.1 I2C_TRISE_TRISE

```
#define I2C_TRISE_TRISE I2C_TRISE_TRISE_Msk
```

TRISE bits

5.76.2.2 I2C_TRISE_TRISE_Msk

```
#define I2C_TRISE_TRISE_Msk (0x3FUL << I2C_TRISE_TRISE_Pos)
```

Bit mask for TRISE bits

5.76.2.3 I2C_TRISE_TRISE_Pos

```
#define I2C_TRISE_TRISE_Pos (0U)
```

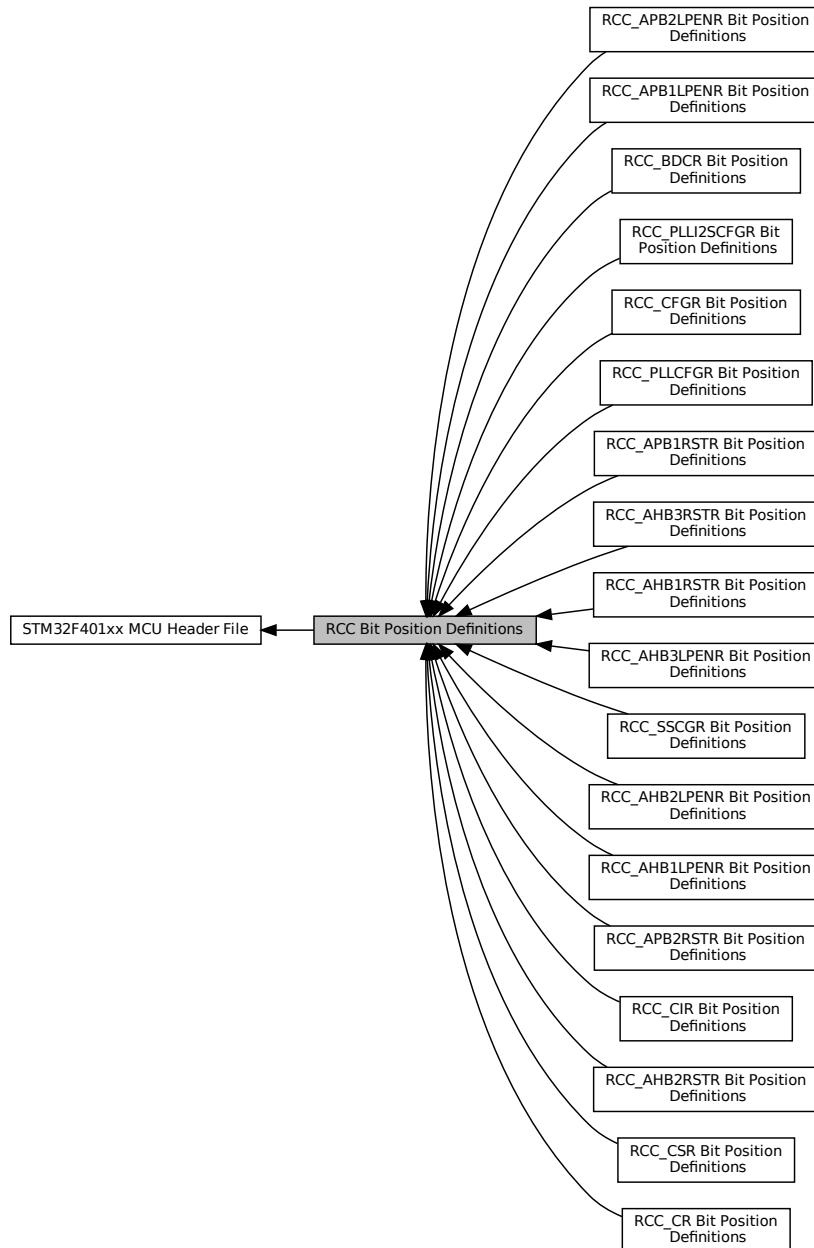
Bit definitions for the I2C_TRISE register.

Position of TRISE bits

5.77 RCC Bit Position Definitions

Bit position definitions for various registers in the RCC peripheral.

Collaboration diagram for RCC Bit Position Definitions:



Modules

- [RCC_CR Bit Position Definitions](#)
Bit position definitions for RCC_CR register.
- [RCC_PLLCFGGR Bit Position Definitions](#)

- [RCC_CFGR Bit Position Definitions](#)
Bit position definitions for RCC_PLLCFGR register.
- [RCC_CIR Bit Position Definitions](#)
Bit position definitions for RCC_CIR register.
- [RCC_AHB1RSTR Bit Position Definitions](#)
Bit position definitions for RCC_AHB1RSTR register.
- [RCC_AHB2RSTR Bit Position Definitions](#)
Bit position definitions for RCC_AHB2RSTR register.
- [RCC_AHB3RSTR Bit Position Definitions](#)
Bit position definitions for RCC_AHB3RSTR register.
- [RCC_APB1RSTR Bit Position Definitions](#)
Bit position definitions for RCC_APB1RSTR register.
- [RCC_APB2RSTR Bit Position Definitions](#)
Bit position definitions for RCC_APB2RSTR register.
- [RCC_AHB1LPENR Bit Position Definitions](#)
Bit position definitions for RCC_AHB1LPENR register.
- [RCC_AHB2LPENR Bit Position Definitions](#)
Bit position definitions for RCC_AHB2LPENR register.
- [RCC_AHB3LPENR Bit Position Definitions](#)
Bit position definitions for RCC_AHB3LPENR register.
- [RCC_APB1LPENR Bit Position Definitions](#)
Bit position definitions for RCC_APB1LPENR register.
- [RCC_APB2LPENR Bit Position Definitions](#)
Bit position definitions for RCC_APB2LPENR register.
- [RCC_BDCR Bit Position Definitions](#)
Bit position definitions for RCC_BDCR register.
- [RCC_CSR Bit Position Definitions](#)
Bit position definitions for RCC_CSR register.
- [RCC_SSCGR Bit Position Definitions](#)
Bit position definitions for RCC_SSCGR register.
- [RCC_PLLI2SCFGR Bit Position Definitions](#)
Bit position definitions for RCC_PLLI2SCFGR register.

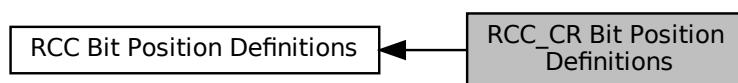
5.77.1 Detailed Description

Bit position definitions for various registers in the RCC peripheral.

5.78 RCC_CR Bit Position Definitions

Bit position definitions for RCC_CR register.

Collaboration diagram for RCC_CR Bit Position Definitions:



Macros

- #define RCC_CR_HSION 0
- #define RCC_CR_HSIRDY 1
- #define RCC_CR_HSITRIM 3
- #define RCC_CR_HSICAL 8
- #define RCC_CR_HSEON 16
- #define RCC_CR_HSERDY 17
- #define RCC_CR_HSEBYP 18
- #define RCC_CR_CSSON 19
- #define RCC_CR_PLLON 24
- #define RCC_CR_PLLRDY 25
- #define RCC_CR_PLLI2SON 26
- #define RCC_CR_PLLI2SRDY 27
- #define RCC_CR_PLLSAION 28
- #define RCC_CR_PLLSAIRDY 29

5.78.1 Detailed Description

Bit position definitions for RCC_CR register.

5.78.2 Macro Definition Documentation

5.78.2.1 RCC_CR_CSSON

```
#define RCC_CR_CSSON 19
```

Clock Security System Enable

5.78.2.2 RCC_CR_HSEBYP

```
#define RCC_CR_HSEBYP 18
```

HSE Oscillator Bypass

5.78.2.3 RCC_CR_HSEON

```
#define RCC_CR_HSEON 16
```

HSE Oscillator Enable

5.78.2.4 RCC_CR_HSERDY

```
#define RCC_CR_HSERDY 17
```

HSE Oscillator Ready

5.78.2.5 RCC_CR_HSICAL

```
#define RCC_CR_HSICAL 8
```

HSI Oscillator Calibration

5.78.2.6 RCC_CR_HSION

```
#define RCC_CR_HSION 0
```

HSI Oscillator Enable

5.78.2.7 RCC_CR_HSIRDY

```
#define RCC_CR_HSIRDY 1
```

HSI Oscillator Ready

5.78.2.8 RCC_CR_HSITRIM

```
#define RCC_CR_HSITRIM 3
```

HSI Oscillator Trimming

5.78.2.9 RCC_CR_PLLI2SON

```
#define RCC_CR_PLLI2SON 26
```

PLL2S Enable

5.78.2.10 RCC_CR_PLLI2SRDY

```
#define RCC_CR_PLLI2SRDY 27
```

PLL2S Ready

5.78.2.11 RCC_CR_PLLON

```
#define RCC_CR_PLLON 24
```

Main PLL Enable

5.78.2.12 RCC_CR_PLLRDY

```
#define RCC_CR_PLLRDY 25
```

Main PLL Ready

5.78.2.13 RCC_CR_PLLSAION

```
#define RCC_CR_PLLSAION 28
```

PLLSAI Enable

5.78.2.14 RCC_CR_PLLSAIRDY

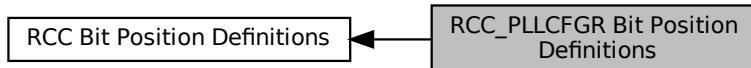
```
#define RCC_CR_PLLSAIRDY 29
```

PLLSAI Ready

5.79 RCC_PLLCFGR Bit Position Definitions

Bit position definitions for RCC_PLLCFGR register.

Collaboration diagram for RCC_PLLCFGR Bit Position Definitions:



Macros

- #define RCC_PLLCFGR_PLLM 0
- #define RCC_PLLCFGR_PLLN 6
- #define RCC_PLLCFGR_PLLP 16
- #define RCC_PLLCFGR_PLLSRC 22
- #define RCC_PLLCFGR_PLLQ 24

5.79.1 Detailed Description

Bit position definitions for RCC_PLLCFGR register.

5.79.2 Macro Definition Documentation

5.79.2.1 RCC_PLLCFGR_PLLM

```
#define RCC_PLLCFGR_PLLM 0
```

Main PLL Division Factor for PLL VCO

5.79.2.2 RCC_PLLCFGR_PLLN

```
#define RCC_PLLCFGR_PLLN 6
```

Main PLL Multiplication Factor for VCO

5.79.2.3 RCC_PLLCFGR_PLLP

```
#define RCC_PLLCFGR_PLLP 16
```

Main PLL Division Factor for Main System Clock

5.79.2.4 RCC_PLLCFGR_PLLQ

```
#define RCC_PLLCFGR_PLLQ 24
```

Main PLLQ Division Factor for PLLI2S Clock Output

5.79.2.5 RCC_PLLCFGR_PLLSRC

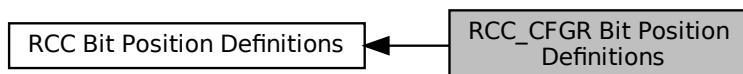
```
#define RCC_PLLCFGR_PLLSRC 22
```

Main PLL, PLLI2S, and PLLSAI Entry Clock Source

5.80 RCC_CFGR Bit Position Definitions

Bit position definitions for RCC_CFGR register.

Collaboration diagram for RCC_CFGR Bit Position Definitions:



Macros

- #define RCC_CFGR_SW 0
- #define RCC_CFGR_SWS 2
- #define RCC_CFGR_HPRE 4
- #define RCC_CFGR_PPREG 10
- #define RCC_CFGR_PPREG2 13
- #define RCC_CFGR_RTCPPREG 16
- #define RCC_CFGR_MCO1 21
- #define RCC_CFGR_I2SSRC 23
- #define RCC_CFGR_MCO1PPREG 24
- #define RCC_CFGR_MCO2PPREG 27
- #define RCC_CFGR_MCO2 30

5.80.1 Detailed Description

Bit position definitions for RCC_CFGR register.

5.80.2 Macro Definition Documentation

5.80.2.1 RCC_CFGR_HPRE

```
#define RCC_CFGR_HPRE 4
```

AHB Prescaler

5.80.2.2 RCC_CFGR_I2SSRC

```
#define RCC_CFGR_I2SSRC 23
```

I2S APB2 Clock Source Selection

5.80.2.3 RCC_CFGR_MCO1

```
#define RCC_CFGR_MCO1 21
```

Microcontroller Clock Output 1

5.80.2.4 RCC_CFGR_MCO1PPREG

```
#define RCC_CFGR_MCO1PPREG 24
```

MCO1 Prescaler

5.80.2.5 RCC_CFGR_MCO2

```
#define RCC_CFGR_MCO2 30
```

Microcontroller Clock Output 2

5.80.2.6 RCC_CFGR_MCO2PRE

```
#define RCC_CFGR_MCO2PRE 27
```

MCO2 Prescaler

5.80.2.7 RCC_CFGR_PPREG1

```
#define RCC_CFGR_PPREG1 10
```

APB1 Low-Speed Prescaler (APB1CLK)

5.80.2.8 RCC_CFGR_PPREG2

```
#define RCC_CFGR_PPREG2 13
```

APB2 High-Speed Prescaler (APB2CLK)

5.80.2.9 RCC_CFGR_RTCPRE

```
#define RCC_CFGR_RTCPRE 16
```

HSE division factor for RTC clock

5.80.2.10 RCC_CFGR_SW

```
#define RCC_CFGR_SW 0
```

System Clock Switch

5.80.2.11 RCC_CFGR_SWS

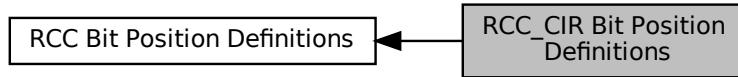
```
#define RCC_CFGR_SWS 2
```

System Clock Switch Status

5.81 RCC_CIR Bit Position Definitions

Bit position definitions for RCC_CIR register.

Collaboration diagram for RCC_CIR Bit Position Definitions:



Macros

- #define RCC_CIR_LSIRDYF 0
- #define RCC_CIR_LSERDYF 1
- #define RCC_CIR_HSIRDYF 2
- #define RCC_CIR_HSERDYF 3
- #define RCC_CIR_PLLRDYF 4
- #define RCC_CIR_PLLI2SRDYF 5
- #define RCC_CIR_PLLSAIRDYF 6
- #define RCC_CIR_CSSF 7
- #define RCC_CIR_LSIRDYIE 8
- #define RCC_CIR_LSERDYIE 9
- #define RCC_CIR_HSIRDYIE 10
- #define RCC_CIR_HSERDYIE 11
- #define RCC_CIR_PLLRDYIE 12
- #define RCC_CIR_PLLI2SRDYIE 13
- #define RCC_CIR_PLLSAIRDYIE 14
- #define RCC_CIR_LSIRDYC 16
- #define RCC_CIR_LSERDYC 17
- #define RCC_CIR_HSIRDYC 18
- #define RCC_CIR_HSERDYC 19
- #define RCC_CIR_PLLRDYC 20
- #define RCC_CIR_PLLI2SRDYC 21
- #define RCC_CIR_PLLSAIRDYC 22

5.81.1 Detailed Description

Bit position definitions for RCC_CIR register.

5.81.2 Macro Definition Documentation

5.81.2.1 RCC_CIR_CSSF

```
#define RCC_CIR_CSSF 7
```

Clock Security System Interrupt flag

5.81.2.2 RCC_CIR_HSERDYC

```
#define RCC_CIR_HSERDYC 19
```

HSE Ready Interrupt Clear

5.81.2.3 RCC_CIR_HSERDYF

```
#define RCC_CIR_HSERDYF 3
```

HSE Ready Interrupt flag

5.81.2.4 RCC_CIR_HSERDYIE

```
#define RCC_CIR_HSERDYIE 11
```

HSE Ready Interrupt Enable

5.81.2.5 RCC_CIR_HSIRDYC

```
#define RCC_CIR_HSIRDYC 18
```

HSI Ready Interrupt Clear

5.81.2.6 RCC_CIR_HSIRDYF

```
#define RCC_CIR_HSIRDYF 2
```

HSI Ready Interrupt flag

5.81.2.7 RCC_CIR_HSIRDYIE

```
#define RCC_CIR_HSIRDYIE 10
```

HSI Ready Interrupt Enable

5.81.2.8 RCC_CIR_LSERDYC

```
#define RCC_CIR_LSERDYC 17
```

LSE Ready Interrupt Clear

5.81.2.9 RCC_CIR_LSERDYF

```
#define RCC_CIR_LSERDYF 1
```

LSE Ready Interrupt flag

5.81.2.10 RCC_CIR_LSERDYIE

```
#define RCC_CIR_LSERDYIE 9
```

LSE Ready Interrupt Enable

5.81.2.11 RCC_CIR_LSIRDYC

```
#define RCC_CIR_LSIRDYC 16
```

LSI Ready Interrupt Clear

5.81.2.12 RCC_CIR_LSIRDYF

```
#define RCC_CIR_LSIRDYF 0
```

LSI Ready Interrupt flag

5.81.2.13 RCC_CIR_LSIRDYIE

```
#define RCC_CIR_LSIRDYIE 8
```

LSI Ready Interrupt Enable

5.81.2.14 RCC_CIR_PLLI2SRDYC

```
#define RCC_CIR_PLLI2SRDYC 21
```

PLL2S Ready Interrupt Clear

5.81.2.15 RCC_CIR_PLLI2SRDYF

```
#define RCC_CIR_PLLI2SRDYF 5
```

PLL2S Ready Interrupt flag

5.81.2.16 RCC_CIR_PLLI2SRDYIE

```
#define RCC_CIR_PLLI2SRDYIE 13
```

PLL2S Ready Interrupt Enable

5.81.2.17 RCC_CIR_PLLRDYC

```
#define RCC_CIR_PLLRDYC 20
```

PLL Ready Interrupt Clear

5.81.2.18 RCC_CIR_PLLRDYF

```
#define RCC_CIR_PLLRDYF 4
```

PLL Ready Interrupt flag

5.81.2.19 RCC_CIR_PLLRDYIE

```
#define RCC_CIR_PLLRDYIE 12
```

PLL Ready Interrupt Enable

5.81.2.20 RCC_CIR_PLLSAIRDYC

```
#define RCC_CIR_PLLSAIRDYC 22
```

PLLSAI Ready Interrupt Clear

5.81.2.21 RCC_CIR_PLLSAIRDYF

```
#define RCC_CIR_PLLSAIRDYF 6
```

PLLSAI Ready Interrupt flag

5.81.2.22 RCC_CIR_PLLSAIRDYIE

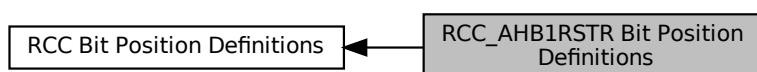
```
#define RCC_CIR_PLLSAIRDYIE 14
```

PLLSAI Ready Interrupt Enable

5.82 RCC_AHB1RSTR Bit Position Definitions

Bit position definitions for RCC_AHB1RSTR register.

Collaboration diagram for RCC_AHB1RSTR Bit Position Definitions:



Macros

- #define RCC_AHB1RSTR_GPIOA 0
- #define RCC_AHB1RSTR_GPIOB 1
- #define RCC_AHB1RSTR_GPIOC 2
- #define RCC_AHB1RSTR_GPIOD 3
- #define RCC_AHB1RSTR_GPIOE 4
- #define RCC_AHB1RSTR_GPIOF 5
- #define RCC_AHB1RSTR_GPIOG 6
- #define RCC_AHB1RSTR_GPIOH 7
- #define RCC_AHB1RSTR_GPIOI 8
- #define RCC_AHB1RSTR_CRC 12
- #define RCC_AHB1RSTR_DMA1 21
- #define RCC_AHB1RSTR_DMA2 22
- #define RCC_AHB1RSTR_ETHMAC 25
- #define RCC_AHB1RSTR_OTGHS 29
- #define RCC_AHB1RSTR_OTGHSULPI 30

5.82.1 Detailed Description

Bit position definitions for RCC_AHB1RSTR register.

5.82.2 Macro Definition Documentation

5.82.2.1 RCC_AHB1RSTR_CRC

```
#define RCC_AHB1RSTR_CRC 12
```

CRC Reset

5.82.2.2 RCC_AHB1RSTR_DMA1

```
#define RCC_AHB1RSTR_DMA1 21
```

DMA1 Reset

5.82.2.3 RCC_AHB1RSTR_DMA2

```
#define RCC_AHB1RSTR_DMA2 22
```

DMA2 Reset

5.82.2.4 RCC_AHB1RSTR_ETHMAC

```
#define RCC_AHB1RSTR_ETHMAC 25
```

Ethernet MAC Reset

5.82.2.5 RCC_AHB1RSTR_GPIOA

```
#define RCC_AHB1RSTR_GPIOA 0
```

GPIOA Reset

5.82.2.6 RCC_AHB1RSTR_GPIOB

```
#define RCC_AHB1RSTR_GPIOB 1
```

GPIOB Reset

5.82.2.7 RCC_AHB1RSTR_GPIOC

```
#define RCC_AHB1RSTR_GPIOC 2
```

GPIOC Reset

5.82.2.8 RCC_AHB1RSTR_GPIOD

```
#define RCC_AHB1RSTR_GPIOD 3
```

GPIOD Reset

5.82.2.9 RCC_AHB1RSTR_GPIOE

```
#define RCC_AHB1RSTR_GPIOE 4
```

GPIOE Reset

5.82.2.10 RCC_AHB1RSTR_GPIOF

```
#define RCC_AHB1RSTR_GPIOF 5
```

GPIOF Reset

5.82.2.11 RCC_AHB1RSTR_GPIOG

```
#define RCC_AHB1RSTR_GPIOG 6
```

GPIOG Reset

5.82.2.12 RCC_AHB1RSTR_GPIOH

```
#define RCC_AHB1RSTR_GPIOH 7
```

GPIOH Reset

5.82.2.13 RCC_AHB1RSTR_GPIOI

```
#define RCC_AHB1RSTR_GPIOI 8
```

GPIOI Reset

5.82.2.14 RCC_AHB1RSTR_OTGHS

```
#define RCC_AHB1RSTR_OTGHS 29
```

USB OTG HS Reset

5.82.2.15 RCC_AHB1RSTR_OTGHSULPI

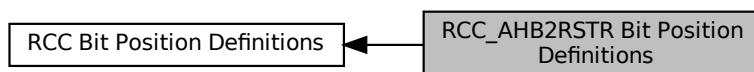
```
#define RCC_AHB1RSTR_OTGHSULPI 30
```

USB OTG HS ULPI Reset

5.83 RCC_AHB2RSTR Bit Position Definitions

Bit position definitions for RCC_AHB2RSTR register.

Collaboration diagram for RCC_AHB2RSTR Bit Position Definitions:



Macros

- #define RCC_AHB2RSTR_DCMI 0
- #define RCC_AHB2RSTR_CRYP 4
- #define RCC_AHB2RSTR_HASH 5
- #define RCC_AHB2RSTR_RNG 6
- #define RCC_AHB2RSTR_OTGFS 7

5.83.1 Detailed Description

Bit position definitions for RCC_AHB2RSTR register.

5.83.2 Macro Definition Documentation

5.83.2.1 RCC_AHB2RSTR_CRYP

```
#define RCC_AHB2RSTR_CRYP 4
```

CRYP Reset

5.83.2.2 RCC_AHB2RSTR_DCMI

```
#define RCC_AHB2RSTR_DCMI 0
```

DCMI Reset

5.83.2.3 RCC_AHB2RSTR_HASH

```
#define RCC_AHB2RSTR_HASH 5
```

HASH Reset

5.83.2.4 RCC_AHB2RSTR_OTGFS

```
#define RCC_AHB2RSTR_OTGFS 7
```

USB OTG FS Reset

5.83.2.5 RCC_AHB2RSTR_RNG

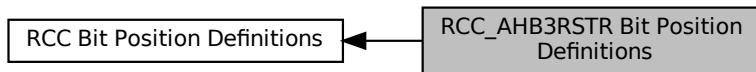
```
#define RCC_AHB2RSTR_RNG 6
```

RNG Reset

5.84 RCC_AHB3RSTR Bit Position Definitions

Bit position definitions for RCC_AHB3RSTR register.

Collaboration diagram for RCC_AHB3RSTR Bit Position Definitions:



Macros

- #define RCC_AHB3RSTR_FSMC 0

5.84.1 Detailed Description

Bit position definitions for RCC_AHB3RSTR register.

5.84.2 Macro Definition Documentation

5.84.2.1 RCC_AHB3RSTR_FSMC

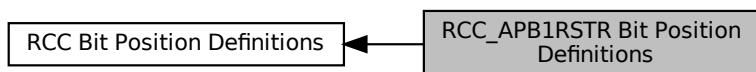
```
#define RCC_AHB3RSTR_FSMC 0
```

FSMC Reset

5.85 RCC_APB1RSTR Bit Position Definitions

Bit position definitions for RCC_APB1RSTR register.

Collaboration diagram for RCC_APB1RSTR Bit Position Definitions:



Macros

- #define RCC_APB1RSTR_TIM2 0
- #define RCC_APB1RSTR_TIM3 1
- #define RCC_APB1RSTR_TIM4 2
- #define RCC_APB1RSTR_TIM5 3
- #define RCC_APB1RSTR_TIM6 4
- #define RCC_APB1RSTR_TIM7 5
- #define RCC_APB1RSTR_TIM12 6
- #define RCC_APB1RSTR_TIM13 7
- #define RCC_APB1RSTR_TIM14 8
- #define RCC_APB1RSTR_WWDG 11
- #define RCC_APB1RSTR_SPI2 14
- #define RCC_APB1RSTR_SPI3 15
- #define RCC_APB1RSTR_USART2 17
- #define RCC_APB1RSTR_USART3 18
- #define RCC_APB1RSTR_UART4 19
- #define RCC_APB1RSTR_UART5 20
- #define RCC_APB1RSTR_I2C1 21
- #define RCC_APB1RSTR_I2C2 22
- #define RCC_APB1RSTR_I2C3 23
- #define RCC_APB1RSTR_CAN1 25
- #define RCC_APB1RSTR_CAN2 26
- #define RCC_APB1RSTR_PWR 28
- #define RCC_APB1RSTR_DAC 29
- #define RCC_APB1RSTR_UART7 30
- #define RCC_APB1RSTR_UART8 31

5.85.1 Detailed Description

Bit position definitions for RCC_APB1RSTR register.

5.85.2 Macro Definition Documentation

5.85.2.1 RCC_APB1RSTR_CAN1

```
#define RCC_APB1RSTR_CAN1 25
```

CAN1 Reset

5.85.2.2 RCC_APB1RSTR_CAN2

```
#define RCC_APB1RSTR_CAN2 26
```

CAN2 Reset

5.85.2.3 RCC_APB1RSTR_DAC

```
#define RCC_APB1RSTR_DAC 29
```

DAC Reset

5.85.2.4 RCC_APB1RSTR_I2C1

```
#define RCC_APB1RSTR_I2C1 21
```

I2C1 Reset

5.85.2.5 RCC_APB1RSTR_I2C2

```
#define RCC_APB1RSTR_I2C2 22
```

I2C2 Reset

5.85.2.6 RCC_APB1RSTR_I2C3

```
#define RCC_APB1RSTR_I2C3 23
```

I2C3 Reset

5.85.2.7 RCC_APB1RSTR_PWR

```
#define RCC_APB1RSTR_PWR 28
```

Power Interface Reset

5.85.2.8 RCC_APB1RSTR_SPI2

```
#define RCC_APB1RSTR_SPI2 14
```

SPI2 Reset

5.85.2.9 RCC_APB1RSTR_SPI3

```
#define RCC_APB1RSTR_SPI3 15
```

SPI3 Reset

5.85.2.10 RCC_APB1RSTR_TIM12

```
#define RCC_APB1RSTR_TIM12 6
```

TIM12 Reset

5.85.2.11 RCC_APB1RSTR_TIM13

```
#define RCC_APB1RSTR_TIM13 7
```

TIM13 Reset

5.85.2.12 RCC_APB1RSTR_TIM14

```
#define RCC_APB1RSTR_TIM14 8
```

TIM14 Reset

5.85.2.13 RCC_APB1RSTR_TIM2

```
#define RCC_APB1RSTR_TIM2 0
```

TIM2 Reset

5.85.2.14 RCC_APB1RSTR_TIM3

```
#define RCC_APB1RSTR_TIM3 1
```

TIM3 Reset

5.85.2.15 RCC_APB1RSTR_TIM4

```
#define RCC_APB1RSTR_TIM4 2
```

TIM4 Reset

5.85.2.16 RCC_APB1RSTR_TIM5

```
#define RCC_APB1RSTR_TIM5 3
```

TIM5 Reset

5.85.2.17 RCC_APB1RSTR_TIM6

```
#define RCC_APB1RSTR_TIM6 4
```

TIM6 Reset

5.85.2.18 RCC_APB1RSTR_TIM7

```
#define RCC_APB1RSTR_TIM7 5
```

TIM7 Reset

5.85.2.19 RCC_APB1RSTR_UART4

```
#define RCC_APB1RSTR_UART4 19
```

UART4 Reset

5.85.2.20 RCC_APB1RSTR_UART5

```
#define RCC_APB1RSTR_UART5 20
```

UART5 Reset

5.85.2.21 RCC_APB1RSTR_UART7

```
#define RCC_APB1RSTR_UART7 30
```

UART7 Reset

5.85.2.22 RCC_APB1RSTR_UART8

```
#define RCC_APB1RSTR_UART8 31
```

UART8 Reset

5.85.2.23 RCC_APB1RSTR_USART2

```
#define RCC_APB1RSTR_USART2 17
```

USART2 Reset

5.85.2.24 RCC_APB1RSTR_USART3

```
#define RCC_APB1RSTR_USART3 18
```

USART3 Reset

5.85.2.25 RCC_APB1RSTR_WWDG

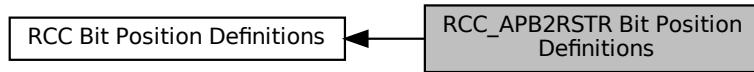
```
#define RCC_APB1RSTR_WWDG 11
```

WWDG Reset

5.86 RCC_APB2RSTR Bit Position Definitions

Bit position definitions for RCC_APB2RSTR register.

Collaboration diagram for RCC_APB2RSTR Bit Position Definitions:



Macros

- #define RCC_APB2RSTR_TIM1 0
- #define RCC_APB2RSTR_TIM8 1
- #define RCC_APB2RSTR_USART1 4
- #define RCC_APB2RSTR_USART6 5
- #define RCC_APB2RSTR_ADC 8
- #define RCC_APB2RSTR_SDIO 11
- #define RCC_APB2RSTR_SPI1 12
- #define RCC_APB2RSTR_SYSCFG 14
- #define RCC_APB2RSTR_TIM9 16
- #define RCC_APB2RSTR_TIM10 17
- #define RCC_APB2RSTR_TIM11 18

5.86.1 Detailed Description

Bit position definitions for RCC_APB2RSTR register.

5.86.2 Macro Definition Documentation

5.86.2.1 RCC_APB2RSTR_ADC

```
#define RCC_APB2RSTR_ADC 8
```

ADC Reset

5.86.2.2 RCC_APB2RSTR_SDIO

```
#define RCC_APB2RSTR_SDIO 11
```

SDIO Reset

5.86.2.3 RCC_APB2RSTR_SPI1

```
#define RCC_APB2RSTR_SPI1 12
```

SPI1 Reset

5.86.2.4 RCC_APB2RSTR_SYSCFG

```
#define RCC_APB2RSTR_SYSCFG 14
```

System Configuration Controller Reset

5.86.2.5 RCC_APB2RSTR_TIM1

```
#define RCC_APB2RSTR_TIM1 0
```

TIM1 Reset

5.86.2.6 RCC_APB2RSTR_TIM10

```
#define RCC_APB2RSTR_TIM10 17
```

TIM10 Reset

5.86.2.7 RCC_APB2RSTR_TIM11

```
#define RCC_APB2RSTR_TIM11 18
```

TIM11 Reset

5.86.2.8 RCC_APB2RSTR_TIM8

```
#define RCC_APB2RSTR_TIM8 1
```

TIM8 Reset

5.86.2.9 RCC_APB2RSTR_TIM9

```
#define RCC_APB2RSTR_TIM9 16
```

TIM9 Reset

5.86.2.10 RCC_APB2RSTR_USART1

```
#define RCC_APB2RSTR_USART1 4
```

USART1 Reset

5.86.2.11 RCC_APB2RSTR_USART6

```
#define RCC_APB2RSTR_USART6 5
```

USART6 Reset

5.87 RCC_AHB1LPENR Bit Position Definitions

Bit position definitions for RCC_AHB1LPENR register.

Collaboration diagram for RCC_AHB1LPENR Bit Position Definitions:



Macros

- #define RCC_AHB1LPENR_GPIOALPEN 0
- #define RCC_AHB1LPENR_GPIOBLPEN 1
- #define RCC_AHB1LPENR_GPIOCLPEN 2
- #define RCC_AHB1LPENR_GPIODLPEN 3
- #define RCC_AHB1LPENR_GPIOELPEN 4
- #define RCC_AHB1LPENR_GPIOFLPEN 5
- #define RCC_AHB1LPENR_GPIOGLPEN 6
- #define RCC_AHB1LPENR_GPIOHLPEN 7
- #define RCC_AHB1LPENR_GPIOILPEN 8
- #define RCC_AHB1LPENR_CRCEN 12
- #define RCC_AHB1LPENR_DMA1LPEN 21
- #define RCC_AHB1LPENR_DMA2LPEN 22
- #define RCC_AHB1LPENR_ETHMACLPEN 25
- #define RCC_AHB1LPENR_ETHMACTXLPEN 26
- #define RCC_AHB1LPENR_ETHMACRXLPEN 27
- #define RCC_AHB1LPENR_ETHMACPTPLPEN 28
- #define RCC_AHB1LPENR_OTGHSLPEN 29
- #define RCC_AHB1LPENR_OTGHSHULPI 30

5.87.1 Detailed Description

Bit position definitions for RCC_AHB1LPENR register.

5.87.2 Macro Definition Documentation

5.87.2.1 RCC_AHB1LPENR_CRCEN

```
#define RCC_AHB1LPENR_CRCEN 12
```

CRC Peripheral Clock in Low Power Mode Enable

5.87.2.2 RCC_AHB1LPENR_DMA1LPEN

```
#define RCC_AHB1LPENR_DMA1LPEN 21
```

DMA1 Peripheral Clock in Low Power Mode Enable

5.87.2.3 RCC_AHB1LPENR_DMA2LPEN

```
#define RCC_AHB1LPENR_DMA2LPEN 22
```

DMA2 Peripheral Clock in Low Power Mode Enable

5.87.2.4 RCC_AHB1LPENR_ETHMACLPEN

```
#define RCC_AHB1LPENR_ETHMACLPEN 25
```

Ethernet MAC Peripheral Clock in Low Power Mode Enable

5.87.2.5 RCC_AHB1LPENR_ETHMACPTPLPEN

```
#define RCC_AHB1LPENR_ETHMACPTPLPEN 28
```

Ethernet MAC PTP Peripheral Clock in Low Power Mode Enable

5.87.2.6 RCC_AHB1LPENR_ETHMACRXLPEN

```
#define RCC_AHB1LPENR_ETHMACRXLPEN 27
```

Ethernet MAC Receive Peripheral Clock in Low Power Mode Enable

5.87.2.7 RCC_AHB1LPENR_ETHMACTXLPEN

```
#define RCC_AHB1LPENR_ETHMACTXLPEN 26
```

Ethernet MAC Transmit Peripheral Clock in Low Power Mode Enable

5.87.2.8 RCC_AHB1LPENR_GPIOALPEN

```
#define RCC_AHB1LPENR_GPIOALPEN 0
```

GPIOA Peripheral Clock in Low Power Mode Enable

5.87.2.9 RCC_AHB1LPENR_GPIOBLPEN

```
#define RCC_AHB1LPENR_GPIOBLPEN 1
```

GPIOB Peripheral Clock in Low Power Mode Enable

5.87.2.10 RCC_AHB1LPENR_GPIOCLPEN

```
#define RCC_AHB1LPENR_GPIOCLPEN 2
```

GPIOC Peripheral Clock in Low Power Mode Enable

5.87.2.11 RCC_AHB1LPENR_GPIODLPEN

```
#define RCC_AHB1LPENR_GPIODLPEN 3
```

GPIOD Peripheral Clock in Low Power Mode Enable

5.87.2.12 RCC_AHB1LPENR_GPIOELPEN

```
#define RCC_AHB1LPENR_GPIOELPEN 4
```

GPIOE Peripheral Clock in Low Power Mode Enable

5.87.2.13 RCC_AHB1LPENR_GPIOFLPEN

```
#define RCC_AHB1LPENR_GPIOFLPEN 5
```

GPIOF Peripheral Clock in Low Power Mode Enable

5.87.2.14 RCC_AHB1LPENR_GPIOGLPEN

```
#define RCC_AHB1LPENR_GPIOGLPEN 6
```

GPIOG Peripheral Clock in Low Power Mode Enable

5.87.2.15 RCC_AHB1LPENR_GPIOHLPEN

```
#define RCC_AHB1LPENR_GPIOHLPEN 7
```

GPIOH Peripheral Clock in Low Power Mode Enable

5.87.2.16 RCC_AHB1LPENR_GPIOILPEN

```
#define RCC_AHB1LPENR_GPIOILPEN 8
```

GPIOI Peripheral Clock in Low Power Mode Enable

5.87.2.17 RCC_AHB1LPENR_OTGHSHULPI

```
#define RCC_AHB1LPENR_OTGHSHULPI 30
```

USB OTG HS ULPI Peripheral Clock in Low Power Mode Enable

5.87.2.18 RCC_AHB1LPENR_OTGHSLPEN

```
#define RCC_AHB1LPENR_OTGHSLPEN 29
```

USB OTG HS Peripheral Clock in Low Power Mode Enable

5.88 RCC_AHB2LPENR Bit Position Definitions

Bit position definitions for RCC_AHB2LPENR register.

Collaboration diagram for RCC_AHB2LPENR Bit Position Definitions:



Macros

- #define RCC_AHB2LPENR_DCMILPEN 0
- #define RCC_AHB2LPENR_CRYPLPEN 4
- #define RCC_AHB2LPENR_HASHLPEN 5
- #define RCC_AHB2LPENR_RNGLPEN 6
- #define RCC_AHB2LPENR_OTGFSLPEN 7

5.88.1 Detailed Description

Bit position definitions for RCC_AHB2LPENR register.

5.88.2 Macro Definition Documentation

5.88.2.1 RCC_AHB2LPENR_CRYPLPEN

```
#define RCC_AHB2LPENR_CRYPLPEN 4
```

CRYP Peripheral Clock in Low Power Mode Enable

5.88.2.2 RCC_AHB2LPENR_DCMILPEN

```
#define RCC_AHB2LPENR_DCMILPEN 0
```

DCMI Peripheral Clock in Low Power Mode Enable

5.88.2.3 RCC_AHB2LPENR_HASHLPEN

```
#define RCC_AHB2LPENR_HASHLPEN 5
```

HASH Peripheral Clock in Low Power Mode Enable

5.88.2.4 RCC_AHB2LPENR_OTGFSLPEN

```
#define RCC_AHB2LPENR_OTGFSLPEN 7
```

USB OTG FS Peripheral Clock in Low Power Mode Enable

5.88.2.5 RCC_AHB2LPENR_RNGLPEN

```
#define RCC_AHB2LPENR_RNGLPEN 6
```

RNG Peripheral Clock in Low Power Mode Enable

5.89 RCC_AHB3LPENR Bit Position Definitions

Bit position definitions for RCC_AHB3LPENR register.

Collaboration diagram for RCC_AHB3LPENR Bit Position Definitions:



Macros

- #define RCC_AHB3LPENR_FSMCLPEN 0

5.89.1 Detailed Description

Bit position definitions for RCC_AHB3LPENR register.

5.89.2 Macro Definition Documentation

5.89.2.1 RCC_AHB3LPENR_FSMCLPEN

```
#define RCC_AHB3LPENR_FSMCLPEN 0
```

FSMC Peripheral Clock in Low Power Mode Enable

5.90 RCC_APB1LPENR Bit Position Definitions

Bit position definitions for RCC_APB1LPENR register.

Collaboration diagram for RCC_APB1LPENR Bit Position Definitions:



Macros

- #define RCC_APB1LPENR_TIM2LPEN 0
- #define RCC_APB1LPENR_TIM3LPEN 1
- #define RCC_APB1LPENR_TIM4LPEN 2
- #define RCC_APB1LPENR_TIM5LPEN 3
- #define RCC_APB1LPENR_TIM6LPEN 4
- #define RCC_APB1LPENR_TIM7LPEN 5
- #define RCC_APB1LPENR_TIM12LPEN 6
- #define RCC_APB1LPENR_TIM13LPEN 7
- #define RCC_APB1LPENR_TIM14LPEN 8
- #define RCC_APB1LPENR_WWDGLPEN 11
- #define RCC_APB1LPENR_SPI2LPEN 14
- #define RCC_APB1LPENR_SPI3LPEN 15
- #define RCC_APB1LPENR_USART2LPEN 17
- #define RCC_APB1LPENR_USART3LPEN 18
- #define RCC_APB1LPENR_UART4LPEN 19
- #define RCC_APB1LPENR_UART5LPEN 20
- #define RCC_APB1LPENR_I2C1LPEN 21
- #define RCC_APB1LPENR_I2C2LPEN 22
- #define RCC_APB1LPENR_I2C3LPEN 23
- #define RCC_APB1LPENR_CAN1LPEN 25
- #define RCC_APB1LPENR_CAN2LPEN 26
- #define RCC_APB1LPENR_PWRLPEN 28
- #define RCC_APB1LPENR_DACLPEN 29
- #define RCC_APB1LPENR_UART7LPEN 30
- #define RCC_APB1LPENR_UART8LPEN 31

5.90.1 Detailed Description

Bit position definitions for RCC_APB1LPENR register.

5.90.2 Macro Definition Documentation

5.90.2.1 RCC_APB1LPENR_CAN1LPEN

```
#define RCC_APB1LPENR_CAN1LPEN 25
```

CAN1 Peripheral Clock in Low Power Mode Enable

5.90.2.2 RCC_APB1LPENR_CAN2LPEN

```
#define RCC_APB1LPENR_CAN2LPEN 26
```

CAN2 Peripheral Clock in Low Power Mode Enable

5.90.2.3 RCC_APB1LPENR_DACLPEN

```
#define RCC_APB1LPENR_DACLPEN 29
```

DAC Peripheral Clock in Low Power Mode Enable

5.90.2.4 RCC_APB1LPENR_I2C1LPEN

```
#define RCC_APB1LPENR_I2C1LPEN 21
```

I2C1 Peripheral Clock in Low Power Mode Enable

5.90.2.5 RCC_APB1LPENR_I2C2LPEN

```
#define RCC_APB1LPENR_I2C2LPEN 22
```

I2C2 Peripheral Clock in Low Power Mode Enable

5.90.2.6 RCC_APB1LPENR_I2C3LPEN

```
#define RCC_APB1LPENR_I2C3LPEN 23
```

I2C3 Peripheral Clock in Low Power Mode Enable

5.90.2.7 RCC_APB1LPENR_PWRLPEN

```
#define RCC_APB1LPENR_PWRLPEN 28
```

Power Interface Peripheral Clock in Low Power Mode Enable

5.90.2.8 RCC_APB1LPENR_SPI2LPEN

```
#define RCC_APB1LPENR_SPI2LPEN 14
```

SPI2 Peripheral Clock in Low Power Mode Enable

5.90.2.9 RCC_APB1LPENR_SPI3LPEN

```
#define RCC_APB1LPENR_SPI3LPEN 15
```

SPI3 Peripheral Clock in Low Power Mode Enable

5.90.2.10 RCC_APB1LPENR_TIM12LPEN

```
#define RCC_APB1LPENR_TIM12LPEN 6
```

TIM12 Peripheral Clock in Low Power Mode Enable

5.90.2.11 RCC_APB1LPENR_TIM13LPEN

```
#define RCC_APB1LPENR_TIM13LPEN 7
```

TIM13 Peripheral Clock in Low Power Mode Enable

5.90.2.12 RCC_APB1LPENR_TIM14LPEN

```
#define RCC_APB1LPENR_TIM14LPEN 8
```

TIM14 Peripheral Clock in Low Power Mode Enable

5.90.2.13 RCC_APB1LPENR_TIM2LPEN

```
#define RCC_APB1LPENR_TIM2LPEN 0
```

TIM2 Peripheral Clock in Low Power Mode Enable

5.90.2.14 RCC_APB1LPENR_TIM3LPEN

```
#define RCC_APB1LPENR_TIM3LPEN 1
```

TIM3 Peripheral Clock in Low Power Mode Enable

5.90.2.15 RCC_APB1LPENR_TIM4LPEN

```
#define RCC_APB1LPENR_TIM4LPEN 2
```

TIM4 Peripheral Clock in Low Power Mode Enable

5.90.2.16 RCC_APB1LPENR_TIM5LPEN

```
#define RCC_APB1LPENR_TIM5LPEN 3
```

TIM5 Peripheral Clock in Low Power Mode Enable

5.90.2.17 RCC_APB1LPENR_TIM6LPEN

```
#define RCC_APB1LPENR_TIM6LPEN 4
```

TIM6 Peripheral Clock in Low Power Mode Enable

5.90.2.18 RCC_APB1LPENR_TIM7LPEN

```
#define RCC_APB1LPENR_TIM7LPEN 5
```

TIM7 Peripheral Clock in Low Power Mode Enable

5.90.2.19 RCC_APB1LPENR_UART4LPEN

```
#define RCC_APB1LPENR_UART4LPEN 19
```

UART4 Peripheral Clock in Low Power Mode Enable

5.90.2.20 RCC_APB1LPENR_UART5LPEN

```
#define RCC_APB1LPENR_UART5LPEN 20
```

UART5 Peripheral Clock in Low Power Mode Enable

5.90.2.21 RCC_APB1LPENR_UART7LPEN

```
#define RCC_APB1LPENR_UART7LPEN 30
```

UART7 Peripheral Clock in Low Power Mode Enable

5.90.2.22 RCC_APB1LPENR_UART8LPEN

```
#define RCC_APB1LPENR_UART8LPEN 31
```

UART8 Peripheral Clock in Low Power Mode Enable

5.90.2.23 RCC_APB1LPENR_USART2LPEN

```
#define RCC_APB1LPENR_USART2LPEN 17
```

USART2 Peripheral Clock in Low Power Mode Enable

5.90.2.24 RCC_APB1LPENR_USART3LPEN

```
#define RCC_APB1LPENR_USART3LPEN 18
```

USART3 Peripheral Clock in Low Power Mode Enable

5.90.2.25 RCC_APB1LPENR_WWDGLPEN

```
#define RCC_APB1LPENR_WWDGLPEN 11
```

WWDG Peripheral Clock in Low Power Mode Enable

5.91 RCC_APB2LPENR Bit Position Definitions

Bit position definitions for RCC_APB2LPENR register.

Collaboration diagram for RCC_APB2LPENR Bit Position Definitions:



Macros

- #define RCC_APB2LPENR_TIM1LPEN 0
- #define RCC_APB2LPENR_TIM8LPEN 1
- #define RCC_APB2LPENR_USART1LPEN 4
- #define RCC_APB2LPENR_USART6LPEN 5
- #define RCC_APB2LPENR_ADCLPEN 8
- #define RCC_APB2LPENR_SDIOLPEN 11
- #define RCC_APB2LPENR_SPI1LPEN 12
- #define RCC_APB2LPENR_SYSCFGLPEN 14
- #define RCC_APB2LPENR_TIM9LPEN 16
- #define RCC_APB2LPENR_TIM10LPEN 17
- #define RCC_APB2LPENR_TIM11LPEN 18

5.91.1 Detailed Description

Bit position definitions for RCC_APB2LPENR register.

5.91.2 Macro Definition Documentation

5.91.2.1 RCC_APB2LPENR_ADCLPEN

```
#define RCC_APB2LPENR_ADCLPEN 8
```

ADC Peripheral Clock in Low Power Mode Enable

5.91.2.2 RCC_APB2LPENR_SDIOLPEN

```
#define RCC_APB2LPENR_SDIOLPEN 11
```

SDIO Peripheral Clock in Low Power Mode Enable

5.91.2.3 RCC_APB2LPENR_SPI1LPEN

```
#define RCC_APB2LPENR_SPI1LPEN 12
```

SPI1 Peripheral Clock in Low Power Mode Enable

5.91.2.4 RCC_APB2LPENR_SYSCFGLPEN

```
#define RCC_APB2LPENR_SYSCFGLPEN 14
```

System Configuration Controller Peripheral Clock in Low Power Mode Enable

5.91.2.5 RCC_APB2LPENR_TIM10LPEN

```
#define RCC_APB2LPENR_TIM10LPEN 17
```

TIM10 Peripheral Clock in Low Power Mode Enable

5.91.2.6 RCC_APB2LPENR_TIM11LPEN

```
#define RCC_APB2LPENR_TIM11LPEN 18
```

TIM11 Peripheral Clock in Low Power Mode Enable

5.91.2.7 RCC_APB2LPENR_TIM1LPEN

```
#define RCC_APB2LPENR_TIM1LPEN 0
```

TIM1 Peripheral Clock in Low Power Mode Enable

5.91.2.8 RCC_APB2LPENR_TIM8LPEN

```
#define RCC_APB2LPENR_TIM8LPEN 1
```

TIM8 Peripheral Clock in Low Power Mode Enable

5.91.2.9 RCC_APB2LPENR_TIM9LPEN

```
#define RCC_APB2LPENR_TIM9LPEN 16
```

TIM9 Peripheral Clock in Low Power Mode Enable

5.91.2.10 RCC_APB2LPENR_USART1LPEN

```
#define RCC_APB2LPENR_USART1LPEN 4
```

USART1 Peripheral Clock in Low Power Mode Enable

5.91.2.11 RCC_APB2LPENR_USART6LPEN

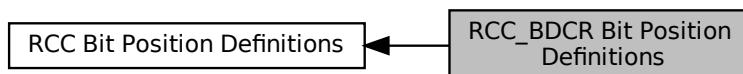
```
#define RCC_APB2LPENR_USART6LPEN 5
```

USART6 Peripheral Clock in Low Power Mode Enable

5.92 RCC_BDCR Bit Position Definitions

Bit position definitions for RCC_BDCR register.

Collaboration diagram for RCC_BDCR Bit Position Definitions:



Macros

- #define RCC_BDCR_LSEON 0
- #define RCC_BDCR_LSERDY 1
- #define RCC_BDCR_LSEBYP 2
- #define RCC_BDCR_RTCSEL 8
- #define RCC_BDCR_RTCEN 15
- #define RCC_BDCR_BDRST 16

5.92.1 Detailed Description

Bit position definitions for RCC_BDCR register.

5.92.2 Macro Definition Documentation

5.92.2.1 RCC_BDCR_BDRST

```
#define RCC_BDCR_BDRST 16
```

Backup Domain Software Reset

5.92.2.2 RCC_BDCR_LSEBYP

```
#define RCC_BDCR_LSEBYP 2
```

External Low-Speed Oscillator Bypass

5.92.2.3 RCC_BDCR_LSEON

```
#define RCC_BDCR_LSEON 0
```

External Low-Speed Oscillator Enable

5.92.2.4 RCC_BDCR_LSERDY

```
#define RCC_BDCR_LSERDY 1
```

External Low-Speed Oscillator Ready

5.92.2.5 RCC_BDCR_RTCEN

```
#define RCC_BDCR_RTCEN 15
```

RTC Clock Enable

5.92.2.6 RCC_BDCR_RTCSEL

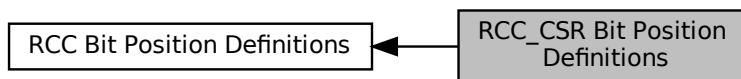
```
#define RCC_BDCR_RTCSEL 8
```

RTC Clock Source Selection

5.93 RCC_CSR Bit Position Definitions

Bit position definitions for RCC_CSR register.

Collaboration diagram for RCC_CSR Bit Position Definitions:



Macros

- #define RCC_CSR_LSION 0
- #define RCC_CSR_LSIRDY 1
- #define RCC_CSR_RMVF 24
- #define RCC_CSR_OBLRSTF 25
- #define RCC_CSR_PINRSTF 26
- #define RCC_CSR_PORRSTF 27
- #define RCC_CSR_SFTRSTF 28
- #define RCC_CSR_IWDGRSTF 29
- #define RCC_CSR_WWDGRSTF 30
- #define RCC_CSR_LPWRRSTF 31

5.93.1 Detailed Description

Bit position definitions for RCC_CSR register.

5.93.2 Macro Definition Documentation

5.93.2.1 RCC_CSR_IWDGRSTF

```
#define RCC_CSR_IWDGRSTF 29
```

Independent Watchdog Reset Flag

5.93.2.2 RCC_CSR_LPWRRSTF

```
#define RCC_CSR_LPWRRSTF 31
```

Low-Power Reset Flag

5.93.2.3 RCC_CSR_LSION

```
#define RCC_CSR_LSION 0
```

Internal Low-Speed Oscillator Enable

5.93.2.4 RCC_CSR_LSIRDY

```
#define RCC_CSR_LSIRDY 1
```

Internal Low-Speed Oscillator Ready

5.93.2.5 RCC_CSR_OBLRSTF

```
#define RCC_CSR_OBLRSTF 25
```

Option Byte Loader Reset Flag

5.93.2.6 RCC_CSR_PINRSTF

```
#define RCC_CSR_PINRSTF 26
```

PIN Reset Flag

5.93.2.7 RCC_CSR_PORRSTF

```
#define RCC_CSR_PORRSTF 27
```

POR/PDR Reset Flag

5.93.2.8 RCC_CSR_RMVF

```
#define RCC_CSR_RMVF 24
```

Remove Reset Flag

5.93.2.9 RCC_CSR_SFTRSTF

```
#define RCC_CSR_SFTRSTF 28
```

Software Reset Flag

5.93.2.10 RCC_CSR_WWDGRSTF

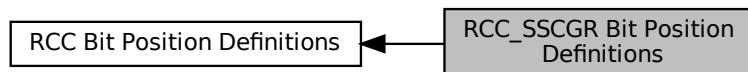
```
#define RCC_CSR_WWDGRSTF 30
```

Window Watchdog Reset Flag

5.94 RCC_SSCGR Bit Position Definitions

Bit position definitions for RCC_SSCGR register.

Collaboration diagram for RCC_SSCGR Bit Position Definitions:



Macros

- #define RCC_SSCGR_MODPER 0
- #define RCC_SSCGR_INCSTEP 13
- #define RCC_SSCGR_SPREADSEL 15
- #define RCC_SSCGR_SSCGEN 31

5.94.1 Detailed Description

Bit position definitions for RCC_SSCGR register.

5.94.2 Macro Definition Documentation

5.94.2.1 RCC_SSCGR_INCSTEP

```
#define RCC_SSCGR_INCSTEP 13
```

Increase Step

5.94.2.2 RCC_SSCGR_MODPER

```
#define RCC_SSCGR_MODPER 0
```

Modulation Period

5.94.2.3 RCC_SSCGR_SPREADSEL

```
#define RCC_SSCGR_SPREADSEL 15
```

Spread Select

5.94.2.4 RCC_SSCGR_SSCEGEN

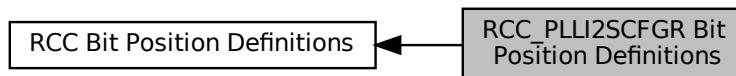
```
#define RCC_SSCGR_SSCEGEN 31
```

Spread Spectrum Clock Generation Enable

5.95 RCC_PLLI2SCFGR Bit Position Definitions

Bit position definitions for RCC_PLLI2SCFGR register.

Collaboration diagram for RCC_PLLI2SCFGR Bit Position Definitions:



Macros

- #define RCC_PLLI2SCFGR_PLLI2SN 6
- #define RCC_PLLI2SCFGR_PLLI2SR 28

5.95.1 Detailed Description

Bit position definitions for RCC_PLLI2SCFGR register.

5.95.2 Macro Definition Documentation

5.95.2.1 RCC_PLLI2SCFGR_PLLI2SN

```
#define RCC_PLLI2SCFGR_PLLI2SN 6
```

PLL2S N Factor

5.95.2.2 RCC_PLLI2SCFGR_PLLI2SR

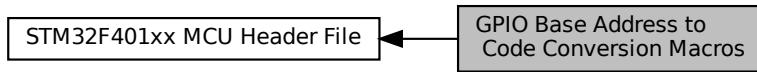
```
#define RCC_PLLI2SCFGR_PLLI2SR 28
```

PLL2S R Factor

5.96 GPIO Base Address to Code Conversion Macros

Macros for converting GPIO base addresses to corresponding port codes.

Collaboration diagram for GPIO Base Address to Code Conversion Macros:



Macros

- `#define GPIO_BASEADDR_TO_CODE(x)`
Macro to convert GPIO base address to port code.

5.96.1 Detailed Description

Macros for converting GPIO base addresses to corresponding port codes.

5.96.2 Macro Definition Documentation

5.96.2.1 GPIO_BASEADDR_TO_CODE

```
#define GPIO_BASEADDR_TO_CODE(
    x )
```

Value:

```
(     (x == GPIOA)?0:\\
(x == GPIOB)?1:\\
(x == GPIOC)?2:\\
(x == GPIOD)?3:\\
(x == GPIOE)?4:0 )
```

Macro to convert GPIO base address to port code.

Parameters

x	GPIOx base address.
---	---------------------

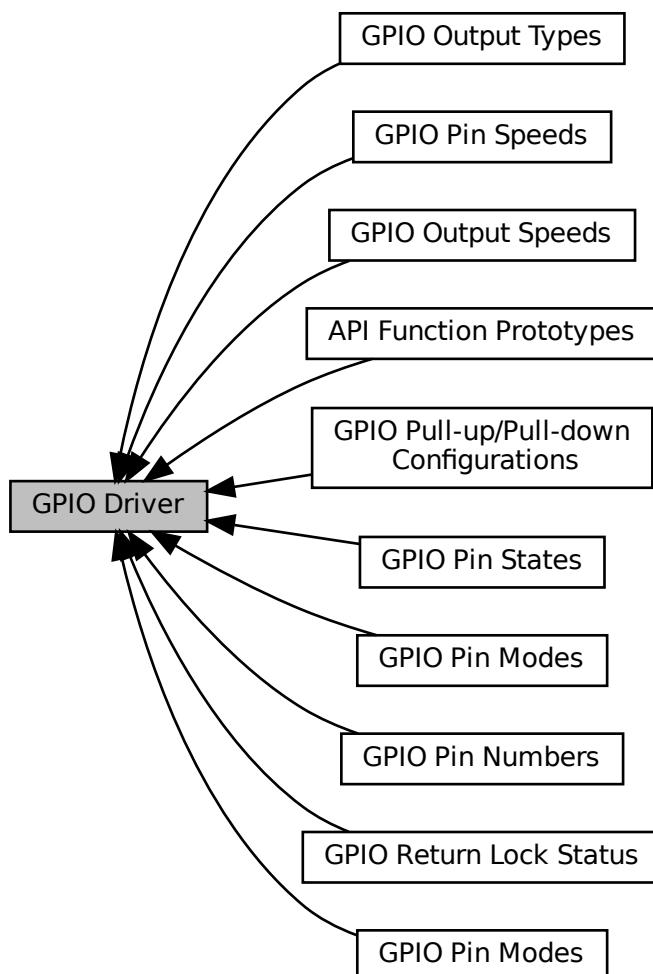
Returns

Corresponding port code for the given GPIO base address.

5.97 GPIO Driver

GPIO driver APIs for STM32F401xx MCU.

Collaboration diagram for GPIO Driver:



Modules

- [GPIO Pin Numbers](#)
Defines the GPIO pin numbers.
- [GPIO Pin Modes](#)
Defines the GPIO pin modes.
- [GPIO Pin Speeds](#)
Defines the GPIO pin speeds.
- [GPIO Pin States](#)
Defines the GPIO pin states.
- [GPIO Return Lock Status](#)
Defines the return status for GPIO lock operations.
- [GPIO Pin Modes](#)
Defines possible GPIO pin modes.
- [GPIO Output Speeds](#)
Defines possible GPIO pin output speeds.
- [GPIO Pull-up/Pull-down Configurations](#)
Defines possible GPIO pin pull-up and pull-down configurations.
- [GPIO Output Types](#)
Defines possible GPIO pin output types.
- [API Function Prototypes](#)

Classes

- struct [GPIO_PinConfig_t](#)
Configuration structure for GPIO pins.

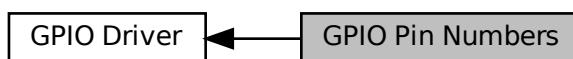
5.97.1 Detailed Description

GPIO driver APIs for STM32F401xx MCU.

5.98 GPIO Pin Numbers

Defines the GPIO pin numbers.

Collaboration diagram for GPIO Pin Numbers:



Macros

- #define **GPIO_PIN_0** 0
- #define **GPIO_PIN_1** 1
- #define **GPIO_PIN_2** 2
- #define **GPIO_PIN_3** 3
- #define **GPIO_PIN_4** 4
- #define **GPIO_PIN_5** 5
- #define **GPIO_PIN_6** 6
- #define **GPIO_PIN_7** 7
- #define **GPIO_PIN_8** 8
- #define **GPIO_PIN_9** 9
- #define **GPIO_PIN_10** 10
- #define **GPIO_PIN_11** 11
- #define **GPIO_PIN_12** 12
- #define **GPIO_PIN_13** 13
- #define **GPIO_PIN_14** 14
- #define **GPIO_PIN_15** 15
- #define **GPIO_PIN_ALL** ((**uint16**)0xFFFF)

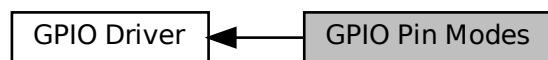
5.98.1 Detailed Description

Defines the GPIO pin numbers.

5.99 GPIO Pin Modes

Defines the GPIO pin modes.

Collaboration diagram for GPIO Pin Modes:



Macros

- #define **GPIO_MODE_INPUT_FLO** 0x00000001U
- #define **GPIO_MODE_INPUT_PU** 0x00000002U
- #define **GPIO_MODE_INPUT_PD** 0x00000003U
- #define **GPIO_MODE_OUTPUT_PP** 0x00000004U
- #define **GPIO_MODE_OUTPUT_OD** 0x00000005U
- #define **GPIO_MODE_OUTPUT_AF_PP** 0x00000006U
- #define **GPIO_MODE_OUTPUT_AF_OD** 0x00000007U
- #define **GPIO_MODE_AF_INPUT** 0x00000008U

5.99.1 Detailed Description

Defines the GPIO pin modes.

5.99.2 Macro Definition Documentation

5.99.2.1 `GPIO_MODE_AF_INPUT`

```
#define GPIO_MODE_AF_INPUT 0x00000008U
```

Alternate function input

5.99.2.2 `GPIO_MODE_INPUT_FLO`

```
#define GPIO_MODE_INPUT_FLO 0x00000001U
```

Floating input

5.99.2.3 `GPIO_MODE_INPUT_PD`

```
#define GPIO_MODE_INPUT_PD 0x00000003U
```

Input with pull-down

5.99.2.4 `GPIO_MODE_INPUT_PU`

```
#define GPIO_MODE_INPUT_PU 0x00000002U
```

Input with pull-up

5.99.2.5 `GPIO_MODE_OUTPUT_AF_OD`

```
#define GPIO_MODE_OUTPUT_AF_OD 0x00000007U
```

Alternate function output open-drain

5.99.2.6 `GPIO_MODE_OUTPUT_AF_PP`

```
#define GPIO_MODE_OUTPUT_AF_PP 0x00000006U
```

Alternate function output push-pull

5.99.2.7 GPIO_MODE_OUTPUT_OD

```
#define GPIO_MODE_OUTPUT_OD 0x00000005U
```

General purpose output open-drain

5.99.2.8 GPIO_MODE_OUTPUT_PP

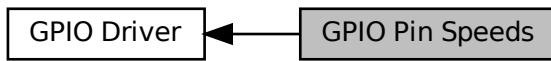
```
#define GPIO_MODE_OUTPUT_PP 0x00000004U
```

General purpose output push-pull

5.100 GPIO Pin Speeds

Defines the GPIO pin speeds.

Collaboration diagram for GPIO Pin Speeds:



Macros

- #define GPIO_SPEED_10M 0x00000001U
- #define GPIO_SPEED_2M 0x00000002U
- #define GPIO_SPEED_50M 0x00000003U

5.100.1 Detailed Description

Defines the GPIO pin speeds.

5.100.2 Macro Definition Documentation

5.100.2.1 GPIO_SPEED_10M

```
#define GPIO_SPEED_10M 0x00000001U
```

Output mode, max speed 10 MHz

5.100.2.2 GPIO_SPEED_2M

```
#define GPIO_SPEED_2M 0x00000002U
```

Output mode, max speed 2 MHz

5.100.2.3 GPIO_SPEED_50M

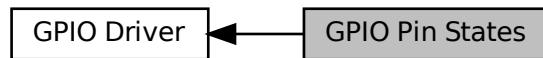
```
#define GPIO_SPEED_50M 0x00000003U
```

Output mode, max speed 50 MHz

5.101 GPIO Pin States

Defines the GPIO pin states.

Collaboration diagram for GPIO Pin States:



Macros

- #define GPIO_PIN_SET 1
- #define GPIO_PIN_RESET 0

5.101.1 Detailed Description

Defines the GPIO pin states.

5.101.2 Macro Definition Documentation

5.101.2.1 GPIO_PIN_RESET

```
#define GPIO_PIN_RESET 0
```

GPIO Pin reset state

5.101.2.2 GPIO_PIN_SET

```
#define GPIO_PIN_SET 1
```

GPIO Pin set state

5.102 GPIO Return Lock Status

Defines the return status for GPIO lock operations.

Collaboration diagram for GPIO Return Lock Status:



Macros

- #define GPIO_RETURN_LOCK_OK 1
- #define GPIO_RETURN_LOCK_ERROR 0

5.102.1 Detailed Description

Defines the return status for GPIO lock operations.

5.102.2 Macro Definition Documentation

5.102.2.1 GPIO_RETURN_LOCK_ERROR

```
#define GPIO_RETURN_LOCK_ERROR 0
```

GPIO pin configuration lock failed

5.102.2.2 GPIO_RETURN_LOCK_OK

```
#define GPIO_RETURN_LOCK_OK 1
```

GPIO pin configuration is locked successfully

5.103 GPIO Pin Modes

Defines possible GPIO pin modes.

Collaboration diagram for GPIO Pin Modes:



Macros

- #define GPIO_MODE_IN 0
- #define GPIO_MODE_OUT 1
- #define GPIO_MODE_ALTFN 2
- #define GPIO_MODE_ANALOG 3
- #define GPIO_MODE_IT_FT 4
- #define GPIO_MODE_IT_RT 5
- #define GPIO_MODE_IT_RFT 6

5.103.1 Detailed Description

Defines possible GPIO pin modes.

5.103.2 Macro Definition Documentation

5.103.2.1 GPIO_MODE_ALTFN

```
#define GPIO_MODE_ALTFN 2
```

GPIO Alternate Function mode

5.103.2.2 GPIO_MODE_ANALOG

```
#define GPIO_MODE_ANALOG 3
```

GPIO Analog mode

5.103.2.3 GPIO_MODE_IN

```
#define GPIO_MODE_IN 0
```

GPIO Input mode

5.103.2.4 GPIO_MODE_IT_FT

```
#define GPIO_MODE_IT_FT 4
```

GPIO Interrupt Falling-Edge Trigger mode

5.103.2.5 GPIO_MODE_IT_RFT

```
#define GPIO_MODE_IT_RFT 6
```

GPIO Interrupt Rising-Falling Edge Trigger mode

5.103.2.6 GPIO_MODE_IT_RT

```
#define GPIO_MODE_IT_RT 5
```

GPIO Interrupt Rising-Edge Trigger mode

5.103.2.7 GPIO_MODE_OUT

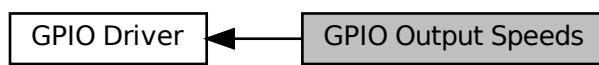
```
#define GPIO_MODE_OUT 1
```

GPIO Output mode

5.104 GPIO Output Speeds

Defines possible GPIO pin output speeds.

Collaboration diagram for GPIO Output Speeds:



Macros

- #define GPIO_SPEED_LOW 0
- #define GPIO_SPEED_MEDIUM 1
- #define GPIO_SPEED_FAST 2
- #define GPIO_SPEED_HIGH 3

5.104.1 Detailed Description

Defines possible GPIO pin output speeds.

5.104.2 Macro Definition Documentation

5.104.2.1 GPIO_SPEED_FAST

```
#define GPIO_SPEED_FAST 2
```

GPIO Output speed Fast

5.104.2.2 GPIO_SPEED_HIGH

```
#define GPIO_SPEED_HIGH 3
```

GPIO Output speed High

5.104.2.3 GPIO_SPEED_LOW

```
#define GPIO_SPEED_LOW 0
```

GPIO Output speed Low

5.104.2.4 GPIO_SPEED_MEDIUM

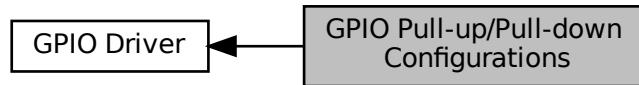
```
#define GPIO_SPEED_MEDIUM 1
```

GPIO Output speed Medium

5.105 GPIO Pull-up/Pull-down Configurations

Defines possible GPIO pin pull-up and pull-down configurations.

Collaboration diagram for GPIO Pull-up/Pull-down Configurations:



Macros

- `#define GPIO_NO_PUPD 0`
- `#define GPIO_PIN_PU 1`
- `#define GPIO_PIN_PD 2`

5.105.1 Detailed Description

Defines possible GPIO pin pull-up and pull-down configurations.

5.105.2 Macro Definition Documentation

5.105.2.1 GPIO_NO_PUPD

```
#define GPIO_NO_PUPD 0
```

No pull-up/pull-down configuration

5.105.2.2 GPIO_PIN_PD

```
#define GPIO_PIN_PD 2
```

GPIO Pull-down configuration

5.105.2.3 GPIO_PIN_PU

```
#define GPIO_PIN_PU 1
```

GPIO Pull-up configuration

5.106 GPIO Output Types

Defines possible GPIO pin output types.

Collaboration diagram for GPIO Output Types:



Macros

- `#define GPIO_OP_TYPE_PP 0`
- `#define GPIO_OP_TYPE_OD 1`

5.106.1 Detailed Description

Defines possible GPIO pin output types.

5.106.2 Macro Definition Documentation

5.106.2.1 GPIO_OP_TYPE_OD

```
#define GPIO_OP_TYPE_OD 1
```

GPIO Output type Open-Drain mode

5.106.2.2 GPIO_OP_TYPE_PP

```
#define GPIO_OP_TYPE_PP 0
```

GPIO Output type Push-Pull mode

5.107 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void **MCAL_GPIO_Init** (**GPIO_TypeDef** *GPIOx, **GPIO_PinConfig_t** *PinConfig)
Initializes the specified GPIO pin according to the provided configuration.
- void **MCAL_GPIO_DeInit** (**GPIO_TypeDef** *GPIOx)
Resets the specified GPIO port.
- **uint8 MCAL_GPIO_ReadPin** (**GPIO_TypeDef** *GPIOx, **uint16** PinNumber)
Reads the value of a specified GPIO pin.
- **uint16 MCAL_GPIO_ReadPort** (**GPIO_TypeDef** *GPIOx)
Reads the value of the entire GPIO port.
- void **MCAL_GPIO_WritePin** (**GPIO_TypeDef** *GPIOx, **uint16** PinNumber, **uint8** Value)
Writes a value to a specified GPIO pin.
- void **MCAL_GPIO_WritePort** (**GPIO_TypeDef** *GPIOx, **uint16** Value)
Writes a value to the entire GPIO port.
- void **MCAL_GPIO_TogglePin** (**GPIO_TypeDef** *GPIOx, **uint16** PinNumber)
Toggles the state of a specified GPIO pin.
- void **MCAL_GPIO_TogglePort** (**GPIO_TypeDef** *GPIOx)
Toggles the value of a specific GPIO Port.
- **uint8 MCAL_GPIO_LockPin** (**GPIO_TypeDef** *GPIOx, **uint16** PinNumber)
Locks the configuration of a specific GPIO pin to prevent further changes.

5.107.1 Detailed Description

5.107.2 Function Documentation

5.107.2.1 MCAL_GPIO_DeInit()

```
void MCAL_GPIO_DeInit (
    GPIO_TypeDef * GPIOx )
```

Resets the specified GPIO port.

This function resets the GPIO port, clearing all configuration settings.

Parameters

in	<i>GPIOx</i>	Pointer to the GPIO peripheral base address.
----	--------------	--

Return values

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

Note

This function does not affect other GPIO ports.

This function resets the GPIO port by toggling the appropriate reset bit in the RCC_AHB1RSTR register.

Parameters

in	<i>GPIOx</i>	The GPIO peripheral to be reset (e.g., GPIOA, GPIOB, GPIOC, etc.).
----	--------------	--

Returns

None

Note

None

5.107.2.2 MCAL_GPIO_Init()

```
void MCAL_GPIO_Init (
    GPIO_TypeDef * GPIOx,
    GPIO_PinConfig_t * PinConfig )
```

Initializes the specified GPIO pin according to the provided configuration.

This function configures the GPIO pin based on the parameters specified in the [GPIO_PinConfig_t](#) structure. It is essential to enable the RCC clock for the corresponding GPIO port before calling this function.

Parameters

in	<i>GPIOx</i>	Pointer to the GPIO peripheral base address.
in	<i>PinConfig</i>	Pointer to a GPIO_PinConfig_t structure containing the pin configuration.

Return values

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

Note

This function configures the GPIO pins and their associated settings.

This function sets up the GPIO pin mode, output type, speed, pull-up/pull-down resistors, and alternate function if needed.

Parameters

in	<i>GPIOx</i>	The GPIO peripheral to be initialized (e.g., GPIOA, GPIOB, GPIOC, etc.).
in	<i>PinConfig</i>	Pointer to a GPIO_PinConfig_t structure containing the configuration details for the pin.

Returns

None

Note

It is mandatory to enable the RCC clock for the corresponding GPIO port before initialization.

5.107.2.3 MCAL_GPIO_LockPin()

```
uint8 MCAL_GPIO_LockPin (
    GPIO_TypeDef * GPIOx,
    uint16 PinNumber )
```

Locks the configuration of a specific GPIO pin to prevent further changes.

This function locks the configuration of the specified GPIO pin. Once locked, the pin's configuration cannot be modified until the reset of the device. This is useful for protecting the pin configuration from accidental changes.

Parameters

in	<i>GPIOx</i>	Pointer to the GPIO peripheral base address. This parameter can be one of the following values depending on the device used: GPIOA, GPIOB, GPIOC, GPIOD, GPIOE.
in	<i>PinNumber</i>	The number of the pin to lock. This parameter should be a value defined in GPIO Pin Numbers .

Return values

<i>uint8</i>	Returns GPIO_RETURN_LOCK_OK if the pin configuration is successfully locked, or GPIO_RETURN_LOCK_ERROR if the locking operation fails.
--------------	--

Note

Ensure that the pin's configuration is correctly set before calling this function, as the configuration cannot be changed once locked.

Locks the configuration of a specific GPIO pin to prevent further changes.

This function implements a locking mechanism to prevent further changes to the configuration of a specified GPIO pin. The locking sequence must be followed to successfully lock the pin configuration.

Parameters

in	<i>GPIOx</i>	The GPIO peripheral to be used (e.g., GPIOA, GPIOB, GPIOC, etc.).
in	<i>PinNumber</i>	The pin number to be locked, specified according to GPIO Pin Numbers .

Return values

<i>uint8</i>	Returns <code>GPIO_RETURN_LOCK_OK</code> if the pin configuration is successfully locked, or <code>GPIO_RETURN_LOCK_ERROR</code> if the locking process fails.
--------------	--

Note

The locking sequence is as follows:

- Write 1 to the lock key bit.
- Write 0 to the lock key bit.
- Write 1 to the lock key bit.
- Read the lock key bit (should be 0).
- Optionally, read the lock key bit again (should be 1 to confirm the lock).

The value of the lock bits (LCK[15:0]) must not change during this sequence, otherwise, the lock process will be aborted.

5.107.2.4 MCAL_GPIO_ReadPin()

```
uint8 MCAL_GPIO_ReadPin (
    GPIO_TypeDef * GPIOx,
    uint16 PinNumber )
```

Reads the value of a specified GPIO pin.

This function retrieves the current state of the specified GPIO pin.

Parameters

in	<i>GPIOx</i>	Pointer to the GPIO peripheral base address.
in	<i>PinNumber</i>	The number of the pin to read. This parameter should be a value defined in GPIO Pin Numbers .

Return values

<i>uint8</i>	The state of the GPIO pin. This value can be one of the values defined in GPIO Pin States .
--------------	---

Note

This function reads the input value of the pin.

Reads the value of a specified GPIO pin.

This function reads the input data register to retrieve the value of the specified GPIO pin.

Parameters

in	<i>GPIOx</i>	The GPIO peripheral from which to read (e.g., GPIOA, GPIOB, GPIOC, etc.).
in	<i>PinNumber</i>	The pin number to be read (e.g., GPIO_PIN_0, GPIO_PIN_1, etc.).

Returns

The value of the specified pin (0 or 1).

Note

None

5.107.2.5 MCAL_GPIO_ReadPort()

```
uint16 MCAL_GPIO_ReadPort (
    GPIO_TypeDef * GPIOx )
```

Reads the value of the entire GPIO port.

This function retrieves the current state of all the pins in the specified GPIO port.

Parameters

in	<i>GPIOx</i>	Pointer to the GPIO peripheral base address.
----	--------------	--

Return values

<i>uint16</i>	The state of the entire GPIO port.
---------------	------------------------------------

Note

This function reads the input values of all pins in the port.

Reads the value of the entire GPIO port.

This function reads the input data register to retrieve the value of all the pins in the specified GPIO port.

Parameters

in	<i>GPIOx</i>	The GPIO peripheral from which to read (e.g., GPIOA, GPIOB, GPIOC, etc.).
----	--------------	---

Returns

The value of the entire port.

Note

None

5.107.2.6 MCAL_GPIO_TogglePin()

```
void MCAL_GPIO_TogglePin (
    GPIO_TypeDef * GPIOx,
    uint16 PinNumber )
```

Toggles the state of a specified GPIO pin.

This function toggles the output state of the specified GPIO pin.

Parameters

in	<i>GPIOx</i>	Pointer to the GPIO peripheral base address.
in	<i>PinNumber</i>	The number of the pin to toggle. This parameter should be a value defined in GPIO Pin Numbers .

Return values

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

Note

This function changes the state of the pin from high to low or low to high.

Toggles the state of a specified GPIO pin.

This function inverts the state of the specified GPIO pin.

Parameters

in	<i>GPIOx</i>	The GPIO peripheral to which to write (e.g., GPIOA, GPIOB, GPIOC, etc.).
in	<i>PinNumber</i>	The pin number to be toggled (e.g., GPIO_PIN_0, GPIO_PIN_1, etc.).

Returns

None

Note

None

5.107.2.7 MCAL_GPIO_TogglePort()

```
void MCAL_GPIO_TogglePort (
    GPIO_TypeDef * GPIOx )
```

Toggles the value of a specific GPIO Port.

Toggles the state of all pins in the specified GPIO port.

This function inverts the state of the specified GPIO Port.

Parameters

in	GPIOx	The GPIO peripheral to which to write (e.g., GPIOA, GPIOB, GPIOC, etc.).
----	-------	--

Returns

None

Note

None

This function toggles the output state of each pin in the given GPIO port. If a pin is high, it will be set to low, and if it is low, it will be set to high.

Parameters

in	GPIOx	Pointer to the GPIO peripheral base address. This parameter can be one of the following values depending on the device used: GPIOA, GPIOB, GPIOC, GPIOD, GPIOE.
----	-------	---

Return values

None	
------	--

Note

This function affects all pins in the specified GPIO port.

5.107.2.8 MCAL_GPIO_WritePin()

```
void MCAL_GPIO_WritePin (
    GPIO_TypeDef * GPIOx,
    uint16 PinNumber,
    uint8 Value )
```

Writes a value to a specified GPIO pin.

This function sets the state of the specified GPIO pin to the provided value.

Parameters

in	<i>GPIOx</i>	Pointer to the GPIO peripheral base address.
in	<i>PinNumber</i>	The number of the pin to write to. This parameter should be a value defined in GPIO Pin Numbers .
in	<i>Value</i>	The value to write to the pin. This parameter should be one of the values defined in GPIO Pin States .

Return values

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

Note

This function sets the output value of the pin.

Writes a value to a specified GPIO pin.

This function sets or clears the specified GPIO pin according to the provided value.

Parameters

in	<i>GPIOx</i>	The GPIO peripheral to which to write (e.g., GPIOA, GPIOB, GPIOC, etc.).
in	<i>PinNumber</i>	The pin number to be written (e.g., GPIO_PIN_0, GPIO_PIN_1, etc.).
in	<i>Value</i>	The value to be written (either GPIO_PIN_SET or GPIO_PIN_RESET).

Returns

None

Note

None

5.107.2.9 MCAL_GPIO_WritePort()

```
void MCAL_GPIO_WritePort (
    GPIO_TypeDef * GPIOx,
    uint16 Value )
```

Writes a value to the entire GPIO port.

This function sets the state of all the pins in the specified GPIO port to the provided value.

Parameters

in	<i>GPIOx</i>	Pointer to the GPIO peripheral base address.
in	<i>Value</i>	The value to write to the port. This parameter is a 16-bit value representing the state of each pin.

Return values

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

Note

This function sets the output values of all pins in the port.

This function sets the output data register of the specified GPIO port to the provided value.

Parameters

in	<i>GPIOx</i>	The GPIO peripheral to which to write (e.g., GPIOA, GPIOB, GPIOC, etc.).
in	<i>Value</i>	The value to be written to the entire port.

Returns

None

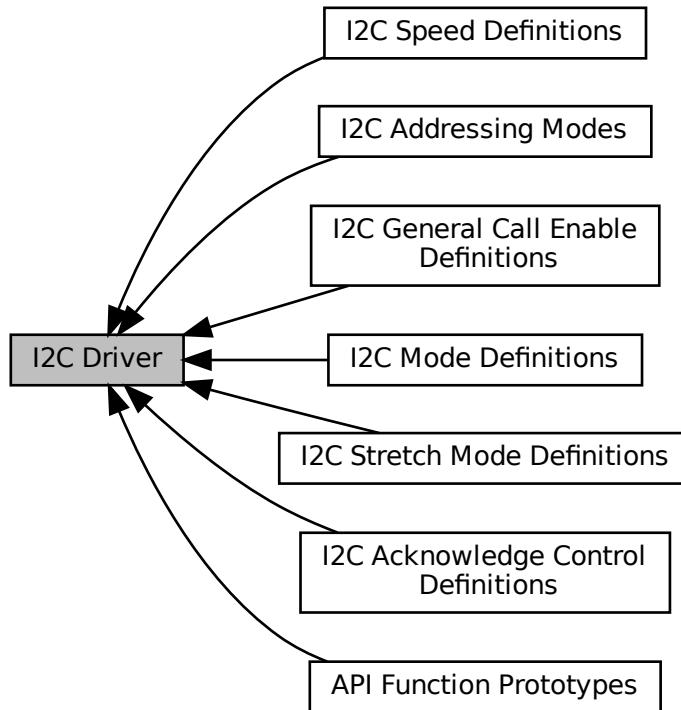
Note

None

5.108 I2C Driver

I2C driver APIs for STM32F401xx MCU.

Collaboration diagram for I2C Driver:



Modules

- I2C Speed Definitions
- I2C Stretch Mode Definitions
- I2C Mode Definitions
- I2C Addressing Modes
- I2C Acknowledge Control Definitions
- I2C General Call Enable Definitions
- API Function Prototypes

Classes

- struct [S_I2C_Slave_address](#)
Structure for I2C slave address configuration.
- struct [S_I2C_Config_t](#)
Structure for I2C configuration.

Enumerations

- enum `Slave_State` {
 I2C_EV_Stop , I2C_Error_AF , I2C_Ev_Address_Matched , I2C_Ev_Data_Req ,
 I2C_Ev_Data_RCV }
 Enumeration for I2C slave states.
- enum `StopCondition` { `WithStop` , `WithoutStop` }
 Enumeration for I2C stop condition.
- enum `Repeated_Start` { `Start` , `Repeated` }
 Enumeration for I2C repeated start condition.
- enum `Functional_State` { `DISABLE` , `ENABLE` }
 Enumeration for functional state.
- enum `FlagStatus` { `RESET` = 0 , `SET` = !`RESET` , `Reset` , `Set` }
 Enumeration for flag status.
- enum `Status` {
 I2C_Flag_Busy , EV5 , EV6 , EV7 ,
 EV8 , EV8_1 , Master_Byte_Transmitting = (uint32_t)(0x00070080) }
 Enumeration for I2C status flags.
- enum `I2C_Direction` { `I2C_Direction_Transmitter` , `I2C_Direction_Receiver` }
 Enumeration for I2C direction.

5.108.1 Detailed Description

I2C driver APIs for STM32F401xx MCU.

5.108.2 Enumeration Type Documentation

5.108.2.1 FlagStatus

```
enum FlagStatus
```

Enumeration for flag status.

Enumerator

Reset	Flag is reset.
Set	Flag is set.

5.108.2.2 Functional_State

```
enum Functional_State
```

Enumeration for functional state.

Enumerator

DISABLE	Disable the functionality.
ENABLE	Enable the functionality.

5.108.2.3 I2C_Direction

```
enum I2C_Direction
```

Enumeration for I2C direction.

Enumerator

I2C_Direction_Transmitter	Transmitter mode.
I2C_Direction_Receiver	Receiver mode.

5.108.2.4 Repeated_Start

```
enum Repeated_Start
```

Enumeration for I2C repeated start condition.

Enumerator

Start	Start condition is generated.
Repeated	Repeated start condition is generated.

5.108.2.5 Slave_State

```
enum Slave_State
```

Enumeration for I2C slave states.

Enumerator

I2C_EV_Stop	Stop condition detected.
I2C_Error_AF	Acknowledge failure error.
I2C_Ev_Address_Matched	Address matched event.
I2C_Ev_Data_Req	Data request event (Slave_Send_Data).
I2C_Ev_Data_RCV	Data received event (Slave_Read_Data).

5.108.2.6 Status

```
enum Status
```

Enumeration for I2C status flags.

Enumerator

I2C_Flag_Busy	I2C is busy.
EV5	Event 5: Start bit transmitted.
EV6	Event 6: Address sent.
EV7	Event 7: Data register not empty.
EV8	Event 8: Transmit data register empty.
EV8_1	Event 8_1: Transmit data register empty, write Data1 in DR.
Master_Byte_Transmitting	Master mode, byte transmitting.

5.108.2.7 StopCondition

```
enum StopCondition
```

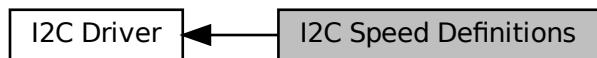
Enumeration for I2C stop condition.

Enumerator

WithStop	Stop condition is generated after the transfer.
WithoutStop	Stop condition is not generated.

5.109 I2C Speed Definitions

Collaboration diagram for I2C Speed Definitions:



Macros

- #define I2C_SCK_SM_50K (0x50000UL)
- #define I2C_SCK_SM_100K (100000UL)
- #define I2C_SCK_FM_200K (200000UL)
- #define I2C_SCK_FM_400K (400000UL)

5.109.1 Detailed Description

Defines for I2C clock speeds.

5.109.2 Macro Definition Documentation

5.109.2.1 I2C_SCK_FM_200K

```
#define I2C_SCK_FM_200K (200000UL)
```

Fast mode 200 kHz clock speed (not supported yet).

5.109.2.2 I2C_SCK_FM_400K

```
#define I2C_SCK_FM_400K (400000UL)
```

Fast mode 400 kHz clock speed (not supported yet).

5.109.2.3 I2C_SCK_SM_100K

```
#define I2C_SCK_SM_100K (100000UL)
```

Standard mode 100 kHz clock speed.

5.109.2.4 I2C_SCK_SM_50K

```
#define I2C_SCK_SM_50K (0x50000UL)
```

Standard mode 50 kHz clock speed.

5.110 I2C Stretch Mode Definitions

Collaboration diagram for I2C Stretch Mode Definitions:



Macros

- `#define I2C_StripMode_enabled 0x00000000U`
- `#define I2C_StripMode_disabled I2C_CR1_NOSTRETCH`

5.110.1 Detailed Description

Defines for I2C clock stretching mode.

5.110.2 Macro Definition Documentation

5.110.2.1 I2C_StripMode_disabled

```
#define I2C_StripMode_disabled I2C_CR1_NOSTRETCH
```

Clock stretching disabled.

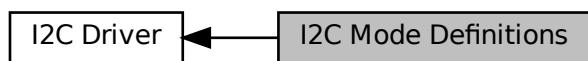
5.110.2.2 I2C_StripMode_enabled

```
#define I2C_StripMode_enabled 0x00000000U
```

Clock stretching enabled.

5.111 I2C Mode Definitions

Collaboration diagram for I2C Mode Definitions:



Macros

- #define I2C_Mode_I2C 0
- #define I2C_Mode_SMBus I2C_CR1_SMBUS

5.111.1 Detailed Description

Defines for I2C and SMBus modes.

5.111.2 Macro Definition Documentation

5.111.2.1 I2C_Mode_I2C

```
#define I2C_Mode_I2C 0
```

I2C mode.

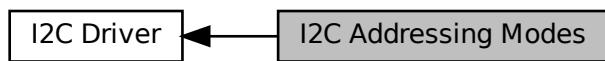
5.111.2.2 I2C_Mode_SMBus

```
#define I2C_Mode_SMBus I2C_CR1_SMBUS
```

SMBus mode.

5.112 I2C Addressing Modes

Collaboration diagram for I2C Addressing Modes:



Macros

- #define I2C_Addressing_Slave_7bits 0
- #define I2C_Addressing_Slave_10bits (uint16_t)(1<<15)

5.112.1 Detailed Description

Defines for I2C slave addressing modes.

5.112.2 Macro Definition Documentation

5.112.2.1 I2C_Addressing_Slave_10bits

```
#define I2C_Addressing_Slave_10bits (uint16_t)(1<<15)
```

10-bit addressing mode.

5.112.2.2 I2C_Addressing_Slave_7bits

```
#define I2C_Addressing_Slave_7bits 0
```

7-bit addressing mode.

5.113 I2C Acknowledge Control Definitions

Collaboration diagram for I2C Acknowledge Control Definitions:



Macros

- #define I2C_Ack_Control_Enable I2C_CR1_ACK
- #define I2C_Ack_Control_Disable 0

5.113.1 Detailed Description

Defines for I2C acknowledge control.

5.113.2 Macro Definition Documentation

5.113.2.1 I2C_Ack_Control_Disable

```
#define I2C_Ack_Control_Disable 0
```

Acknowledge disabled.

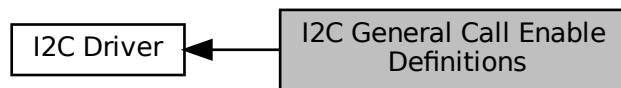
5.113.2.2 I2C_Ack_Control_Enable

```
#define I2C_Ack_Control_Enable I2C_CR1_ACK
```

Acknowledge enabled.

5.114 I2C General Call Enable Definitions

Collaboration diagram for I2C General Call Enable Definitions:



Macros

- #define I2C_ENGC_Enable I2C_CR1_ENGC
- #define I2C_ENGC_Disable 0x00

5.114.1 Detailed Description

Defines for enabling or disabling general call.

5.114.2 Macro Definition Documentation

5.114.2.1 I2C_ENGC_Disable

```
#define I2C_ENGC_Disable 0x00
```

General call disabled.

5.114.2.2 I2C_ENGC_Enable

```
#define I2C_ENGC_Enable I2C_CR1_ENGC
```

General call enabled.

5.115 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void [MCAL_I2C_Init \(I2C_TypeDef *I2Cx, S_I2C_Config_t *I2C_Init_Struct\)](#)
Initializes the I2C peripheral.
- void [MCAL_I2C_Deinit \(I2C_TypeDef *I2Cx\)](#)
Deinitializes the I2C peripheral.
- void [MCAL_I2C_Set_GPIO \(I2C_TypeDef *I2Cx\)](#)
Sets the GPIO configuration for I2C.
- void [MCAL_I2C_Master_Tx \(I2C_TypeDef *I2Cx, uint16_t SlaveAdd, uint8_t *dataout, uint32_t datalen, StopCondition Stop, Repeated_Start start\)](#)
Transmits data from the master to the slave.
- void [MCAL_I2C_Master_Rx \(I2C_TypeDef *I2Cx, uint16_t SlaveAdd, uint8_t *dataout, uint32_t datalen, StopCondition Stop, Repeated_Start start\)](#)
Receives data from the master.
- void [MCAL_I2C_Slave_SendData \(I2C_TypeDef *I2Cx, uint8_t data\)](#)
Sends data from the slave.
- uint8_t [MCAL_I2C_SlaveReceiveData \(I2C_TypeDef *I2Cx\)](#)
Receives data in slave mode.
- void [I2C_Generate_Start \(I2C_TypeDef *I2Cx, Functional_State state, Repeated_Start start\)](#)
Generates a start condition on the I2C bus.
- FlagStatus [I2C_Get_FlagStatus \(I2C_TypeDef *I2Cx, Status flag\)](#)
Gets the status of a specific flag.
- void [I2C_SendAddress \(I2C_TypeDef *I2Cx, uint16_t SlaveAddress, I2C_Direction I2C_Direction\)](#)
Sends an address to the I2C bus.
- void [I2C_Generate_Stop \(I2C_TypeDef *I2Cx, Functional_State NewState\)](#)
Generates a stop condition on the I2C bus.
- void [I2C_ACKnowledgeConfig \(I2C_TypeDef *I2Cx, Functional_State NewState\)](#)
Configures the I2C acknowledge feature.

5.115.1 Detailed Description

5.115.2 Function Documentation

5.115.2.1 I2C_ACKnowledgeConfig()

```
void I2C_ACKnowledgeConfig (
    I2C_TypeDef * I2Cx,
    Functional_State NewState )
```

Configures the I2C acknowledge feature.

This function configures the acknowledge feature for the I2C bus.

Parameters

in	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
in	<i>NewState</i>	Specifies whether to enable or disable acknowledgment.

Configures the I2C acknowledge feature.

This function enables or disables the acknowledgment feature of the I2C peripheral.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral instance.
<i>NewState</i>	The state to enable or disable acknowledgment.

5.115.2.2 I2C_Generate_Start()

```
void I2C_Generate_Start (
    I2C_TypeDef * I2Cx,
    Functional_State state,
    Repeated_Start start )
```

Generates a start condition on the I2C bus.

This function generates a start condition on the I2C bus.

Parameters

in	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
in	<i>state</i>	Specifies whether to enable or disable the start condition.
in	<i>start</i>	Specifies whether to generate a repeated start condition.

Generates a start condition on the I2C bus.

This function sets the START bit in the I2C_CR1 register to generate a start condition. If the start condition is repeated, it waits for the I2C bus to be free before generating the start condition.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral instance.
<i>state</i>	The state to enable or disable the start condition.
<i>start</i>	Indicates whether to generate a repeated start condition.

5.115.2.3 I2C_Generate_Stop()

```
void I2C_Generate_Stop (
    I2C_TypeDef * I2Cx,
    Functional_State NewState )
```

Generates a stop condition on the I2C bus.

This function generates a stop condition on the I2C bus.

Parameters

<i>in</i>	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
<i>in</i>	<i>NewState</i>	Specifies whether to enable or disable the stop condition.

This function sets the STOP bit in the I2C_CR1 register to generate a stop condition.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral instance.
<i>NewState</i>	The state to enable or disable the stop condition.

5.115.2.4 I2C_Get_FlagStatus()

```
FlagStatus I2C_Get_FlagStatus (
    I2C_TypeDef * I2Cx,
    Status flag )
```

Gets the status of a specific flag.

This function checks the status of a specific I2C flag.

Parameters

<i>in</i>	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
<i>in</i>	<i>flag</i>	The flag to check.

Returns

The status of the flag (Set or Reset).

Gets the status of a specific flag.

This function returns the status of the specified I2C flag.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral instance.
<i>flag</i>	The flag to check.

Returns

FlagStatus The status of the specified flag (Set or Reset).

5.115.2.5 I2C_SendAddress()

```
void I2C_SendAddress (
    I2C_TypeDef * I2Cx,
    uint16_t SlaveAddress,
    I2C_Direction Direction )
```

Sends an address to the I2C bus.

This function sends an address to the I2C bus.

Parameters

<i>in</i>	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
<i>in</i>	<i>SlaveAddress</i>	The address of the I2C slave.
<i>in</i>	<i>I2C_Direction</i>	The direction of the data transfer (transmitter or receiver).

Sends an address to the I2C bus.

This function sends the 7-bit slave address along with the read/write bit to the I2C peripheral.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral instance.
<i>SlaveAddress</i>	The 7-bit slave address to be sent.
<i>Direction</i>	The direction of communication (transmit or receive).

5.115.2.6 MCAL_I2C_Deinit()

```
void MCAL_I2C_Deinit (
    I2C_TypeDef * I2Cx )
```

Deinitializes the I2C peripheral.

This function deinitializes the I2C peripheral and resets it to its default state.

Parameters

in	I2Cx	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
----	------	---

Deinitializes the I2C peripheral.

Parameters

I2Cx	Pointer to the I2C peripheral (I2C1 or I2C2).
------	---

Return values

None	
------	--

Note

This function disables the I2C peripheral and resets its configuration.

5.115.2.7 MCAL_I2C_Init()

```
void MCAL_I2C_Init (
    I2C_TypeDef * I2Cx,
    S_I2C_Config_t * I2C_Init_Struct )
```

Initializes the I2C peripheral.

This function initializes the I2C peripheral with the provided configuration.

Parameters

in	I2Cx	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
in	I2C_Init_Struct	Pointer to the configuration structure for the I2C peripheral.
	I2Cx	Pointer to the I2C peripheral (I2C1 or I2C2).
	I2C_Init_Struct	Pointer to the I2C initialization structure.

Return values

None	
------	--

Note

This function configures the I2C peripheral according to the specified parameters and enables the I2C peripheral.

5.115.2.8 MCAL_I2C_Master_Rx()

```
void MCAL_I2C_Master_Rx (
    I2C_TypeDef * I2Cx,
    uint16_t SlaveAdd,
    uint8_t * dataout,
    uint32_t datalen,
    StopCondition Stop,
    Repeated_Start start )
```

Receives data from the master.

This function receives data from the I2C master.

Parameters

in	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
in	<i>SlaveAdd</i>	Address of the I2C slave.
out	<i>dataout</i>	Pointer to the buffer where received data will be stored.
in	<i>datalen</i>	Length of the data to be received.
in	<i>Stop</i>	Specifies whether to generate a stop condition.
in	<i>start</i>	Specifies whether to generate a repeated start condition.

Receives data from the master.

This function initiates a start condition, sends the address of the slave device for receiving, and then receives data. It supports optional stop conditions after the reception.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral instance.
<i>SlaveAdd</i>	The 7-bit address of the slave device.
<i>dataout</i>	Pointer to the buffer where received data will be stored.
<i>datalen</i>	The number of data bytes to receive.
<i>Stop</i>	Indicates whether to generate a stop condition after reception.
<i>start</i>	Indicates whether to generate a repeated start condition.

5.115.2.9 MCAL_I2C_Master_Tx()

```
void MCAL_I2C_Master_Tx (
    I2C_TypeDef * I2Cx,
```

```
    uint16_t SlaveAdd,
    uint8_t * dataout,
    uint32_t datalen,
    StopCondition Stop,
    Repeated_Start start )
```

Transmits data from the master to the slave.

This function sends data from the I2C master to a specified I2C slave.

Parameters

in	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
in	<i>SlaveAdd</i>	Address of the I2C slave.
in	<i>dataout</i>	Pointer to the data to be transmitted.
in	<i>datalen</i>	Length of the data to be transmitted.
in	<i>Stop</i>	Specifies whether to generate a stop condition.
in	<i>start</i>	Specifies whether to generate a repeated start condition.

Transmits data from the master to the slave.

This function initiates a start condition, sends the address of the slave device, and then transmits data. It supports optional repeated start conditions and the generation of a stop condition after the transmission.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral instance.
<i>SlaveAdd</i>	The 7-bit address of the slave device.
<i>dataout</i>	Pointer to the data buffer to be transmitted.
<i>datalen</i>	The number of data bytes to transmit.
<i>Stop</i>	Indicates whether to generate a stop condition after transmission.
<i>start</i>	Indicates whether to generate a repeated start condition.

5.115.2.10 MCAL_I2C_Set_GPIO()

```
void MCAL_I2C_Set_GPIO (
    I2C_TypeDef * I2Cx )
```

Sets the GPIO configuration for I2C.

This function configures the GPIO pins associated with the I2C peripheral.

Parameters

in	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
----	-------------	---

Sets the GPIO configuration for I2C.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1 or I2C2).
-------------	---

Return values

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

Note

This function configures the GPIO pins associated with the I2C peripheral for alternate function mode, open-drain configuration, and the appropriate alternate function number.

5.115.2.11 MCAL_I2C_Slave_SendData()

```
void MCAL_I2C_Slave_SendData (
    I2C_TypeDef * I2Cx,
    uint8_t data )
```

Sends data from the slave.

This function sends data from the I2C slave to the master.

Parameters

in	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
in	<i>data</i>	Data to be sent by the slave.

Sends data from the slave.

This function sends a byte of data from the I2C slave device.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral instance.
<i>data</i>	The byte of data to be sent.

5.115.2.12 MCAL_I2C_SlaveReceiveData()

```
uint8_t MCAL_I2C_SlaveReceiveData (
    I2C_TypeDef * I2Cx )
```

Receives data in slave mode.

This function receives data from the I2C master in slave mode.

Parameters

in	<i>I2Cx</i>	Pointer to the I2C peripheral (I2C1, I2C2, etc.).
----	-------------	---

Returns

The received data.

Receives data in slave mode.

This function receives a byte of data from the I2C slave device.

Parameters

<i>I2Cx</i>	Pointer to the I2C peripheral instance.
-------------	---

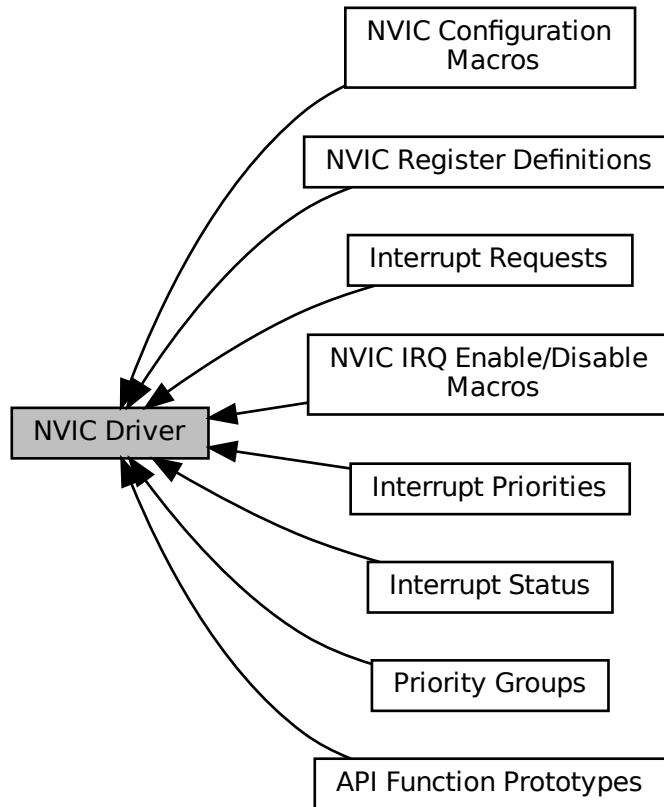
Returns

`uint8_t` The received byte of data.

5.116 NVIC Driver

NVIC driver APIs for STM32F401xx MCU.

Collaboration diagram for NVIC Driver:



Modules

- [NVIC Register Definitions](#)
Macros for accessing NVIC registers.
- [NVIC Configuration Macros](#)
- [Interrupt Status](#)
- [Priority Groups](#)
- [Interrupt Priorities](#)
- [Interrupt Requests](#)
- [API Function Prototypes](#)
- [NVIC IRQ Enable/Disable Macros](#)

Macros for enabling and disabling NVIC IRQ for different peripherals.

5.116.1 Detailed Description

NVIC driver APIs for STM32F401xx MCU.

5.117 NVIC Register Definitions

Macros for accessing NVIC registers.

Collaboration diagram for NVIC Register Definitions:



Macros

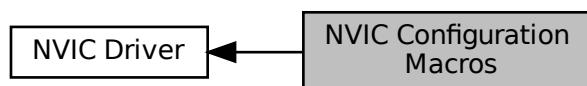
- `#define NVIC_ISER0 (*(volatile uint32_t *) (NVIC_BASE + 0x0))`
Interrupt Set Enable Register 0 (ISER0) Used to enable interrupts 0 to 31. Write a 1 to the corresponding bit position to enable an interrupt.
- `#define NVIC_ISER1 (*(volatile uint32_t *) (NVIC_BASE + 0x4))`
Interrupt Set Enable Register 1 (ISER1) Used to enable interrupts 32 to 63. Write a 1 to the corresponding bit position to enable an interrupt.
- `#define NVIC_ISER2 (*(volatile uint32_t *) (NVIC_BASE + 0x8))`
Interrupt Set Enable Register 2 (ISER2) Used to enable interrupts 64 to 95. Write a 1 to the corresponding bit position to enable an interrupt.
- `#define NVIC_ICER0 (*(volatile uint32_t *) (NVIC_BASE + 0x80))`
Interrupt Clear Enable Register 0 (ICER0) Used to disable interrupts 0 to 31. Write a 1 to the corresponding bit position to disable an interrupt.
- `#define NVIC_ICER1 (*(volatile uint32_t *) (NVIC_BASE + 0x84))`
Interrupt Clear Enable Register 1 (ICER1) Used to disable interrupts 32 to 63. Write a 1 to the corresponding bit position to disable an interrupt.
- `#define NVIC_ICER2 (*(volatile uint32_t *) (NVIC_BASE + 0x88))`
Interrupt Clear Enable Register 2 (ICER2) Used to disable interrupts 64 to 95. Write a 1 to the corresponding bit position to disable an interrupt.

5.117.1 Detailed Description

Macros for accessing NVIC registers.

5.118 NVIC Configuration Macros

Collaboration diagram for NVIC Configuration Macros:



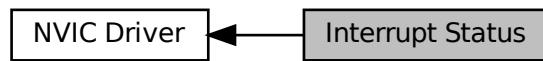
Macros

- #define **SCB_VECTKEY** 0x05FA0000UL
Vector key for setting priority grouping in NVIC.
- #define **SCB_VECTKEY_MASK** 0xFFFF0000UL
Mask for vector key.
- #define **NVIC_PRIGROUP_SET_MASK** 0x700UL
Mask for setting NVIC priority grouping.
- #define **NVIC_PRIGROUP_CLEAR_MASK** 0xFFFFF8FFUL
Mask for clearing NVIC priority grouping.

5.118.1 Detailed Description

5.119 Interrupt Status

Collaboration diagram for Interrupt Status:



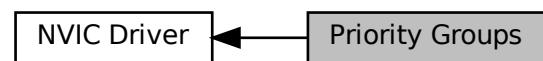
Macros

- #define **NVIC_INTERRUPT_ACTIVE** 1UL
Status indicating an active interrupt.
- #define **NVIC_INTERRUPT_INACTIVE** 0UL
Status indicating an inactive interrupt.

5.119.1 Detailed Description

5.120 Priority Groups

Collaboration diagram for Priority Groups:



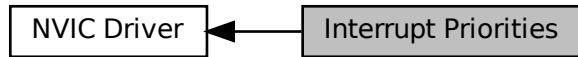
Macros

- `#define NVIC_PRIO_16GRP_0SUBGRP 0x300U`
Priority grouping configuration for 16 groups and 0 subgroups.
- `#define NVIC_PRIO_8GRP_2SUBGRP 0x400U`
Priority grouping configuration for 8 groups and 2 subgroups.
- `#define NVIC_PRIO_4GRP_4SUBGRP 0x500U`
Priority grouping configuration for 4 groups and 4 subgroups.
- `#define NVIC_PRIO_2GRP_8SUBGRP 0x600U`
Priority grouping configuration for 2 groups and 8 subgroups.
- `#define NVIC_PRIO_0GRP_8SUBGRP 0x700U`
Priority grouping configuration for 0 groups and 8 subgroups.

5.120.1 Detailed Description

5.121 Interrupt Priorities

Collaboration diagram for Interrupt Priorities:



Macros

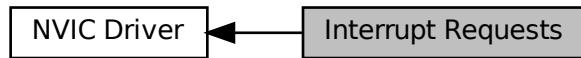
- `#define NVIC_PRIO_0000 0x00U`
Priority level 0.
- `#define NVIC_PRIO_0001 0x10U`
Priority level 1.
- `#define NVIC_PRIO_0010 0x20U`
Priority level 2.
- `#define NVIC_PRIO_0011 0x30U`
Priority level 3.
- `#define NVIC_PRIO_0100 0x40U`
Priority level 4.
- `#define NVIC_PRIO_0101 0x50U`
Priority level 5.
- `#define NVIC_PRIO_0110 0x60U`
Priority level 6.
- `#define NVIC_PRIO_0111 0x70U`
Priority level 7.
- `#define NVIC_PRIO_1000 0x80U`
Priority level 8.

- #define `NVIC_PRIO_1001` 0x90U
Priority level 9.
- #define `NVIC_PRIO_1010` 0xA0U
Priority level 10.
- #define `NVIC_PRIO_1011` 0xB0U
Priority level 11.
- #define `NVIC_PRIO_1100` 0xC0U
Priority level 12.
- #define `NVIC_PRIO_1101` 0xD0U
Priority level 13.
- #define `NVIC_PRIO_1110` 0xE0U
Priority level 14.
- #define `NVIC_PRIO_1111` 0xF0U
Priority level 15.

5.121.1 Detailed Description

5.122 Interrupt Requests

Collaboration diagram for Interrupt Requests:



Macros

- #define `EXTI0_IRQ` 6
External Interrupt Request 0.
- #define `EXTI1_IRQ` 7
External Interrupt Request 1.
- #define `EXTI2_IRQ` 8
External Interrupt Request 2.
- #define `EXTI3_IRQ` 9
External Interrupt Request 3.
- #define `EXTI4_IRQ` 10
External Interrupt Request 4.
- #define `EXTI5_IRQ` 23
External Interrupt Request 5.
- #define `EXTI6_IRQ` 23
External Interrupt Request 6.
- #define `EXTI7_IRQ` 23
External Interrupt Request 7.

- #define `EXTI8_IRQ` 23
External Interrupt Request 8.
- #define `EXTI9_IRQ` 23
External Interrupt Request 9.
- #define `EXTI10_IRQ` 40
External Interrupt Request 10.
- #define `EXTI11_IRQ` 40
External Interrupt Request 11.
- #define `EXTI12_IRQ` 40
External Interrupt Request 12.
- #define `EXTI13_IRQ` 40
External Interrupt Request 13.
- #define `EXTI14_IRQ` 40
External Interrupt Request 14.
- #define `EXTI15_IRQ` 40
External Interrupt Request 15.
- #define `USART1_IRQ` 37
USART1 Interrupt Request.
- #define `USART2_IRQ` 38
USART2 Interrupt Request.
- #define `USART6_IRQ` 71
USART6 Interrupt Request.
- #define `SPI1_IRQ` 35
SPI1 Interrupt Request.
- #define `SPI2_IRQ` 36
SPI2 Interrupt Request.
- #define `I2C1_EV_IRQ` 31
I2C1 Event Interrupt Request.
- #define `I2C1_ER_IRQ` 32
I2C1 Error Interrupt Request.
- #define `I2C2_EV_IRQ` 33
I2C2 Event Interrupt Request.
- #define `I2C2_ER_IRQ` 34
I2C2 Error Interrupt Request.

5.122.1 Detailed Description

5.123 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void [MCAL_NVIC_SetPriorityGrouping](#) (`uint32 priority_grouping`)

Set the priority grouping for the NVIC.
- `uint32 MCAL_NVIC_GetPriorityGrouping` (`void`)

Get the current priority grouping configuration of the NVIC.
- void [MCAL_NVIC_EnableIRQ](#) (`uint8 IRQn`)

Enable a specified IRQ.
- void [MCAL_NVIC_DisableIRQ](#) (`uint8 IRQn`)

Disable a specified IRQ.
- `uint8 MCAL_NVIC_GetPendingIRQ` (`uint8 IRQn`)

Get the pending status of a specified IRQ.
- void [MCAL_NVIC_SetPendingIRQ](#) (`uint8 IRQn`)

Set the pending status of a specified IRQ.
- void [MCAL_NVIC_ClearPendingIRQ](#) (`uint8 IRQn`)

Clear the pending status of a specified IRQ.
- `uint8 MCAL_NVIC_GetActive` (`uint8 IRQn`)

Get the active status of a specified IRQ.
- void [MCAL_NVIC_SetPriority](#) (`uint8 IRQn, uint8 priority`)

Set the priority of a specified IRQ.
- `uint8 MCAL_NVIC_GetPriority` (`uint8 IRQn`)

Retrieves the priority level of a specified IRQn.
- void [MCAL_NVIC_SystemReset](#) (`void`)

Performs a system reset.

5.123.1 Detailed Description

5.123.2 Function Documentation

5.123.2.1 MCAL_NVIC_ClearPendingIRQ()

```
void MCAL_NVIC_ClearPendingIRQ (
    uint8 IRQn )
```

Clear the pending status of a specified IRQ.

Parameters

<code>IRQn</code>	IRQ number to clear pending status as defined in Interrupt Requests
-------------------	---

Returns

None

Clear the pending status of a specified IRQ.

Parameters

<i>IRQn</i>	Number of interrupt request as defined in vector table or in Interrupt_Requests_Numbers_define .
-------------	--

Returns

None

Note

None

5.123.2.2 MCAL_NVIC_DisableIRQ()

```
void MCAL_NVIC_DisableIRQ (
    uint8 IRQn )
```

Disable a specified IRQ.

Parameters

<i>IRQn</i>	IRQ number to disable as defined in Interrupt Requests
-------------	--

Returns

None

Disable a specified IRQ.

Parameters

<i>IRQn</i>	Number of interrupt request as defined in vector table or in Interrupt_Requests_Numbers_define .
-------------	--

Returns

None

Note

None

5.123.2.3 MCAL_NVIC_EnableIRQ()

```
void MCAL_NVIC_EnableIRQ (
    uint8 IRQn )
```

Enable a specified IRQ.

Parameters

<i>IRQn</i>	IRQ number to enable as defined in Interrupt Requests
-------------	---

Returns

None

Enable a specified IRQ.

Parameters

<i>IRQn</i>	Number of interrupt request as defined in vector table or in Interrupt_Requests_Numbers_define .
-------------	--

Returns

None

Note

None

5.123.2.4 MCAL_NVIC_GetActive()

```
uint8 MCAL_NVIC_GetActive (
    uint8 IRQn )
```

Get the active status of a specified IRQ.

Parameters

<i>IRQn</i>	IRQ number to check as defined in Interrupt Requests
-------------	--

Returns

1 if IRQ is active, 0 otherwise

Get the active status of a specified IRQ.

Parameters

<i>IRQn</i>	Number of interrupt request as defined in vector table or in Interrupt_Requests_Numbers_define .
-------------	--

Returns

1 if IRQn is active, 0 otherwise.

Note

None

5.123.2.5 MCAL_NVIC_GetPendingIRQ()

```
uint8 MCAL_NVIC_GetPendingIRQ (
    uint8 IRQn )
```

Get the pending status of a specified IRQ.

Parameters

<i>IRQn</i>	IRQ number to check as defined in Interrupt Requests
-------------	--

Returns

1 if IRQ is pending, 0 otherwise

Get the pending status of a specified IRQ.

Parameters

<i>IRQn</i>	Number of interrupt request as defined in vector table or in Interrupt_Requests_Numbers_define .
-------------	--

Returns

1 if IRQn is pending, 0 otherwise.

Note

None

5.123.2.6 MCAL_NVIC_GetPriority()

```
uint8 MCAL_NVIC_GetPriority (
    uint8 IRQn )
```

Retrieves the priority level of a specified IRQn.

This function reads the priority level assigned to the interrupt request number (IRQn). The priority value returned should be one of the defined values in [Interrupt_Priorities_define](#).

Parameters

in	<i>IRQn</i>	The interrupt request number as defined in the vector table or in <code>Interrupt_Requests_Numbers_define</code> .
out	<i>None</i>	

Returns

`uint8` The priority value for the specified IRQn.

Note

None

Retrieves the priority level of a specified IRQn.

Parameters

<i>IRQn</i>	Number of interrupt request as defined in vector table or in <code>Interrupt_Requests_Numbers_define</code> .
-------------	---

Returns

Priority level of the interrupt.

Note

None

5.123.2.7 MCAL_NVIC_GetPriorityGrouping()

```
uint32 MCAL_NVIC_GetPriorityGrouping (
    void )
```

Get the current priority grouping configuration of the NVIC.

Returns

Current priority grouping configuration as defined in [Priority Groups](#)

Get the current priority grouping configuration of the NVIC.

Returns

Configuration of priority grouping as defined in `priority_groups_define`.

Note

None

5.123.2.8 MCAL_NVIC_SetPendingIRQ()

```
void MCAL_NVIC_SetPendingIRQ (
    uint8 IRQn )
```

Set the pending status of a specified IRQ.

Parameters

<i>IRQn</i>	IRQ number to set as pending as defined in Interrupt Requests
-------------	---

Returns

None

Set the pending status of a specified IRQ.

Parameters

<i>IRQn</i>	Number of interrupt request as defined in vector table or in Interrupt_Requests_Numbers_define .
-------------	--

Returns

None

Note

None

5.123.2.9 MCAL_NVIC_SetPriority()

```
void MCAL_NVIC_SetPriority (
    uint8 IRQn,
    uint8 priority )
```

Set the priority of a specified IRQ.

Parameters

<i>IRQn</i>	IRQ number for which to set the priority as defined in Interrupt Requests
<i>priority</i>	Priority level to set as defined in Interrupt Priorities

Returns

None

Set the priority of a specified IRQ.

Parameters

<i>IRQn</i>	Number of interrupt request as defined in vector table or in Interrupt_Requests_Numbers_define .
<i>priority</i>	Priority level of the interrupt as defined in Interrupt_Priorities_define .

Returns

None

Note

None

5.123.2.10 MCAL_NVIC_SetPriorityGrouping()

```
void MCAL_NVIC_SetPriorityGrouping (
    uint32 priority_grouping )
```

Set the priority grouping for the NVIC.

Parameters

<i>priority_grouping</i>	Priority grouping configuration as defined in Priority Groups
--------------------------	---

Returns

None

Set the priority grouping for the NVIC.

Parameters

<i>priority_grouping</i>	Configuration of priority grouping as defined in <code>priority_groups_define</code> .
--------------------------	--

Returns

None

Note

None

5.123.2.11 MCAL_NVIC_SystemReset()

```
void MCAL_NVIC_SystemReset (
    void )
```

Performs a system reset.

This function resets the entire system. It does not alter the priority grouping configuration.

Parameters

in	<i>None</i>	
out	<i>None</i>	

Returns

None

Note

The priority groups remain unchanged after the system reset.

Parameters

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

Returns

None

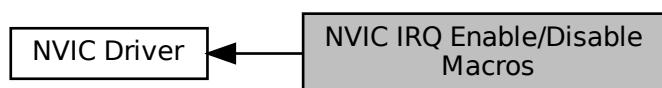
Note

Keeps priority grouping configuration unchanged.

5.124 NVIC IRQ Enable/Disable Macros

Macros for enabling and disabling NVIC IRQ for different peripherals.

Collaboration diagram for NVIC IRQ Enable/Disable Macros:



Macros

- #define NVIC_IRQ6_EXTI0_Enable (NVIC_ISER0 |= (1<<6))
- #define NVIC_IRQ7_EXTI1_Enable (NVIC_ISER0 |= (1<<7))
- #define NVIC_IRQ8_EXTI2_Enable (NVIC_ISER0 |= (1<<8))
- #define NVIC_IRQ9_EXTI3_Enable (NVIC_ISER0 |= (1<<9))
- #define NVIC_IRQ10_EXTI4_Enable (NVIC_ISER0 |= (1<<10))

- #define NVIC_IRQ23_EXTI5_9_Enable (NVIC_ISERO |= (1<<23))
- #define NVIC_IRQ40_EXTI10_15_Enable (NVIC_ISER1 |= (1<<8))
- #define NVIC_IRQ6_EXTI0_Disable (NVIC_ICERO |= (1<<6))
- #define NVIC_IRQ7_EXTI1_Disable (NVIC_ICERO |= (1<<7))
- #define NVIC_IRQ8_EXTI2_Disable (NVIC_ICERO |= (1<<8))
- #define NVIC_IRQ9_EXTI3_Disable (NVIC_ICERO |= (1<<9))
- #define NVIC_IRQ10_EXTI4_Disable (NVIC_ICERO |= (1<<10))
- #define NVIC_IRQ23_EXTI5_9_Disable (NVIC_ICERO |= (1<<23))
- #define **NVIC_IRQ40_EXTI10_15_Disable** (NVIC_ICER1 |= (1<<8))
- #define NVIC_IRQ37_USART1_Enable (NVIC_ISER1 |= 1<<(USART1_IRQ - 32))
- #define NVIC_IRQ38_USART2_Enable (NVIC_ISER1 |= 1<<(USART2_IRQ - 32))
- #define NVIC_IRQ39_USART3_Enable (NVIC_ISER1 |= 1<<(USART3_IRQ - 32))
- #define NVIC_IRQ37_USART1_Disable (NVIC_ICER1 |= 1<<(USART1_IRQ - 32))
- #define NVIC_IRQ38_USART2_Disable (NVIC_ICER1 |= 1<<(USART2_IRQ - 32))
- #define **NVIC_IRQ39_USART3_Disable** (NVIC_ICER1 |= 1<<(USART3_IRQ - 32))
- #define NVIC_IRQ35_SPI1_Enable (NVIC_ISER1 |= 1<<(SPI1_IRQ - 32))
- #define NVIC_IRQ36_SPI2_Enable (NVIC_ISER1 |= 1<<(SPI2_IRQ - 32))
- #define NVIC_IRQ35_SPI1_Disable (NVIC_ICER1 &= ~ (1<<(SPI1_IRQ - 32)))
- #define **NVIC_IRQ36_SPI2_Disable** (NVIC_ICER1 &= ~ (1<<(SPI2_IRQ - 32)))
- #define NVIC_IRQ31_I2C1_EV_Enable (NVIC_ISER0 |= 1<<(I2C1_EV_IRQ))
- #define NVIC_IRQ32_I2C1_ER_Enable (NVIC_ISER1 |= 1<<(I2C1_ER_IRQ - 32))
- #define NVIC_IRQ33_I2C2_EV_Enable (NVIC_ISER1 |= 1<<(I2C2_EV_IRQ - 32))
- #define NVIC_IRQ34_I2C2_ER_Enable (NVIC_ISER1 |= 1<<(I2C2_ER_IRQ - 32))
- #define NVIC_IRQ31_I2C1_EV_Disable (NVIC_ICER0 &= ~ (1<<(I2C1_EV_IRQ)))
- #define NVIC_IRQ32_I2C1_ER_Disable (NVIC_ICER1 &= ~ (1<<(I2C1_ER_IRQ - 32)))
- #define NVIC_IRQ33_I2C2_EV_Disable (NVIC_ICER1 &= ~ (1<<(I2C2_EV_IRQ - 32)))
- #define **NVIC_IRQ34_I2C2_ER_Disable** (NVIC_ICER1 &= ~ (1<<(I2C2_ER_IRQ - 32)))

5.124.1 Detailed Description

Macros for enabling and disabling NVIC IRQ for different peripherals.

These macros are used to enable or disable NVIC IRQ for EXTI, USART, SPI, and I2C peripherals.

5.124.2 Macro Definition Documentation

5.124.2.1 NVIC_IRQ10_EXTI4_Disable

```
#define NVIC_IRQ10_EXTI4_Disable (NVIC_ICERO |= (1<<10))
```

Disable EXTI Line 5 to 9 interrupt (clear bit 23 in NVIC_ICER0 register).

5.124.2.2 NVIC_IRQ10_EXTI4_Enable

```
#define NVIC_IRQ10_EXTI4_Enable (NVIC_ISER0 |= (1<<10))
```

Enable EXTI Line 5 to 9 interrupt (set bit 23 in NVIC_ISER0 register).

5.124.2.3 NVIC IRQ23 EXTI5_9 Disable

```
#define NVIC_IRQ23_EXTI5_9_Disable (NVIC_ICERO |= (1<<23))
```

Disable EXTI Line 10 to 15 interrupt (clear bit 8 in NVIC_ICER1 register).

5.124.2.4 NVIC IRQ23 EXTI5_9 Enable

```
#define NVIC_IRQ23_EXTI5_9_Enable (NVIC_ISERO |= (1<<23))
```

Enable EXTI Line 10 to 15 interrupt (set bit 8 in NVIC_ISER1 register).

5.124.2.5 NVIC IRQ31_I2C1_EV Disable

```
#define NVIC_IRQ31_I2C1_EV_Disable (NVIC_ICERO &= ~(1<<(I2C1_EV_IRQ)))
```

Disable I2C1 error interrupt (clear bit in NVIC_ICER1 for I2C1_ER_IRQ).

5.124.2.6 NVIC IRQ31_I2C1_EV Enable

```
#define NVIC_IRQ31_I2C1_EV_Enable (NVIC_ISERO |= 1<<(I2C1_EV_IRQ))
```

< Enable I2C1 event interrupt (set bit in NVIC_ISER0 for I2C1_EV_IRQ). Enable I2C1 error interrupt (set bit in NVIC_ISER1 for I2C1_ER_IRQ).

5.124.2.7 NVIC IRQ32_I2C1_ER Disable

```
#define NVIC_IRQ32_I2C1_ER_Disable (NVIC_ICER1 &= ~(1<<(I2C1_ER_IRQ - 32)))
```

Disable I2C2 event interrupt (clear bit in NVIC_ICER1 for I2C2_EV_IRQ).

5.124.2.8 NVIC IRQ32_I2C1_ER Enable

```
#define NVIC_IRQ32_I2C1_ER_Enable (NVIC_ISER1 |= 1<<(I2C1_ER_IRQ - 32))
```

Enable I2C2 event interrupt (set bit in NVIC_ISER1 for I2C2_EV_IRQ).

5.124.2.9 NVIC IRQ33_I2C2_EV Disable

```
#define NVIC_IRQ33_I2C2_EV_Disable (NVIC_ICER1 &= ~(1<<(I2C2_EV_IRQ - 32)))
```

Disable I2C2 error interrupt (clear bit in NVIC_ICER1 for I2C2_ER_IRQ).

5.124.2.10 NVIC_IRQ33_I2C2_EV_Enable

```
#define NVIC_IRQ33_I2C2_EV_Enable (NVIC_ISER1 |= 1<<(I2C2_EV_IRQ - 32))
```

Enable I2C2 error interrupt (set bit in NVIC_ISER1 for I2C2_ER_IRQ).

5.124.2.11 NVIC_IRQ34_I2C2_ER_Enable

```
#define NVIC_IRQ34_I2C2_ER_Enable (NVIC_ISER1 |= 1<<(I2C2_ER_IRQ - 32))
```

Disable I2C1 event interrupt (clear bit in NVIC_ICER0 for I2C1_EV_IRQ).

5.124.2.12 NVIC_IRQ35_SPI1_Disable

```
#define NVIC_IRQ35_SPI1_Disable (NVIC_ICER1 &= ~ (1<<(SPI1_IRQ - 32)))
```

Disable SPI2 interrupt (clear bit in NVIC_ICER1 for SPI2 IRQ).

5.124.2.13 NVIC_IRQ35_SPI1_Enable

```
#define NVIC_IRQ35_SPI1_Enable (NVIC_ISER1 |= 1<<(SPI1_IRQ - 32))
```

< Enable SPI1 interrupt (set bit in NVIC_ISER1 for SPI1 IRQ). Enable SPI2 interrupt (set bit in NVIC_ISER1 for SPI2 IRQ).

5.124.2.14 NVIC_IRQ36_SPI2_Enable

```
#define NVIC_IRQ36_SPI2_Enable (NVIC_ISER1 |= 1<<(SPI2_IRQ - 32))
```

Disable SPI1 interrupt (clear bit in NVIC_ICER1 for SPI1 IRQ).

5.124.2.15 NVIC_IRQ37_USART1_Disable

```
#define NVIC_IRQ37_USART1_Disable (NVIC_ICER1 |= 1<<(USART1_IRQ - 32))
```

Disable USART2 interrupt (clear bit in NVIC_ICER1 for USART2 IRQ).

5.124.2.16 NVIC_IRQ37_USART1_Enable

```
#define NVIC_IRQ37_USART1_Enable (NVIC_ISER1 |= 1<<(USART1_IRQ - 32))
```

< Enable USART1 interrupt (set bit in NVIC_ISER1 for USART1 IRQ). Enable USART2 interrupt (set bit in NVIC_ISER1 for USART2 IRQ).

5.124.2.17 NVIC_IRQ38_USART2_Disable

```
#define NVIC_IRQ38_USART2_Disable (NVIC_ICER1 |= 1<<(USART2_IRQ - 32))
```

Disable USART3 interrupt (clear bit in NVIC_ICER1 for USART3 IRQ).

5.124.2.18 NVIC_IRQ38_USART2_Enable

```
#define NVIC_IRQ38_USART2_Enable (NVIC_ISER1 |= 1<<(USART2_IRQ - 32))
```

Enable USART3 interrupt (set bit in NVIC_ISER1 for USART3 IRQ).

5.124.2.19 NVIC_IRQ39_USART3_Enable

```
#define NVIC_IRQ39_USART3_Enable (NVIC_ISER1 |= 1<<(USART3_IRQ - 32))
```

Disable USART1 interrupt (clear bit in NVIC_ICER1 for USART1 IRQ).

5.124.2.20 NVIC_IRQ40_EXTI10_15_Enable

```
#define NVIC_IRQ40_EXTI10_15_Enable (NVIC_ISER1 |= (1<<8))
```

Disable EXTI Line 0 interrupt (clear bit 6 in NVIC_ICER0 register).

5.124.2.21 NVIC_IRQ6_EXTI0_Disable

```
#define NVIC_IRQ6_EXTI0_Disable (NVIC_ICERO |= (1<<6))
```

Disable EXTI Line 1 interrupt (clear bit 7 in NVIC_ICER0 register).

5.124.2.22 NVIC_IRQ6_EXTI0_Enable

```
#define NVIC_IRQ6_EXTI0_Enable (NVIC_ISERO |= (1<<6))
```

< Enable EXTI Line 0 interrupt (set bit 6 in NVIC_ISER0 register). Enable EXTI Line 1 interrupt (set bit 7 in NVIC_ISER0 register).

5.124.2.23 NVIC_IRQ7_EXTI1_Disable

```
#define NVIC_IRQ7_EXTI1_Disable (NVIC_ICERO |= (1<<7))
```

Disable EXTI Line 2 interrupt (clear bit 8 in NVIC_ICER0 register).

5.124.2.24 NVIC_IRQ7_EXTI1_Enable

```
#define NVIC_IRQ7_EXTI1_Enable (NVIC_ISER0 |= (1<<7))
```

Enable EXTI Line 2 interrupt (set bit 8 in NVIC_ISER0 register).

5.124.2.25 NVIC_IRQ8_EXTI2_Disable

```
#define NVIC_IRQ8_EXTI2_Disable (NVIC_ICER0 |= (1<<8))
```

Disable EXTI Line 3 interrupt (clear bit 9 in NVIC_ICER0 register).

5.124.2.26 NVIC_IRQ8_EXTI2_Enable

```
#define NVIC_IRQ8_EXTI2_Enable (NVIC_ISER0 |= (1<<8))
```

Enable EXTI Line 3 interrupt (set bit 9 in NVIC_ISER0 register).

5.124.2.27 NVIC_IRQ9_EXTI3_Disable

```
#define NVIC_IRQ9_EXTI3_Disable (NVIC_ICER0 |= (1<<9))
```

Disable EXTI Line 4 interrupt (clear bit 10 in NVIC_ICER0 register).

5.124.2.28 NVIC_IRQ9_EXTI3_Enable

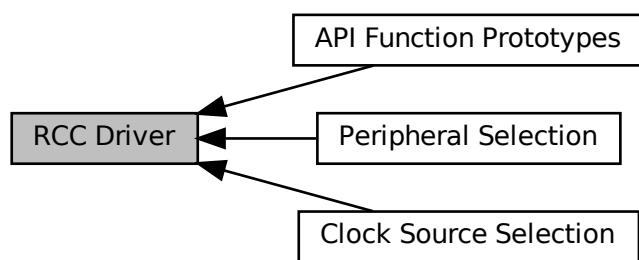
```
#define NVIC_IRQ9_EXTI3_Enable (NVIC_ISER0 |= (1<<9))
```

Enable EXTI Line 4 interrupt (set bit 10 in NVIC_ISER0 register).

5.125 RCC Driver

RCC driver APIs for STM32F401xx MCU.

Collaboration diagram for RCC Driver:



Modules

- [Clock Source Selection](#)
- [Peripheral Selection](#)
- [API Function Prototypes](#)

5.125.1 Detailed Description

RCC driver APIs for STM32F401xx MCU.

5.126 Clock Source Selection

Collaboration diagram for Clock Source Selection:



Macros

- [#define RCC_PLLCFGR_PLLSRC_HSE \(\(uint32_t\)0x00400000\)](#)
- [#define HSI_VALUE 16000000U](#)
- [#define HSE_VALUE 8000000U](#)
- [#define HSI_RC_CLK 8000000UL](#)
- [#define HSE_CLK 8000000UL](#)
- [#define RCC_SELECT_HSI \(\(uint8_t\)0x00\)](#)
- [#define RCC_SELECT_HSE \(\(uint8_t\)0x01\)](#)
- [#define RCC_SELECT_PLL \(\(uint8_t\)0x02\)](#)

5.126.1 Detailed Description

5.126.2 Macro Definition Documentation

5.126.2.1 HSE_CLK

```
#define HSE_CLK 8000000UL
```

External Oscillator Clock Source (8 MHz)

5.126.2.2 HSE_VALUE

```
#define HSE_VALUE 8000000U
```

External High-Speed Oscillator frequency (8 MHz)

5.126.2.3 HSI_RC_CLK

```
#define HSI_RC_CLK 8000000UL
```

RC Clock Source (8 MHz)

5.126.2.4 HSI_VALUE

```
#define HSI_VALUE 16000000U
```

Internal High-Speed Oscillator frequency (16 MHz)

5.126.2.5 RCC_PLLCFGR_PLLSRC_HSE

```
#define RCC_PLLCFGR_PLLSRC_HSE ((uint32_t)0x00400000)
```

PLL source is HSE (High-Speed External)

5.126.2.6 RCC_SELECT_HSE

```
#define RCC_SELECT_HSE ((uint8_t)0x01)
```

Select HSE as clock source

5.126.2.7 RCC_SELECT_HSI

```
#define RCC_SELECT_HSI ((uint8_t)0x00)
```

Select HSI as clock source

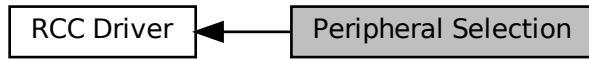
5.126.2.8 RCC_SELECT_PLL

```
#define RCC_SELECT_PLL ((uint8_t)0x02)
```

Select PLL as clock source

5.127 Peripheral Selection

Collaboration diagram for Peripheral Selection:



Macros

- #define RCC_GPIOA ((uint8_t)0x00)
- #define RCC_GPIOB ((uint8_t)0x01)
- #define RCC_GPIOC ((uint8_t)0x02)
- #define RCC_GPIOD ((uint8_t)0x03)
- #define RCC_GPIOE ((uint8_t)0x04)
- #define RCC_USART1 ((uint8_t)0x08)
- #define RCC_USART2 ((uint8_t)0x09)
- #define RCC_USART6 ((uint8_t)0xA)
- #define RCC_SPI1 ((uint8_t)0xB)
- #define RCC_SPI2 ((uint8_t)0xC)
- #define RCC_I2C1 ((uint8_t)0xD)
- #define RCC_I2C2 ((uint8_t)0xE)
- #define RCC_CRC ((uint8_t)0xF)
- #define RCC_TIM2 ((uint8_t)0x10)
- #define RCC_SYSCFG ((uint8_t)0x11)

5.127.1 Detailed Description

5.127.2 Macro Definition Documentation

5.127.2.1 RCC_CRC

```
#define RCC_CRC ((uint8_t)0x0F)
```

CRC Peripheral

5.127.2.2 RCC_GPIOA

```
#define RCC_GPIOA ((uint8_t)0x00)
```

GPIOA Peripheral

5.127.2.3 RCC_GPIOB

```
#define RCC_GPIOB ((uint8_t)0x01)
```

GPIOB Peripheral

5.127.2.4 RCC_GPIOC

```
#define RCC_GPIOC ((uint8_t)0x02)
```

GPIOC Peripheral

5.127.2.5 RCC_GPIOD

```
#define RCC_GPIOD ((uint8_t)0x03)
```

GPIOD Peripheral

5.127.2.6 RCC_GPIOE

```
#define RCC_GPIOE ((uint8_t)0x04)
```

GPIOE Peripheral

5.127.2.7 RCC_I2C1

```
#define RCC_I2C1 ((uint8_t)0x0D)
```

I2C1 Peripheral

5.127.2.8 RCC_I2C2

```
#define RCC_I2C2 ((uint8_t)0x0E)
```

I2C2 Peripheral

5.127.2.9 RCC_SPI1

```
#define RCC_SPI1 ((uint8_t)0x0B)
```

SPI1 Peripheral

5.127.2.10 RCC_SPI2

```
#define RCC_SPI2 ((uint8_t)0x0C)
```

SPI2 Peripheral

5.127.2.11 RCC_SYSCFG

```
#define RCC_SYSCFG ((uint8_t)0x11)
```

SYSCFG Peripheral

5.127.2.12 RCC_TIM2

```
#define RCC_TIM2 ((uint8_t)0x10)
```

TIM2 Peripheral

5.127.2.13 RCC_USART1

```
#define RCC_USART1 ((uint8_t)0x08)
```

USART1 Peripheral

5.127.2.14 RCC_USART2

```
#define RCC_USART2 ((uint8_t)0x09)
```

USART2 Peripheral

5.127.2.15 RCC_USART6

```
#define RCC_USART6 ((uint8_t)0x0A)
```

USART6 Peripheral

5.128 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void [MCAL_RCC_Select_Clock](#) (uint8_t clock)
Select the clock source for the MCU.
- void [MCAL_RCC_Enable_Peripheral](#) (uint8_t peripheral)
Enable the clock for a specific peripheral.
- void [MCAL_RCC_Reset_Peripheral](#) (uint8_t peripheral)
Reset a specific peripheral.
- uint32_t [MCAL_RCC_GetSYS_CLKFreq](#) (void)
Get the frequency of the system clock.
- uint32_t [MCAL_RCC_GetHCLKFreq](#) (void)
Get the frequency of the AHB bus clock.
- uint32_t [MCAL_RCC_GetPCLK1Freq](#) (void)
Get the frequency of the APB1 bus clock.
- uint32_t [MCAL_RCC_GetPCLK2Freq](#) (void)
Get the frequency of the APB2 bus clock.
- uint32_t [RCC_GetPLLOutputClock](#) (void)
Get the frequency of the PLL (Phase-Locked Loop) output clock.

5.128.1 Detailed Description

5.128.2 Function Documentation

5.128.2.1 MCAL_RCC_Enable_Peripheral()

```
void MCAL_RCC_Enable_Peripheral (
    uint8_t peripheral )
```

Enable the clock for a specific peripheral.

This function enables the clock for the specified peripheral.

Parameters

in	<i>peripheral</i>	Peripheral to enable, as defined in Peripheral Selection .
----	-------------------	--

Returns

None

5.128.2.2 MCAL_RCC_GetHCLKFreq()

```
uint32_t MCAL_RCC_GetHCLKFreq (
    void )
```

Get the frequency of the AHB bus clock.

This function retrieves the frequency of the AHB (Advanced High-performance Bus) clock.

Returns

Frequency of the AHB bus clock in Hertz.

Get the frequency of the AHB bus clock.

This function calculates and returns the frequency of the AHB bus clock (HCLK) in Hz.

Parameters

in	None	
----	------	--

Returns

Frequency of the AHB bus clock in Hz.

5.128.2.3 MCAL_RCC_GetPCLK1Freq()

```
uint32_t MCAL_RCC_GetPCLK1Freq (
    void )
```

Get the frequency of the APB1 bus clock.

This function retrieves the frequency of the APB1 (Advanced Peripheral Bus 1) clock.

Returns

Frequency of the APB1 bus clock in Hertz.

Get the frequency of the APB1 bus clock.

This function calculates and returns the frequency of the APB1 bus clock (PCLK1) in Hz.

Parameters

in	None	
----	------	--

Returns

Frequency of the APB1 bus clock in Hz.

5.128.2.4 MCAL_RCC_GetPCLK2Freq()

```
uint32_t MCAL_RCC_GetPCLK2Freq (
    void )
```

Get the frequency of the APB2 bus clock.

This function retrieves the frequency of the APB2 (Advanced Peripheral Bus 2) clock.

Returns

Frequency of the APB2 bus clock in Hertz.

Get the frequency of the APB2 bus clock.

This function calculates and returns the frequency of the APB2 bus clock (PCLK2) in Hz.

Parameters

in	None	
----	------	--

Returns

Frequency of the APB2 bus clock in Hz.

5.128.2.5 MCAL_RCC_GetSYS_CLKFreq()

```
uint32_t MCAL_RCC_GetSYS_CLKFreq (
    void )
```

Get the frequency of the system clock.

This function retrieves the frequency of the system clock (SYSCLK).

Returns

Frequency of the system clock in Hertz.

Get the frequency of the system clock.

This function calculates and returns the frequency of the system clock (SYSCLK) in Hz.

Parameters

in	None	
----	------	--

Returns

Frequency of the system clock in Hz.

5.128.2.6 MCAL_RCC_Reset_Peripheral()

```
void MCAL_RCC_Reset_Peripheral (
    uint8_t peripheral )
```

Reset a specific peripheral.

This function resets the specified peripheral.

Parameters

in	<i>peripheral</i>	Peripheral to reset, as defined in Peripheral Selection .
----	-------------------	---

Returns

None

5.128.2.7 MCAL_RCC_Select_Clock()

```
void MCAL_RCC_Select_Clock (
    uint8_t clock )
```

Select the clock source for the MCU.

This function configures the clock source used by the microcontroller.

Parameters

in	<i>clock</i>	Clock source to select, as defined in Clock Source Selection .
----	--------------	--

Returns

None

5.128.2.8 RCC_GetPLLOutputClock()

```
uint32_t RCC_GetPLLOutputClock (
    void )
```

Get the frequency of the PLL (Phase-Locked Loop) output clock.

This function calculates and returns the frequency of the PLL output clock.

Returns

Frequency of the PLL output clock in Hertz.

This function calculates and returns the frequency of the PLL output clock.

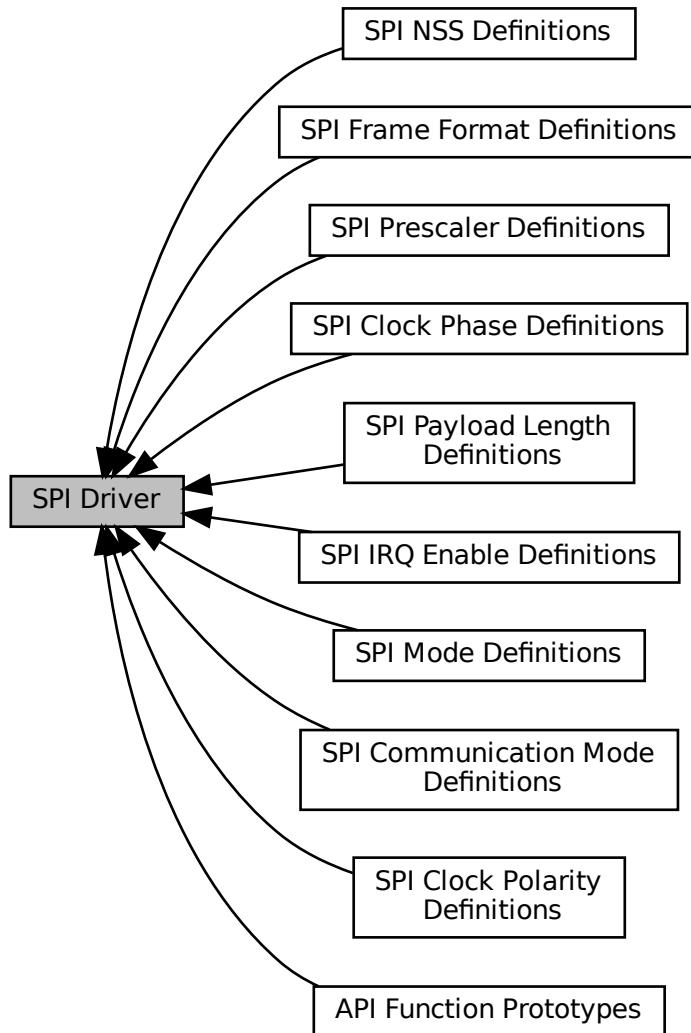
Returns

The PLL output clock frequency in Hertz.

5.129 SPI Driver

SPI driver APIs for STM32F401xx MCU.

Collaboration diagram for SPI Driver:



Modules

- SPI Mode Definitions
- SPI Communication Mode Definitions
- SPI Payload Length Definitions
- SPI Frame Format Definitions
- SPI Clock Polarity Definitions
- SPI Clock Phase Definitions
- SPI NSS Definitions
- SPI Prescaler Definitions
- SPI IRQ Enable Definitions
- API Function Prototypes

Classes

- struct [S_IRQ_SRC](#)
Structure to identify the source of SPI interrupts.
- struct [S_SPI_Config_t](#)
Configuration structure for SPI.

Enumerations

- enum [SPI_PollingMechanism](#) { Enabled , Disabled }
- Enumeration for SPI polling mechanisms.*

5.129.1 Detailed Description

SPI driver APIs for STM32F401xx MCU.

5.129.2 Enumeration Type Documentation

5.129.2.1 SPI_PollingMechanism

enum [SPI_PollingMechanism](#)

Enumeration for SPI polling mechanisms.

Enumerator

Enabled	Polling is enabled
Disabled	Polling is disabled

5.130 SPI Mode Definitions

Collaboration diagram for SPI Mode Definitions:



Macros

- `#define SPI_Mode_Master (0x1<<2)`
- `#define SPI_Mode_Slave (0x00000000U)`

5.130.1 Detailed Description

5.130.2 Macro Definition Documentation

5.130.2.1 SPI_Mode_Master

```
#define SPI_Mode_Master (0x1<<2)
```

SPI master mode

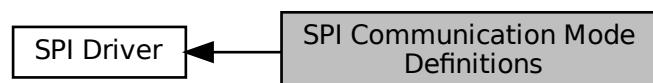
5.130.2.2 SPI_Mode_Slave

```
#define SPI_Mode_Slave (0x00000000U)
```

SPI slave mode

5.131 SPI Communication Mode Definitions

Collaboration diagram for SPI Communication Mode Definitions:



Macros

- #define SPI_Direction_2lines (0x00000000U)
- #define SPI_Direction_2lines_RX_Only (0x1<<10)
- #define SPI_Direction_1line_Receive_only (0x1<<15)
- #define SPI_Direction_1line_Transmit_only ((0x1<<15)|(0x1<<14))

5.131.1 Detailed Description

5.131.2 Macro Definition Documentation

5.131.2.1 SPI_Direction_1line_Receive_only

```
#define SPI_Direction_1line_Receive_only (0x1<<15)
```

1-line receive only mode

5.131.2.2 SPI_Direction_1line_Transmit_only

```
#define SPI_Direction_1line_Transmit_only ((0x1<<15)|(0x1<<14))
```

1-line transmit only mode

5.131.2.3 SPI_Direction_2lines

```
#define SPI_Direction_2lines (0x00000000U)
```

2-line communication mode

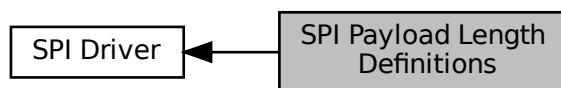
5.131.2.4 SPI_Direction_2lines_RX_Only

```
#define SPI_Direction_2lines_RX_Only (0x1<<10)
```

2-line RX only mode

5.132 SPI Payload Length Definitions

Collaboration diagram for SPI Payload Length Definitions:



Macros

- `#define SPI_Payload_Length_8bit (0x00000000U)`
- `#define SPI_Payload_Length_16bit (0x1<<11)`

5.132.1 Detailed Description

5.132.2 Macro Definition Documentation

5.132.2.1 SPI_Payload_Length_16bit

```
#define SPI_Payload_Length_16bit (0x1<<11)
```

16-bit payload length

5.132.2.2 SPI_Payload_Length_8bit

```
#define SPI_Payload_Length_8bit (0x00000000U)
```

8-bit payload length

5.133 SPI Frame Format Definitions

Collaboration diagram for SPI Frame Format Definitions:



Macros

- `#define SPI_Frame_Format_MSB (0x00000000U)`
- `#define SPI_Frame_Format_LSB (0x1<<7)`

5.133.1 Detailed Description

5.133.2 Macro Definition Documentation

5.133.2.1 SPI_Frame_Format_LSB

```
#define SPI_Frame_Format_LSB (0x1<<7)
```

LSB first frame format

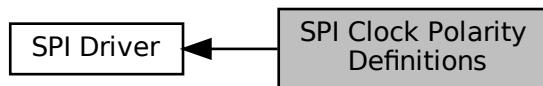
5.133.2.2 SPI_Frame_Format_MSB

```
#define SPI_Frame_Format_MSB (0x00000000U)
```

MSB first frame format

5.134 SPI Clock Polarity Definitions

Collaboration diagram for SPI Clock Polarity Definitions:



Macros

- #define SPI_Clock_Polarity_Low_Idle (0x00000000U)
- #define SPI_Clock_Polarity_High_Idle (0x1<<1)

5.134.1 Detailed Description

5.134.2 Macro Definition Documentation

5.134.2.1 SPI_Clock_Polarity_High_Idle

```
#define SPI_Clock_Polarity_High_Idle (0x1<<1)
```

Clock polarity idle high

5.134.2.2 SPI_Clock_Polarity_Low_Idle

```
#define SPI_Clock_Polarity_Low_Idle (0x00000000U)
```

Clock polarity idle low

5.135 SPI Clock Phase Definitions

Collaboration diagram for SPI Clock Phase Definitions:



Macros

- `#define SPI_Clock_Phase_Leading (0x00000000U)`
- `#define SPI_Clock_Phase_Trailing (0x1)`

5.135.1 Detailed Description

5.135.2 Macro Definition Documentation

5.135.2.1 SPI_Clock_Phase_Leading

```
#define SPI_Clock_Phase_Leading (0x00000000U)
```

Clock phase leading edge

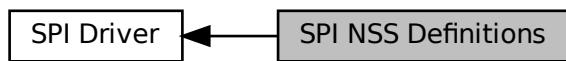
5.135.2.2 SPI_Clock_Phase_Trailing

```
#define SPI_Clock_Phase_Trailing (0x1)
```

Clock phase trailing edge

5.136 SPI NSS Definitions

Collaboration diagram for SPI NSS Definitions:



Macros

- #define SPI_NSS_Hard_Slave (0x00000000U)
- #define SPI_NSS_Hard_Master_SS_Output_Enable (0x1<<2)
- #define SPI_NSS_Hard_Master_SS_No_output (0x00000000U)
- #define SPI_NSS_Internal_Soft_Reset (0x1<<9)
- #define SPI_NSS_Internal_Soft_Set ((0x1<<9)|(0x1<<8))

5.136.1 Detailed Description

5.136.2 Macro Definition Documentation

5.136.2.1 SPI_NSS_Hard_Master_SS_No_output

```
#define SPI_NSS_Hard_Master_SS_No_output (0x00000000U)
```

NSS managed by hardware in master mode with SS output disabled

5.136.2.2 SPI_NSS_Hard_Master_SS_Output_Enable

```
#define SPI_NSS_Hard_Master_SS_Output_Enable (0x1<<2)
```

NSS managed by hardware in master mode with SS output enabled

5.136.2.3 SPI_NSS_Hard_Slave

```
#define SPI_NSS_Hard_Slave (0x00000000U)
```

NSS managed by hardware in slave mode

5.136.2.4 SPI_NSS_Internal_Soft_Reset

```
#define SPI_NSS_Internal_Soft_Reset (0x1<<9)
```

NSS managed by software with internal reset

5.136.2.5 SPI_NSS_Internal_Soft_Set

```
#define SPI_NSS_Internal_Soft_Set ((0x1<<9)|(0x1<<8))
```

NSS managed by software with internal set

5.137 SPI Prescaler Definitions

Collaboration diagram for SPI Prescaler Definitions:



Macros

- #define SPI_Prescaler_By2 (0x00000000U)
- #define SPI_Prescaler_By4 (0b001<<3)
- #define SPI_Prescaler_By8 (0b010<<3)
- #define SPI_Prescaler_By16 (0b011<<3)
- #define SPI_Prescaler_By32 (0b100<<3)
- #define SPI_Prescaler_By64 (0b101<<3)
- #define SPI_Prescaler_By128 (0b110<<3)
- #define SPI_Prescaler_By256 (0b111<<3)

5.137.1 Detailed Description

5.137.2 Macro Definition Documentation

5.137.2.1 SPI_Prescaler_By128

```
#define SPI_Prescaler_By128 (0b110<<3)
```

SPI prescaler divided by 128

5.137.2.2 SPI_Prescaler_By16

```
#define SPI_Prescaler_By16 (0b011<<3)
```

SPI prescaler divided by 16

5.137.2.3 SPI_Prescaler_By2

```
#define SPI_Prescaler_By2 (0x00000000U)
```

SPI prescaler divided by 2

5.137.2.4 SPI_Prescaler_By256

```
#define SPI_Prescaler_By256 (0b111<<3)
```

SPI prescaler divided by 256

5.137.2.5 SPI_Prescaler_By32

```
#define SPI_Prescaler_By32 (0b100<<3)
```

SPI prescaler divided by 32

5.137.2.6 SPI_Prescaler_By4

```
#define SPI_Prescaler_By4 (0b001<<3)
```

SPI prescaler divided by 4

5.137.2.7 SPI_Prescaler_By64

```
#define SPI_Prescaler_By64 (0b101<<3)
```

SPI prescaler divided by 64

5.137.2.8 SPI_Prescaler_By8

```
#define SPI_Prescaler_By8 (0b010<<3)
```

SPI prescaler divided by 8

5.138 SPI IRQ Enable Definitions

Collaboration diagram for SPI IRQ Enable Definitions:



Macros

- #define SPI_IRQ_Enable_None (uint32_t)(0)
- #define SPI_IRQ_Enable_Tx_Only (uint32_t)(1<<7)
- #define SPI_IRQ_Enable_Rx_Only (uint32_t)(1<<6)
- #define SPI_IRQ_Enable_Err (uint32_t)(1<<5)

5.138.1 Detailed Description

5.138.2 Macro Definition Documentation

5.138.2.1 SPI_IRQ_Enable_Err

```
#define SPI_IRQ_Enable_Err (uint32_t) (1<<5)
```

Enable Error interrupt

5.138.2.2 SPI_IRQ_Enable_None

```
#define SPI_IRQ_Enable_None (uint32_t) (0)
```

No IRQ enabled

5.138.2.3 SPI_IRQ_Enable_Rx_Only

```
#define SPI_IRQ_Enable_Rx_Only (uint32_t) (1<<6)
```

Enable RX interrupt

5.138.2.4 SPI_IRQ_Enable_Tx_Only

```
#define SPI_IRQ_Enable_Tx_Only (uint32_t)(1<<7)
```

Enable TX interrupt

5.139 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void [MCAL_SPI_Init](#) (SPI_TypeDef *SPIx, [S_SPI_Config_t](#) *Config)
Initializes the SPI peripheral according to the specified configuration.
- void [MCAL_SPI_DeInit](#) (SPI_TypeDef *SPIx)
Deinitializes the SPI peripheral.
- void [MCAL_SPI_GPIO_Set_Pins](#) (SPI_TypeDef *SPIx)
Configures the GPIO pins used for SPI communication.
- void [MCAL_SPI_SendData](#) (SPI_TypeDef *SPIx, uint16_t *pTXBuffer, enum [SPI_PollingMechanism](#) Polling_En)
Sends data through SPI.
- void [MCAL_SPI_ReceiveData](#) (SPI_TypeDef *SPIx, uint16_t *pRXBuffer, enum [SPI_PollingMechanism](#) Polling_En)
Receives data from SPI.
- void [MCAL_SPI_Tx_Rx](#) (SPI_TypeDef *SPIx, uint16_t *TX_RX_pBuffer, enum [SPI_PollingMechanism](#) Polling_En)
Transmits and receives data through SPI.

5.139.1 Detailed Description

5.139.2 Function Documentation

5.139.2.1 MCAL_SPI_DeInit()

```
void MCAL_SPI_DeInit (
    SPI_TypeDef * SPIx )
```

Deinitializes the SPI peripheral.

Parameters

<i>SPIx</i>	Pointer to the SPI peripheral register base address.
-------------	--

This function resets the SPI peripheral and disables its interrupt.

Parameters

in	<i>SPIx</i>	Pointer to the SPI peripheral (SPI1 or SPI2) to deinitialize.
----	-------------	---

Return values

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

Note

This function disables the SPI peripheral clock and IRQ.

5.139.2.2 MCAL_SPI_GPIO_Set_Pins()

```
void MCAL_SPI_GPIO_Set_Pins (   
    SPI_TypeDef * SPIx )
```

Configures the GPIO pins used for SPI communication.

Parameters

<i>SPIx</i>	Pointer to the SPI peripheral register base address.
-------------	--

Configures the GPIO pins used for SPI communication.

This function configures the necessary GPIO pins for SPI communication (NSS, SCK, MISO, MOSI) according to the selected SPI peripheral (SPI1 or SPI2) and the device mode (Master or Slave).

Parameters

in	<i>SPIx</i>	Pointer to the SPI peripheral (SPI1 or SPI2).
----	-------------	---

Note

This function sets the alternate function for the pins used in SPI communication.

Return values

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

5.139.2.3 MCAL_SPI_Init()

```
void MCAL_SPI_Init (
    SPI_TypeDef * SPIx,
    S_SPI_Config_t * Config )
```

Initializes the SPI peripheral according to the specified configuration.

Parameters

<i>SPIx</i>	Pointer to the SPI peripheral register base address.
<i>Config</i>	Pointer to the SPI configuration structure.

Initializes the SPI peripheral according to the specified configuration.

This function configures the SPI peripheral according to the provided configuration.

Parameters

in	<i>SPIx</i>	Pointer to the SPI peripheral (SPI1 or SPI2) to initialize.
in	<i>Config</i>	Pointer to the SPI configuration structure.

Return values

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

Note

This function enables the SPI peripheral clock and configures the SPI registers according to the specified settings.

```
< Temporary register for SPI CR1 configuration
< Temporary register for SPI CR2 configuration
< Enable SPI bit in CR1
< Set device mode (Master/Slave)
< Set communication mode (Full Duplex/Transmit/Receive)
< Set frame format (MSB/LSB)
< Set payload length (8-bit/16-bit)
< Set clock polarity
< Set clock phase
< Set SPI prescaler
```

5.139.2.4 MCAL_SPI_ReceiveData()

```
void MCAL_SPI_ReceiveData (
    SPI_TypeDef * SPIx,
    uint16_t * pRXBuffer,
    enum SPI_PollingMechanism Polling_En )
```

Receives data from SPI.

Parameters

<i>SPIx</i>	Pointer to the SPI peripheral register base address.
<i>pRXBuffer</i>	Pointer to the buffer where received data will be stored.
<i>Polling_En</i>	Specifies whether polling is enabled or disabled.

Receives data from SPI.

This function receives data from the SPI peripheral. It can operate in polling mode or non-polling mode.

Parameters

in	<i>SPIx</i>	Pointer to the SPI peripheral (SPI1 or SPI2) to use for reception.
in	<i>pRXBuffer</i>	Pointer to the buffer where received data will be stored.
in	<i>Polling_En</i>	Indicates whether polling is enabled or not.

Return values

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

Note

In polling mode, the function waits until the RXNE flag is set before reading data from the data register.

< Wait until RXNE flag is set

< Receive data

5.139.2.5 MCAL_SPI_SendData()

```
void MCAL_SPI_SendData (
    SPI_TypeDef * SPIx,
    uint16_t * pTXBuffer,
    enum SPI_PollingMechanism Polling_En )
```

Sends data through SPI.

Parameters

<i>SPIx</i>	Pointer to the SPI peripheral register base address.
<i>pTXBuffer</i>	Pointer to the buffer containing the data to be transmitted.
<i>Polling_En</i>	Specifies whether polling is enabled or disabled.

Sends data through SPI.

This function transmits data via the SPI peripheral. It can operate in polling mode or non-polling mode.

Parameters

in	<i>SPIx</i>	Pointer to the SPI peripheral (SPI1 or SPI2) to use for transmission.
in	<i>pTXBuffer</i>	Pointer to the data buffer to transmit.
in	<i>Polling_En</i>	Indicates whether polling is enabled or not.

Return values

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

Note

In polling mode, the function waits until the TXE flag is set before writing data to the data register.

< Wait until TXE flag is set

< Send data

5.139.2.6 MCAL_SPI_Tx_Rx()

```
void MCAL_SPI_Tx_Rx (
    SPI_TypeDef * SPIx,
    uint16_t * TX_RX_pBuffer,
    enum SPI_PollingMechanism Polling_En )
```

Transmits and receives data through SPI.

Parameters

<i>SPIx</i>	Pointer to the SPI peripheral register base address.
<i>TX_RX_pBuffer</i>	Pointer to the buffer containing data to be transmitted and where received data will be stored.
<i>Polling_En</i>	Specifies whether polling is enabled or disabled.

Transmits and receives data through SPI.

This function handles the transmission and reception of data through the specified SPI peripheral. It writes data to the SPI data register and waits for the transmission and reception to complete.

Parameters

in	<i>SPIx</i>	Pointer to the SPI peripheral (SPI1 or SPI2).
in, out	<i>TX_RX_pBuffer</i>	Pointer to the data buffer for transmission and reception.
in	<i>Polling_En</i>	Specifies whether polling should be used (Enabled/Disabled).

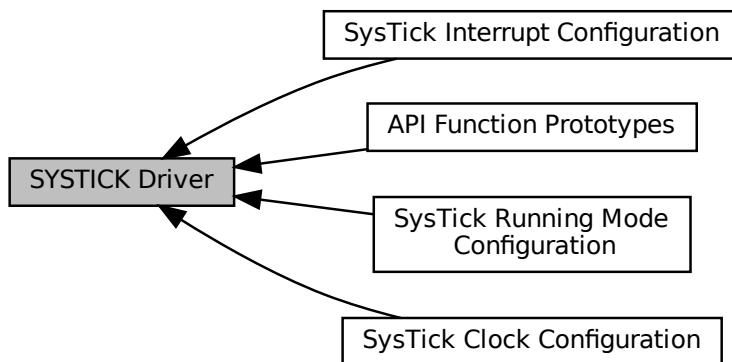
Return values

None

5.140 SYSTICK Driver

SYSTICK driver APIs for STM32F401xx MCU.

Collaboration diagram for SYSTICK Driver:



Modules

- SysTick Interrupt Configuration
- SysTick Clock Configuration
- SysTick Running Mode Configuration
- API Function Prototypes

Classes

- struct [STK_config_t](#)
SysTick Configuration Structure Definition.

Macros

- #define [STK_INTERRUPT_MASK](#) 0x02UL
Interrupt enable/disable mask for the SysTick timer.
- #define [STK_CLK_MASK](#) 0x04UL
Clock source mask for the SysTick timer.
- #define [STK_RELOAD_MASK](#) 0x00FFFFFFUL
Reload value mask for the SysTick timer.
- #define [STK_FCPU](#) 80000000UL
CPU frequency definition for the SysTick timer.

5.140.1 Detailed Description

SYSTICK driver APIs for STM32F401xx MCU.

5.140.2 Macro Definition Documentation

5.140.2.1 STK_FCPU

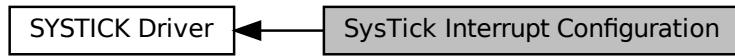
```
#define STK_FCPU 80000000UL
```

CPU frequency definition for the SysTick timer.

8 MHz CPU frequency

5.141 SysTick Interrupt Configuration

Collaboration diagram for SysTick Interrupt Configuration:



Macros

- #define STK_INTERRUPT_ENABLED 0x02UL
- #define STK_INTERRUPT_DISABLED 0x00UL

5.141.1 Detailed Description

5.141.2 Macro Definition Documentation

5.141.2.1 STK_INTERRUPT_DISABLED

```
#define STK_INTERRUPT_DISABLED 0x00UL
```

Disable SysTick interrupt

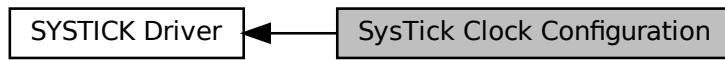
5.141.2.2 STK_INTERRUPT_ENABLED

```
#define STK_INTERRUPT_ENABLED 0x02UL
```

Enable SysTick interrupt

5.142 SysTick Clock Configuration

Collaboration diagram for SysTick Clock Configuration:



Macros

- `#define STK_CLK_AHB 0x04UL`
- `#define STK_CLK_AHB_8 0x00UL`

5.142.1 Detailed Description

5.142.2 Macro Definition Documentation

5.142.2.1 STK_CLK_AHB

```
#define STK_CLK_AHB 0x04UL
```

SysTick clock sourced from AHB

5.142.2.2 STK_CLK_AHB_8

```
#define STK_CLK_AHB_8 0x00UL
```

SysTick clock sourced from AHB/8

5.143 SysTick Running Mode Configuration

Collaboration diagram for SysTick Running Mode Configuration:



Macros

- `#define STK_PERIODIC_MODE 0x01U`
- `#define STK_ONE_SHOT_MODE 0x00U`

5.143.1 Detailed Description

5.143.2 Macro Definition Documentation

5.143.2.1 STK_ONE_SHOT_MODE

```
#define STK_ONE_SHOT_MODE 0x00U
```

One-shot mode for SysTick

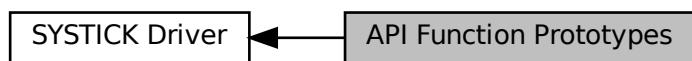
5.143.2.2 STK_PERIODIC_MODE

```
#define STK_PERIODIC_MODE 0x01U
```

Periodic mode for SysTick

5.144 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void **MCAL_STK_Config** (**STK_config_t** *_cfg)
Configures the SysTick timer with the specified parameters.
- void **MCAL_STK_SetReload** (**uint32** value)
Sets the reload value for the SysTick timer.
- void **MCAL_STK_SetCallback** (**void(*pfCallback)(void)**)
Sets the callback function for the SysTick timer interrupt.
- void **MCAL_STK_StartTimer** ()
Starts the SysTick timer.
- void **MCAL_STK_StopTimer** ()
Stops the SysTick timer.
- void **MCAL_STK_Delay** (**uint32** delay_ticks)
Delays the system by a specified number of ticks using the SysTick timer.
- void **MCAL_STK_Delay1ms** (**uint32** delay_ms)
Delays the system by a specified number of milliseconds.

5.144.1 Detailed Description

5.144.2 Function Documentation

5.144.2.1 MCAL_STK_Config()

```
void MCAL_STK_Config (
    STK_config_t * _cfg )
```

Configures the SysTick timer with the specified parameters.

This function configures the SysTick timer's clock source, interrupt settings, and reload value based on the input configuration structure.

Parameters

in	_cfg	Pointer to the configuration structure containing the SysTick settings.
----	------	---

Returns

None

Note

The SysTick timer is stopped when this function is called. It must be re-enabled manually.

Configures the SysTick timer with the specified parameters.

This function initializes the SysTick timer based on the provided configuration. It sets up the clock source, interrupt enable status, reload value, and callback function if interrupts are enabled.

Parameters

in	_cfg	Pointer to a struct containing SysTick configuration parameters.
----	------	--

Returns

None

Note

This function stops the timer; you need to re-enable it after configuration.

5.144.2.2 MCAL_STK_Delay()

```
void MCAL_STK_Delay (
    uint32 delay_ticks )
```

Delays the system by a specified number of ticks using the SysTick timer.

This function uses the SysTick timer to generate a delay for a specified number of ticks. After the delay, the timer configuration and reload value are restored.

Parameters

in	delay_ticks	The number of ticks to delay.
----	-------------	-------------------------------

Returns

None

Delays the system by a specified number of ticks using the SysTick timer.

This function generates a delay based on the number of SysTick timer ticks specified. It temporarily disables the SysTick timer, sets the reload value, and waits for the timer to expire before restoring the previous configuration.

Parameters

in	delay_ticks	Number of ticks to delay.
----	-------------	---------------------------

Returns

None

5.144.2.3 MCAL_STK_Delay1ms()

```
void MCAL_STK_Delay1ms (
    uint32 delay_ms )
```

Delays the system by a specified number of milliseconds.

This function generates a delay for the specified number of milliseconds using the SysTick timer. The user must define the system frequency (stk_cpu_freq_define) to ensure accurate delays.

Parameters

in	<i>delay_ms</i>	The number of milliseconds to delay.
----	-----------------	--------------------------------------

Returns

None

Delays the system by a specified number of milliseconds.

This function generates a delay of the specified number of milliseconds by repeatedly calling MCAL_STK_Delay for each millisecond.

Parameters

in	<i>delay_ms</i>	Number of milliseconds to delay.
----	-----------------	----------------------------------

Returns

None

Note

The user must define the frequency of the SysTick timer in stk_cpu_freq_define.

5.144.2.4 MCAL_STK_SetCallback()

```
void MCAL_STK_SetCallback (
    void(*)(void) pfCallback )
```

Sets the callback function for the SysTick timer interrupt.

This function assigns a user-defined callback function that will be executed when the SysTick interrupt is triggered.

Parameters

in	<i>pfCallback</i>	Pointer to the callback function to be executed on SysTick interrupt.
----	-------------------	---

Returns

None

This function allows the user to specify a function that will be called when the SysTick timer interrupt occurs.

Parameters

in	<i>pfCallback</i>	Pointer to the callback function.
----	-------------------	-----------------------------------

Returns

None

5.144.2.5 MCAL_STK_SetReload()

```
void MCAL_STK_SetReload (
    uint32 value )
```

Sets the reload value for the SysTick timer.

This function sets the reload value which determines the SysTick timer's period.

Parameters

in	<i>value</i>	The reload value to be set.
----	--------------	-----------------------------

Returns

None

Sets the reload value for the SysTick timer.

This function sets the value that the SysTick timer counts down from before generating an interrupt.

Parameters

in	<i>value</i>	The value to be set as the reload value.
----	--------------	--

Returns

None

5.144.2.6 MCAL_STK_StartTimer()

```
void MCAL_STK_StartTimer ( )
```

Starts the SysTick timer.

This function starts the SysTick timer. The timer must be configured using [MCAL_STK_Config\(\)](#) before starting it unless default values are used.

Returns

None

This function initializes and starts the SysTick timer. The timer will begin counting from the reload value.

Parameters

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

Returns

None

Note

`MCAL_STK_Config` should be called first to configure the SysTick timer unless default values are acceptable.

5.144.2.7 MCAL_STK_StopTimer()

```
void MCAL_STK_StopTimer ( )
```

Stops the SysTick timer.

This function stops the SysTick timer, halting its operation until restarted.

Returns

None

This function stops the SysTick timer, halting the countdown and disabling interrupts.

Parameters

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

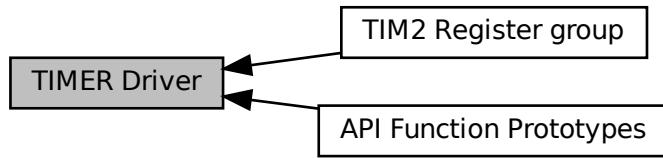
Returns

None

5.145 TIMER Driver

TIMER driver APIs for STM32F401xx MCU.

Collaboration diagram for TIMER Driver:



Modules

- [TIM2 Register group](#)
TIM2 Register group.
- [API Function Prototypes](#)

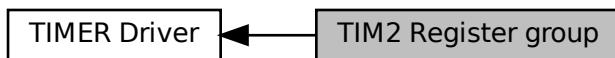
5.145.1 Detailed Description

TIMER driver APIs for STM32F401xx MCU.

5.146 TIM2 Register group

TIM2 Register group.

Collaboration diagram for TIM2 Register group:



Macros

- `#define RCC_APB1ENR *(volatile uint32 *) (RCC_BASE + 0x40)`
Enable the APB1 peripheral clock for TIM2.
- `#define RCC_APB2ENR *(volatile uint32 *) (RCC_BASE + 0x44)`
Enable the APB2 peripheral clock for TIM2.
- `#define TIM2_TIMER_BASE 0x40000000`
TIM2_Register TIM2 Register Base Addresses and Offsets.

- `#define TIM2_CR1 *(volatile uint32 *) (TIM2_TIMER_BASE + 0x00)`
Control register 1 (CR1) for TIM2.
- `#define TIM2_DIER *(volatile uint32 *) (TIM2_TIMER_BASE + 0x0C)`
DMA/interrupt enable register (DIER) for TIM2.
- `#define TIM2_SR *(volatile uint32 *) (TIM2_TIMER_BASE + 0x10)`
Status register (SR) for TIM2.
- `#define TIM2_CNT *(volatile uint32 *) (TIM2_TIMER_BASE + 0x24)`
Counter register (CNT) for TIM2.
- `#define TIM2_PSC *(volatile uint32 *) (TIM2_TIMER_BASE + 0x28)`
Prescaler register (PSC) for TIM2.
- `#define TIM2_ARR *(volatile uint32 *) (TIM2_TIMER_BASE + 0x2C)`
Auto-reload register (ARR) for TIM2.

5.146.1 Detailed Description

TIM2 Register group.

5.146.2 Macro Definition Documentation

5.146.2.1 TIM2_TIMER_BASE

```
#define TIM2_TIMER_BASE 0x40000000
```

TIM2_Register TIM2 Register Base Addresses and Offsets.

Base address of TIM2

5.147 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void `Timer2_init` (void)
Initialize Timer 2 (TIM2) for basic timing functionalities.
- void `dus` (int us)
Delay execution for a specified number of microseconds.
- void `dms` (int ms)
Delay execution for a specified number of milliseconds.

5.147.1 Detailed Description

5.147.2 Function Documentation

5.147.2.1 dms()

```
void dms (
    int ms )
```

Delay execution for a specified number of milliseconds.

This function uses TIM2 to generate a delay in milliseconds.

Parameters

<code>ms</code>	Number of milliseconds to delay.
-----------------	----------------------------------

Delay execution for a specified number of milliseconds.

This function generates a delay by repeatedly calling the `dus` function for each millisecond required. It effectively creates a delay by generating multiple microsecond delays.

Parameters

<code>in</code>	<code>ms</code>	Number of milliseconds to delay.
-----------------	-----------------	----------------------------------

Returns

None

5.147.2.2 dus()

```
void dus (
    int us )
```

Delay execution for a specified number of microseconds.

This function uses TIM2 to generate a delay in microseconds.

Parameters

<i>us</i>	Number of microseconds to delay.
-----------	----------------------------------

Delay execution for a specified number of microseconds.

This function generates a delay by counting timer ticks for the specified number of microseconds using Timer2. The timer is reset and counts up until the desired delay duration is reached.

Parameters

<i>in</i>	<i>us</i>	Number of microseconds to delay.
-----------	-----------	----------------------------------

Returns

None

5.147.2.3 Timer2_init()

```
void Timer2_init (
    void )
```

Initialize Timer 2 (TIM2) for basic timing functionalities.

This function configures the TIM2 peripheral with the necessary prescaler and auto-reload values, enabling it to function as a basic timer for timing and delay applications.

Initialize Timer 2 (TIM2) for basic timing functionalities.

This function configures Timer2 with a prescaler and auto-reload register to generate interrupts at a specific interval. The timer is set to produce a delay of 50 milliseconds based on the configured settings.

Parameters

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

Returns

None

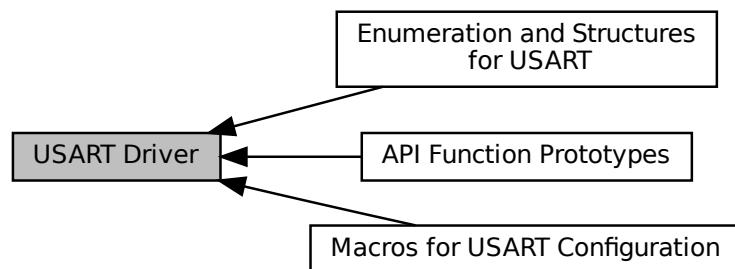
Note

The function enables the clock for Timer2, sets the prescaler to achieve a 1 MHz clock input, and configures the auto-reload register for a delay of 50 milliseconds. The timer is started, and the function waits until the timer's update interrupt flag is set.

5.148 USART Driver

USART peripheral driver for STM32F401 microcontrollers.

Collaboration diagram for USART Driver:



Modules

- Enumeration and Structures for USART
- Macros for USART Configuration
- API Function Prototypes

Classes

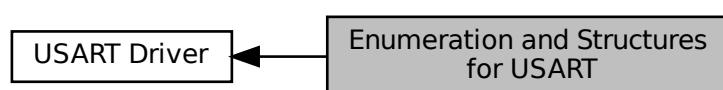
- struct [USART_cfg_t](#)
Configuration structure for USART (Universal Synchronous Asynchronous Receiver Transmitter) peripheral.

5.148.1 Detailed Description

USART peripheral driver for STM32F401 microcontrollers.

5.149 Enumeration and Structures for USART

Collaboration diagram for Enumeration and Structures for USART:



Enumerations

- enum **Polling_Mechanism** { **enable** , **disable** }
- Enumeration for Polling Mechanism.*

5.149.1 Detailed Description

5.150 Macros for USART Configuration

Collaboration diagram for Macros for USART Configuration:



Macros

- #define **UART_Mode_RX** ((uint32_t)(1UL << 2))
USART Mode Definitions.
- #define **UART_Mode_TX** ((uint32_t)(1UL << 3))
- #define **UART_Mode_TX_RX** ((uint32_t)(1UL << 2 | 1UL << 3))
- #define **UART_BaudRate_2400** 2400
Baud Rate Definitions.
- #define **UART_BaudRate_9600** 9600
- #define **UART_BaudRate_19200** 19200
- #define **UART_BaudRate_57600** 57600
- #define **UART_BaudRate_115200** 115200
- #define **UART_BaudRate_230400** 230400
- #define **UART_BaudRate_460800** 460800
- #define **UART_BaudRate_921600** 921600
- #define **UART_BaudRate_2250000** 2250000
- #define **UART_BaudRate_4500000** 4500000
- #define **UART_Payload_Length_8B** ((uint32_t)(0))
Payload Length Definitions.
- #define **UART_Payload_Length_9B** ((uint32_t)(1UL << 12))
- #define **UART_Parity_NONE** ((uint32_t)(0))
Parity Definitions.
- #define **UART_Parity_EVEN** ((uint32_t)(1UL << 10))
- #define **UART_Parity_ODD** ((uint32_t)(1UL << 10) | (1UL << 9))
- #define **UART_StopBits_half** ((uint32_t)(1 << 12))
Stop Bits Definitions.
- #define **UART_StopBits_1** ((uint32_t)(0))
- #define **UART_StopBits_1_half** ((uint32_t)(3 << 12))
- #define **UART_StopBits_2** ((uint32_t)(2 << 12))
- #define **UART_HwFlowCtl_NONE** ((uint32_t)(0))

- Hardware Flow Control Definitions.*
- #define **UART_HwFlowCtl_RTS** ((uint32_t)(1UL << 8))
 - #define **UART_HwFlowCtl_CTS** ((uint32_t)(1UL << 9))
 - #define **UART_HwFlowCtl_RTS_CTS** ((uint32_t)(1UL << 8 | 1UL << 9))
 - #define **UART_IRQ_Enable_NONE** ((uint32_t)(0))
- IRQ Enable Definitions.*
- #define **UART_IRQ_Enable_TXE** ((uint32_t)(1UL << 7))
 - #define **UART_IRQ_Enable_TC** ((uint32_t)(1UL << 6))
 - #define **UART_IRQ_Enable_RXNE** ((uint32_t)(1UL << 5))
 - #define **UART_IRQ_Enable_PE** ((uint32_t)(1UL << 8))

5.150.1 Detailed Description

5.151 API Function Prototypes

Collaboration diagram for API Function Prototypes:



Functions

- void **MCAL_USART_Init** (USART_TypeDef *USARTx, **USART_cfg_t** *USART_cfg)

Initializes UART (Asynchronous mode only).
- void **MCAL_USART_DeInit** (USART_TypeDef *USARTx)

Deinitializes UART (Asynchronous mode only).
- void **MCAL_USART_GPIO_Set_Pins** (USART_TypeDef *USARTx)

Initializes GPIO pins for USART.
- void **MCAL_USART_SendData** (USART_TypeDef *USARTx, uint16_t *pTxBuffer, **Polling_Mechanism** PollingEn)

Sends buffer via USART.
- void **MCAL_USART_SendString** (USART_TypeDef *USARTx, uint8_t *str, uint8_t str_len)

Sends a string via USART.
- void **MCAL_USART_ReceiveData** (USART_TypeDef *USARTx, uint16_t *pRxBuffer, **Polling_Mechanism** PollingEn)

Receives buffer from USART.
- void **MCAL_USART_ReceiveBuffer** (USART_TypeDef *USARTx, uint16_t *pRxBuffer, uint8_t length)

Receives buffer from USART with specified length.
- void **MCAL_USART_Wait_TC** (USART_TypeDef *USARTx)

Waits until transmission is completed by polling on TC flag.

5.151.1 Detailed Description

5.151.2 Function Documentation

5.151.2.1 MCAL_USART_DeInit()

```
void MCAL_USART_DeInit (
    USART_TypeDef * USARTx )
```

Deinitializes UART (Asynchronous mode only).

Parameters

<i>USARTx</i>	Pointer to the USART peripheral instance.
---------------	---

Deinitializes UART (Asynchronous mode only).

This function resets the USART peripheral and disables its associated NVIC interrupt.

Parameters

in	<i>USARTx</i>	Pointer to the USART peripheral instance (USART1, USART2, or USART6).
----	---------------	---

Returns

None

5.151.2.2 MCAL_USART_GPIO_Set_Pins()

```
void MCAL_USART_GPIO_Set_Pins (
    USART_TypeDef * USARTx )
```

Initializes GPIO pins for UART.

Parameters

<i>USARTx</i>	Pointer to the USART peripheral instance.
---------------	---

Initializes GPIO pins for UART.

This function sets up the GPIO pins used by the USART peripheral, including TX, RX, and optional CTS/RTS pins, depending on the hardware flow control settings.

Parameters

in	<i>USARTx</i>	Pointer to the USART peripheral instance (USART1, USART2, or USART6).
----	---------------	---

Returns

None

NoteCalled automatically at the end of [MCAL_USART_Init\(\)](#).**5.151.2.3 MCAL_USART_Init()**

```
void MCAL_USART_Init (
    USART_TypeDef * USARTx,
    USART_cfg_t * USART_cfg )
```

Initializes UART (Asynchronous mode only).

Parameters

<i>USARTx</i>	Pointer to the USART peripheral instance.
<i>USART_cfg</i>	Pointer to the USART configuration.

Initializes UART (Asynchronous mode only).

This function configures the USART peripheral based on the settings provided in the USART configuration structure. It supports asynchronous mode and sets up the necessary GPIO pins.

Parameters

in	<i>USARTx</i>	Pointer to the USART peripheral instance (USART1, USART2, or USART6).
in	<i>USART_cfg</i>	Pointer to the USART configuration structure.

Returns

None

Note

Supports asynchronous mode only with an 8MHz clock.

5.151.2.4 MCAL_USART_ReceiveBuffer()

```
void MCAL_USART_ReceiveBuffer (
    USART_TypeDef * USARTx,
    uint16_t * pRxBUFFER,
    uint8_t length )
```

Receives buffer from UART with specified length.

Parameters

<i>USARTx</i>	Pointer to the USART peripheral instance.
<i>pRxBUFFER</i>	Buffer to be received.
<i>length</i>	Length of data to be received (0 = until receiving '\r').

5.151.2.5 MCAL_USART_ReceiveData()

```
void MCAL_USART_ReceiveData (
    USART_TypeDef * USARTx,
    uint16_t * pRxBUFFER,
    Polling_Mechanism PollingEn )
```

Receives buffer from UART.

Parameters

<i>USARTx</i>	Pointer to the USART peripheral instance.
<i>pRxBUFFER</i>	Buffer to be received.
<i>PollingEn</i>	Enable or disable polling mechanism.

5.151.2.6 MCAL_USART_SendData()

```
void MCAL_USART_SendData (
    USART_TypeDef * USARTx,
    uint16_t * pTxBuffer,
    Polling_Mechanism PollingEn )
```

Sends buffer via UART.

Parameters

<i>USARTx</i>	Pointer to the USART peripheral instance.
<i>pTxBuffer</i>	Buffer to be transmitted.
<i>PollingEn</i>	Enable or disable polling mechanism.

5.151.2.7 MCAL_USART_SendString()

```
void MCAL_USART_SendString (
    USART_TypeDef * USARTx,
    uint8_t * str,
    uint8_t str_len )
```

Sends a string via UART.

Parameters

<i>USARTx</i>	Pointer to the USART peripheral instance.
<i>str</i>	Pointer to the string to be transmitted.
<i>str_len</i>	Length of string to be transmitted (0 = send until null '\0').

5.151.2.8 MCAL_USART_Wait_TC()

```
void MCAL_USART_Wait_TC (
    USART_TypeDef * USARTx )
```

Waits until transmission is completed by polling on TC flag.

Parameters

<i>USARTx</i>	Pointer to the USART peripheral instance.
---------------	---

Waits until transmission is completed by polling on TC flag.

Parameters

in	<i>USARTx</i>	Pointer to the USART peripheral instance. It can be one of the following: USART1, USART2, or USART6.
----	---------------	--

Returns

None

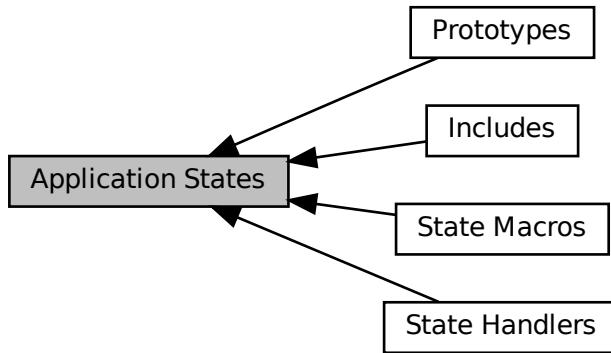
Note

This function continuously polls the TC flag in the SR (Status Register) until the flag is set, indicating that the transmission is complete.

5.152 Application States

Definitions and APIs for managing application states.

Collaboration diagram for Application States:



Modules

- **Includes**
Required libraries for managing application states.
- **State Macros**
Macros for defining and managing application state handler functions.
- **Prototypes**
Function Prototypes for the Application states APIs.
- **State Handlers**
Function prototypes for handling different application states.

Enumerations

- enum STATES {
 Init_STATE , Admin_STATE , Idle_STATE , Enter_Gate_STATE ,
 Exit_Gate_STATE , Full_STATE }
Enum representing various application states.

Functions

- **STATE_API (Init_STATE)**
Initializes the system and sets up the initial state.
- **STATE_API (Admin_STATE)**
Admin state to initialize the user IDs and prepare the system.
- **STATE_API (Idle_STATE)**
Idle state where the system waits for user input (entry/exit requests).
- **STATE_API (Enter_Gate_STATE)**
Handles the logic for opening the entry gate.
- **STATE_API (Exit_Gate_STATE)**
Handles the logic for opening the exit gate.
- **STATE_API (Full_STATE)**
Full state where the parking lot is full, and no entry is allowed.

Variables

- volatile uint8 Enter_Flag
Flags and variables used for managing system state.
- volatile uint8 Exit_Flag
Flags for entry and exit state triggers.
- uint8 Free_Slots
- uint8 Print_Slots_LCD_Flag
Available parking slots and LCD print flag.
- void(* fp_App_State_Handler)() = STATE_NAME(Init_STATE)
Function pointer to manage current state transitions.
- STATES APP_Current_State
Current application state.

5.152.1 Detailed Description

Definitions and APIs for managing application states.

5.152.2 Enumeration Type Documentation

5.152.2.1 STATES

```
enum STATES
```

Enum representing various application states.

< Include header for ECU definitions and functions

This enumeration defines the different states that the application can be in.

Enumerator

Init_STATE	Initialization state
Admin_STATE	Administration state
Idle_STATE	Idle state
Enter_Gate_STATE	State for entering the gate
Exit_Gate_STATE	State for exiting the gate
Full_STATE	Full state

5.152.3 Function Documentation

5.152.3.1 STATE_API() [1/6]

```
STATE_API (
    Admin_STATE )
```

Admin state to initialize the user IDs and prepare the system.

Handler function for the administration state.

5.152.3.2 STATE_API() [2/6]

```
STATE_API (
    Enter_Gate_STATE )
```

Handles the logic for opening the entry gate.

Handler function for the entering gate state.

5.152.3.3 STATE_API() [3/6]

```
STATE_API (
    Exit_Gate_STATE )
```

Handles the logic for opening the exit gate.

Handler function for the exiting gate state.

5.152.3.4 STATE_API() [4/6]

```
STATE_API (
    Full_STATE )
```

Full state where the parking lot is full, and no entry is allowed.

Handler function for the full state.

5.152.3.5 STATE_API() [5/6]

```
STATE_API (
    Idle_STATE )
```

Idle state where the system waits for user input (entry/exit requests).

Handler function for the idle state.

5.152.3.6 STATE_API() [6/6]

```
STATE_API (
    Init_STATE )
```

Initializes the system and sets up the initial state.

Handler function for the initialization state.

5.152.4 Variable Documentation

5.152.4.1 fp_App_State_Handler

```
void(* fp_App_State_Handler) () () = STATE_NAME(Init_STATE)
```

Function pointer to manage current state transitions.

Function pointer for the current application state handler.

Pointer to current state handler function.

5.152.4.2 Free_Slots

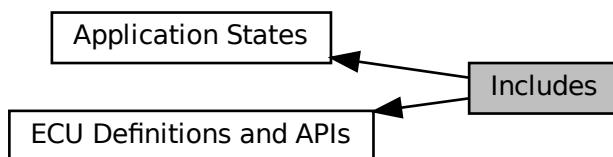
```
uint8 Free_Slots [extern]
```

Number of free parking slots

5.153 Includes

Required libraries for managing application states.

Collaboration diagram for Includes:



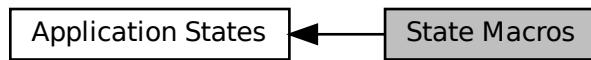
Required libraries for managing application states.

Required libraries for ECU module functionalities.

5.154 State Macros

Macros for defining and managing application state handler functions.

Collaboration diagram for State Macros:



Macros

- `#define STATE_API(_NAME) void ST_##_NAME(void)`
Macro to define a state handler function.
- `#define STATE_NAME(_NAME) ST_##_NAME`
Macro to create a state handler function name.

5.154.1 Detailed Description

Macros for defining and managing application state handler functions.

5.154.2 Macro Definition Documentation

5.154.2.1 STATE_API

```
#define STATE_API(  
    _NAME ) void ST_##_NAME(void)
```

Macro to define a state handler function.

@macro STATE_API

Parameters

<code>_NAME</code>	Name of the state.
--------------------	--------------------

This macro defines a function prototype for the state handler function corresponding to the specified state.

5.154.2.2 STATE_NAME

```
#define STATE_NAME(  
    _NAME ) ST_##_NAME
```

Macro to create a state handler function name.

@macro STATE_NAME

Parameters

<code>_NAME</code>	Name of the state.
--------------------	--------------------

This macro generates the name of the state handler function based on the state name.

5.155 Prototypes

Function Prototypes for the Application states APIs.

Collaboration diagram for Prototypes:



Variables

- `void(* fp_App_State_Handler)()`
Function pointer for the current application state handler.

5.155.1 Detailed Description

Function Prototypes for the Application states APIs.

5.155.2 Variable Documentation

5.155.2.1 fp_App_State_Handler

```
void(* fp_App_State_Handler) () () [extern]
```

Function pointer for the current application state handler.

This pointer is used to reference the handler function corresponding to the current application state. Pointer to the current application state handler function

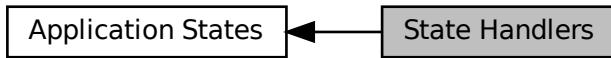
Function pointer for the current application state handler.

Pointer to current state handler function.

5.156 State Handlers

Function prototypes for handling different application states.

Collaboration diagram for State Handlers:



Functions

- [STATE_API \(Init_STATE\)](#)
Handler function for the initialization state.
- [STATE_API \(Admin_STATE\)](#)
Handler function for the administration state.
- [STATE_API \(Idle_STATE\)](#)
Handler function for the idle state.
- [STATE_API \(Enter_Gate_STATE\)](#)
Handler function for the entering gate state.
- [STATE_API \(Exit_Gate_STATE\)](#)
Handler function for the exiting gate state.
- [STATE_API \(Full_STATE\)](#)
Handler function for the full state.

5.156.1 Detailed Description

Function prototypes for handling different application states.

5.156.2 Function Documentation

5.156.2.1 STATE_API() [1/6]

```
STATE_API (
    Admin_STATE )
```

Handler function for the administration state.

Handler function for the administration state.

5.156.2.2 STATE_API() [2/6]

```
STATE_API (
    Enter_Gate_STATE )
```

Handler function for the entering gate state.

Handler function for the entering gate state.

5.156.2.3 STATE_API() [3/6]

```
STATE_API (
    Exit_Gate_STATE )
```

Handler function for the exiting gate state.

Handler function for the exiting gate state.

5.156.2.4 STATE_API() [4/6]

```
STATE_API (
    Full_STATE )
```

Handler function for the full state.

Handler function for the full state.

5.156.2.5 STATE_API() [5/6]

```
STATE_API (
    Idle_STATE )
```

Handler function for the idle state.

Handler function for the idle state.

5.156.2.6 STATE_API() [6/6]

```
STATE_API (
    Init_STATE )
```

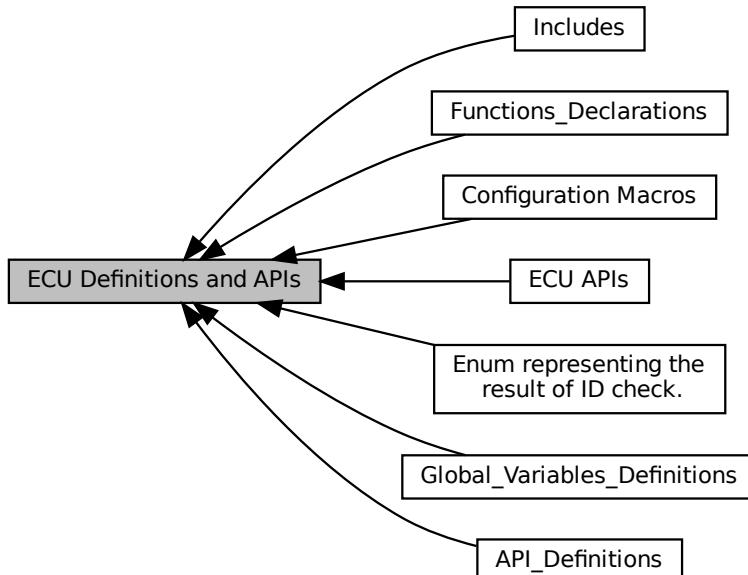
Handler function for the initialization state.

Handler function for the initialization state.

5.157 ECU Definitions and APIs

Definitions and APIs for the ECU (Electronic Control Unit) system.

Collaboration diagram for ECU Definitions and APIs:



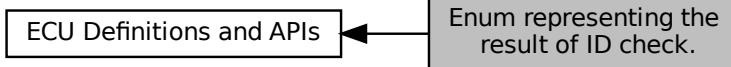
Modules

- **Includes**
Required libraries for managing application states.
- **Enum representing the result of ID check.**
- **Configuration Macros**
Macros for system configuration.
- **ECU APIs**
APIs for managing the ECU functionalities.
- **Functions_Declarations**
Declaration of callback functions for UART operations.
- **Global_Variables_Definitions**
Global variable definitions for LEDs, LCDs, UART, and system states.
- **API_Definitions**
API definitions for the ECU functionalities.

5.157.1 Detailed Description

Definitions and APIs for the ECU (Electronic Control Unit) system.

5.158 Enum representing the result of ID check.

Collaboration diagram for Enum representing the result of ID check.:


Enumerations

- enum `ID_Check_Result { ID_NOT_Found , ID_Found }`

5.158.1 Detailed Description

This enumeration defines possible outcomes of checking an ID in the system.

5.158.2 Enumeration Type Documentation

5.158.2.1 `ID_Check_Result`

enum `ID_Check_Result`

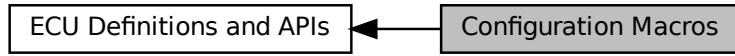
Enumerator

<code>ID_NOT_Found</code>	ID was not found
<code>ID_Found</code>	ID was found

5.159 Configuration Macros

Macros for system configuration.

Collaboration diagram for Configuration Macros:



Macros

- `#define NO_OF_SLOTS 3`
Number of parking slots.
- `#define USERS_COUNT 3`
Total number of users.
- `#define USER1 0`
User 1 ID.
- `#define USER2 1`
User 2 ID.
- `#define USER3 2`
User 3 ID.
- `#define ENTER_USART_INSTANT USART1`
Enter gate USART instance.
- `#define EXIT_USART_INSTANT USART2`
Exit gate USART instance.
- `#define ENTER_PIR_PORT GPIOA`
Enter gate PIR sensor GPIO port.
- `#define ENTER_PIR_PIN GPIO_PIN_7`
Enter gate PIR sensor GPIO pin.
- `#define EXIT_PIR_PORT GPIOA`
Exit gate PIR sensor GPIO port.
- `#define EXIT_PIR_PIN GPIO_PIN_1`
Exit gate PIR sensor GPIO pin.

5.159.1 Detailed Description

Macros for system configuration.

5.160 ECU APIs

APIs for managing the ECU functionalities.

Collaboration diagram for ECU APIs:



Functions

- void [ECU_Init](#) (void)
Initializes all hardware components and peripherals.
- void [Admin_Init](#) (void)
Initializes the admin settings, including setting up user IDs.
- void [UserLCD_PrintFreeSlots](#) (void)
Prints the number of free parking slots on the LCD.
- [ID_Check_Result Check_ID](#) (uint8 _ID)
Checks if the given ID exists in the system.
- [uint8 Check_Flag](#) (uint8 _ID)
Returns the status flag associated with the given ID.
- void [Flag_SET_RESET](#) (uint8 _ID)
Toggles the flag associated with the given ID.
- [uint8 Check_Password](#) (uint8 _ID)
Checks if the entered password matches the stored password for the given ID.
- void [combineArrays](#) (uint8_t *array1, int size1, uint8_t *array2, int size2, uint8_t *array3, int size3, uint8_t *array4, int size4, uint8_t *combinedArray)
Combines four arrays into a single array.
- void [Enter_Gate_Open](#) (void)
Opens the entrance gate and displays the status on the LCD.
- void [Exit_Gate_Open](#) (void)
Opens the exit gate and displays the status on the LCD.
- void [Wrong_RFID](#) (void)
Triggers the alarm and prints "UNKNOWN ID!" on the LCD.
- void [Trigger_Alarm](#) (USART_TypeDef *_USART)
Echoes the entered ID via USART and flashes the red LED.

5.160.1 Detailed Description

APIs for managing the ECU functionalities.

5.160.2 Function Documentation

5.160.2.1 Admin_Init()

```
void Admin_Init (
    void )
```

Initializes the admin settings, including setting up user IDs.

This function is called at system startup to set the user IDs and configure the initial admin settings.

Returns

None

Initializes the admin settings, including setting up user IDs.

This function is called at the very start of the system to set the users' IDs. It prompts the admin to enter user IDs via a keypad and displays the entered IDs on the LCD. It also sets the default screen for the rest of the program.

Parameters

in	None	
----	------	--

Returns

None

5.160.2.2 Check_Flag()

```
uint8 Check_Flag (
    uint8 _ID )
```

Returns the status flag associated with the given ID.

This function checks the flag status for the specified user ID. It compares the input ID with stored user IDs and returns the corresponding flag value if a match is found.

Parameters

in	_ID	ID whose flag status is to be checked.
----	-----	--

Returns

User flag if the ID matches; 0 if no match is found.

5.160.2.3 Check_ID()

```
ID_Check_Result Check_ID (
    uint8 _ID )
```

Checks if the given ID exists in the system.

This function checks if the provided ID exists among the saved user IDs.

Parameters

in	_ID	The ID to check.
----	-----	------------------

Returns

ID_Found if the ID is found, ID_NOT_Found otherwise.

Checks if the given ID exists in the system.

This function searches for the specified ID in the list of saved IDs and returns the result of the search.

Parameters

in	_ID	ID to check.
----	-----	--------------

Returns

ID_Found if the ID is found, ID_NOT_Found otherwise.

5.160.2.4 Check_Password()

```
uint8 Check_Password (
    uint8 _ID )
```

Checks if the entered password matches the stored password for the given ID.

This function prompts the user to enter a 4-digit password using a keypad. It compares the entered password with the stored password for the corresponding ID. If the passwords match, the function returns 1; otherwise, it returns 0.

Parameters

in	_ID	ID for which the password is being checked.
----	-----	---

Returns

1 if the entered password matches the stored password, 0 otherwise.

5.160.2.5 combineArrays()

```
void combineArrays (
    uint8_t * array1,
    int size1,
    uint8_t * array2,
    int size2,
    uint8_t * array3,
    int size3,
    uint8_t * array4,
    int size4,
    uint8_t * combinedArray )
```

Combines four arrays into a single array.

This function merges four separate arrays into one single array. It copies the contents of each input array sequentially into the `combinedArray`. The sizes of the arrays and the final combined array should be managed carefully to avoid buffer overflows.

Parameters

in	<code>array1</code>	Pointer to the first input array.
in	<code>size1</code>	Number of elements in the first array.
in	<code>array2</code>	Pointer to the second input array.
in	<code>size2</code>	Number of elements in the second array.
in	<code>array3</code>	Pointer to the third input array.
in	<code>size3</code>	Number of elements in the third array.
in	<code>array4</code>	Pointer to the fourth input array.
in	<code>size4</code>	Number of elements in the fourth array.
out	<code>combinedArray</code>	Pointer to the array where the combined data will be stored. The size of this array must be at least <code>size1 + size2 + size3 + size4</code> .

Returns

void

5.160.2.6 ECU_Init()

```
void ECU_Init (
    void  )
```

Initializes all hardware components and peripherals.

This function initializes the necessary peripherals for the application, including USART, servo motors, LCD, LED, and keypad drivers.

Returns

None

Note

Must be called on system boot.

This function sets up all the essential peripherals, including clock initialization, LED configuration, LCD, UART, and sensor setups. Must be called during system startup.

Returns

None

5.160.2.7 Enter_Gate_Open()

```
void Enter_Gate_Open (
    void )
```

Opens the entrance gate and displays the status on the LCD.

This function opens the entrance gate and prints a message on the LCD indicating that the gate is open.

Returns

None

Opens the entrance gate and displays the status on the LCD.

This function activates the servo to open the entry gate, turns on the green LED to indicate the gate is open, and waits until the PIR sensor detects that the gate is closed before closing the gate.

Parameters

in	None	
----	------	--

Returns

None

5.160.2.8 Exit_Gate_Open()

```
void Exit_Gate_Open (
    void )
```

Opens the exit gate and displays the status on the LCD.

This function opens the exit gate and prints a message on the LCD indicating that the gate is open.

Returns

None

Opens the exit gate and displays the status on the LCD.

This function activates the servo to open the exit gate, turns on the green LED to indicate the gate is open, and waits until the PIR sensor detects that the gate is closed before closing the gate.

Parameters

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

Returns

None

5.160.2.9 Flag_SET_RESET()

```
void Flag_SET_RESET (
    uint8 _ID )
```

Toggles the flag associated with the given ID.

This function toggles the flag of a user identified by the provided ID. If the flag is currently set (1), it will be reset (0), and vice versa.

Parameters

in	<i>_ID</i>	ID whose flag is to be toggled.
----	------------	---------------------------------

5.160.2.10 Trigger_Alarm()

```
void Trigger_Alarm (
    USART_TypeDef * _USART )
```

Echoes the entered ID via USART and flashes the red LED.

This function echoes the entered ID through the specified USART and flashes the red LED to signal an alarm condition.

Parameters

in	<i>_USART</i>	Pointer to the USART instance that received the ID.
----	---------------	---

Returns

None

Echoes the entered ID via USART and flashes the red LED.

This function retrieves the entered ID from the specified USART instance, echoes the ID back on UART, and triggers the red LED to flash multiple times as an alarm indication.

Parameters

in	_USART	Pointer to the USART instance that contains the received ID.
----	--------	--

Returns

None

Note

The function uses blocking delays to flash the LED.

5.160.2.11 UserLCD_PrintFreeSlots()

```
void UserLCD_PrintFreeSlots (
    void )
```

Prints the number of free parking slots on the LCD.

This function prints the number of available parking slots on the user LCD when requested by setting the `Print←_Slots_LCD_Flag`.

Returns

None

Note

`Print_Slots_LCD_Flag` must be set to 1 before calling this function.

Prints the number of free parking slots on the LCD.

This function updates the UserLCD with the number of free slots if the `Print_Slots_LCD_Flag` is set. If there are no free slots, it displays a message indicating that parking is full.

Parameters

in	None	
----	------	--

Returns

None

Note

Must set Print_Slots_LCD_Flag to 1 before calling this function.

5.160.2.12 Wrong_RFID()

```
void Wrong_RFID (
    void )
```

Triggers the alarm and prints "UNKNOWN ID!" on the LCD.

This function triggers an alarm when an unrecognized RFID tag is presented, and displays a message indicating the unknown ID on the LCD.

Returns

None

Triggers the alarm and prints "UNKNOWN ID!" on the LCD.

This function activates the red LED to indicate an error and sets the servos to the down position. It also displays "UNKNOWN ID!" on the LCD to inform the user of the error.

Parameters

in	None	
----	------	--

Returns

None

5.161 Functions_Declarations

Declaration of callback functions for UART operations.

Collaboration diagram for Functions_Declarations:



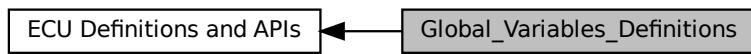
5.161.1 Detailed Description

Declaration of callback functions for UART operations.

5.162 Global_Variables_Definitions

Global variable definitions for LEDs, LCDs, UART, and system states.

Collaboration diagram for Global_Variables_Definitions:



Macros

- `#define SIZE1 3`
Definitions for combining arrays.
- `#define SIZE2 4`
- `#define SIZE3 4`
- `#define SIZE4 4`
- `#define TOTAL_SIZE (SIZE1 + SIZE2 + SIZE3 + SIZE4)`

Variables

- `volatile uint8 Enter_Flag`
Flags and variables used for managing system state.
- `volatile uint8 Exit_Flag`
- `uint8 Free_Slots = 3`
- `uint8 Print_Slots_LCD_Flag`

5.162.1 Detailed Description

Global variable definitions for LEDs, LCDs, UART, and system states.

5.162.2 Macro Definition Documentation

5.162.2.1 SIZE1

```
#define SIZE1 3
```

Definitions for combining arrays.

- SIZE1: Size of the first array.
- SIZE2: Size of the second array.
- SIZE3: Size of the third array.
- SIZE4: Size of the fourth array.
- TOTAL_SIZE: Total size for the combined array.

5.162.3 Variable Documentation

5.162.3.1 Exit_Flag

```
volatile uint8 Exit_Flag
```

Flags to manage entry and exit

5.162.3.2 Free_Slots

```
uint8 Free_Slots = 3
```

Number of free parking slots

5.162.3.3 Print_Slots_LCD_Flag

```
uint8 Print_Slots_LCD_Flag
```

Flag to trigger printing free slots

5.163 API_Definitions

API definitions for the ECU functionalities.

Collaboration diagram for API_Definitions:



Functions

- void [ECU_Init](#) (void)
Initializes all hardware components and peripherals.
- void [Admin_Init](#) (void)
Initializes user IDs and displays them on the LCD.
- void [UserLCD_PrintFreeSlots](#) (void)
Prints the number of free slots on the UserLCD.
- [ID_Check_Result Check_ID](#) (uint8 _ID)
Checks if the given ID is present in the saved IDs.
- [uint8 Check_Flag](#) (uint8 _ID)
Returns the status flag associated with the given ID.
- void [Flag_SET_RESET](#) (uint8 _ID)
Toggles the flag associated with the given ID.
- [uint8 Check_Password](#) (uint8 _ID)
Checks if the entered password matches the stored password for the given ID.
- void [combineArrays](#) (uint8_t *array1, int size1, uint8_t *array2, int size2, uint8_t *array3, int size3, uint8_t *array4, int size4, uint8_t *combinedArray)
Combines four arrays into a single array.
- void [Enter_Gate_Open](#) (void)
Opens the entry gate and displays a message on the LCD.
- void [Exit_Gate_Open](#) (void)
Opens the exit gate and displays a message on the LCD.
- void [Wrong_RFID](#) (void)
Triggers an alarm and displays an "UNKNOWN ID!" message on the LCD.
- void [Trigger_Alarm](#) (USART_TypeDef *_USART)
Triggers an alarm by echoing the received ID on UART and flashing the red LED.
- void [Enter_UART_CallBack](#) (void)
Callback function for handling entry UART events.
- void [Exit_UART_CallBack](#) (void)
Callback function for handling exit UART events.

5.163.1 Detailed Description

API definitions for the ECU functionalities.

5.163.2 Function Documentation

5.163.2.1 Admin_Init()

```
void Admin_Init (
    void )
```

Initializes user IDs and displays them on the LCD.

Initializes the admin settings, including setting up user IDs.

This function is called at the very start of the system to set the users' IDs. It prompts the admin to enter user IDs via a keypad and displays the entered IDs on the LCD. It also sets the default screen for the rest of the program.

Parameters

in	None	
----	------	--

Returns

None

5.163.2.2 Check_Flag()

```
uint8 Check_Flag (
    uint8 _ID )
```

Returns the status flag associated with the given ID.

This function checks the flag status for the specified user ID. It compares the input ID with stored user IDs and returns the corresponding flag value if a match is found.

Parameters

in	_ID	ID whose flag status is to be checked.
----	-----	--

Returns

User flag if the ID matches; 0 if no match is found.

5.163.2.3 Check_ID()

```
ID_Check_Result Check_ID (
    uint8 _ID )
```

Checks if the given ID is present in the saved IDs.

Checks if the given ID exists in the system.

This function searches for the specified ID in the list of saved IDs and returns the result of the search.

Parameters

in	_ID	ID to check.
----	-----	--------------

Returns

ID_Found if the ID is found, ID_NOT_Found otherwise.

5.163.2.4 Check_Password()

```
uint8 Check_Password (
    uint8 _ID )
```

Checks if the entered password matches the stored password for the given ID.

This function prompts the user to enter a 4-digit password using a keypad. It compares the entered password with the stored password for the corresponding ID. If the passwords match, the function returns 1; otherwise, it returns 0.

Parameters

in	<i>_ID</i>	ID for which the password is being checked.
----	------------	---

Returns

1 if the entered password matches the stored password, 0 otherwise.

5.163.2.5 combineArrays()

```
void combineArrays (
    uint8_t * array1,
    int size1,
    uint8_t * array2,
    int size2,
    uint8_t * array3,
    int size3,
    uint8_t * array4,
    int size4,
    uint8_t * combinedArray )
```

Combines four arrays into a single array.

This function merges four separate arrays into one single array. It copies the contents of each input array sequentially into the `combinedArray`. The sizes of the arrays and the final combined array should be managed carefully to avoid buffer overflows.

Parameters

in	<i>array1</i>	Pointer to the first input array.
in	<i>size1</i>	Number of elements in the first array.
in	<i>array2</i>	Pointer to the second input array.
in	<i>size2</i>	Number of elements in the second array.
in	<i>array3</i>	Pointer to the third input array.
in	<i>size3</i>	Number of elements in the third array.
in	<i>array4</i>	Pointer to the fourth input array.
in	<i>size4</i>	Number of elements in the fourth array.
out	<i>combinedArray</i>	Pointer to the array where the combined data will be stored. The size of this array must be at least <code>size1 + size2 + size3 + size4</code> .

Returns

```
void
```

5.163.2.6 ECU_Init()

```
void ECU_Init (
    void )
```

Initializes all hardware components and peripherals.

This function sets up all the essential peripherals, including clock initialization, LED configuration, LCD, UART, and sensor setups. Must be called during system startup.

Returns

```
None
```

5.163.2.7 Enter_Gate_Open()

```
void Enter_Gate_Open (
    void )
```

Opens the entry gate and displays a message on the LCD.

Opens the entrance gate and displays the status on the LCD.

This function activates the servo to open the entry gate, turns on the green LED to indicate the gate is open, and waits until the PIR sensor detects that the gate is closed before closing the gate.

Parameters

in	None	
----	------	--

Returns

```
None
```

5.163.2.8 Enter_UART_CallBack()

```
void Enter_UART_CallBack (
    void )
```

Callback function for handling entry UART events.

This function sets the `Enter_Flag` to 1 when called, indicating that an entry event has occurred via UART.

Parameters

in	None	
----	------	--

Returns

None

5.163.2.9 Exit_Gate_Open()

```
void Exit_Gate_Open (
    void )
```

Opens the exit gate and displays a message on the LCD.

Opens the exit gate and displays the status on the LCD.

This function activates the servo to open the exit gate, turns on the green LED to indicate the gate is open, and waits until the PIR sensor detects that the gate is closed before closing the gate.

Parameters

in	None	
----	------	--

Returns

None

5.163.2.10 Exit_UART_CallBack()

```
void Exit_UART_CallBack (
    void )
```

Callback function for handling exit UART events.

This function sets the `Exit_Flag` to 1 when called, indicating that an exit event has occurred via UART.

Parameters

in	None	
----	------	--

Returns

None

5.163.2.11 Flag_SET_RESET()

```
void Flag_SET_RESET (
    uint8 _ID )
```

Toggles the flag associated with the given ID.

This function toggles the flag of a user identified by the provided ID. If the flag is currently set (1), it will be reset (0), and vice versa.

Parameters

in	_ID	ID whose flag is to be toggled.
----	-----	---------------------------------

5.163.2.12 Trigger_Alarm()

```
void Trigger_Alarm (
    USART_TypeDef * _USART )
```

Triggers an alarm by echoing the received ID on UART and flashing the red LED.

Echoes the entered ID via USART and flashes the red LED.

This function retrieves the entered ID from the specified USART instance, echoes the ID back on UART, and triggers the red LED to flash multiple times as an alarm indication.

Parameters

in	_USART	Pointer to the USART instance that contains the received ID.
----	--------	--

Returns

None

Note

The function uses blocking delays to flash the LED.

5.163.2.13 UserLCD_PrintFreeSlots()

```
void UserLCD_PrintFreeSlots (
    void )
```

Prints the number of free slots on the UserLCD.

Prints the number of free parking slots on the LCD.

This function updates the UserLCD with the number of free slots if the Print_Slots_LCD_Flag is set. If there are no free slots, it displays a message indicating that parking is full.

Parameters

in	None	
----	------	--

Returns

None

Note

Must set Print_Slots_LCD_Flag to 1 before calling this function.

5.163.2.14 Wrong_RFID()

```
void Wrong_RFID (
    void )
```

Triggers an alarm and displays an "UNKNOWN ID!" message on the LCD.

Triggers the alarm and prints "UNKNOWN ID!" on the LCD.

This function activates the red LED to indicate an error and sets the servos to the down position. It also displays "UNKNOWN ID!" on the LCD to inform the user of the error.

Parameters

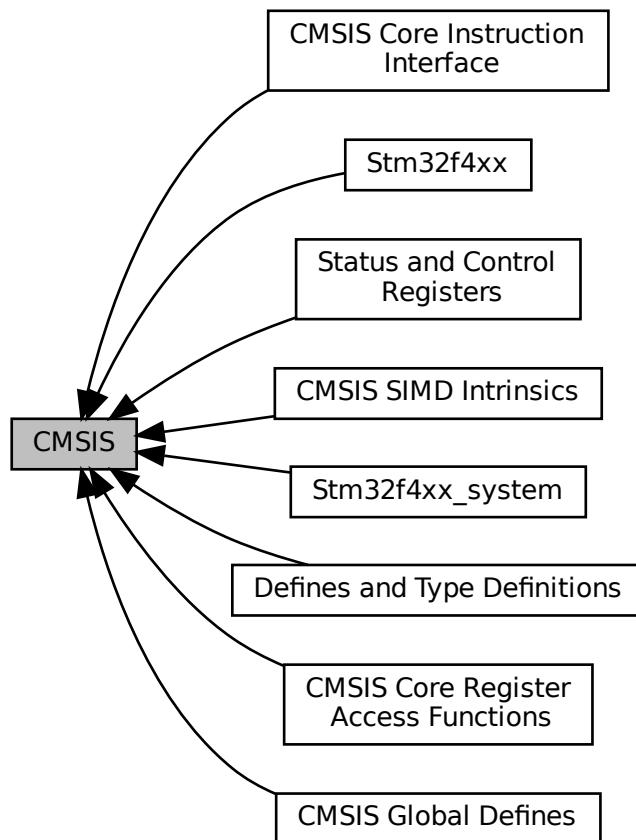
in	None	
----	------	--

Returns

None

5.164 CMSIS

Collaboration diagram for CMSIS:



Modules

- CMSIS Global Defines
 - Defines and Type Definitions
- Type definitions and defines for Cortex-M processor based devices.*
- Status and Control Registers
- Core Register type definitions.*
- CMSIS Core Register Access Functions
 - CMSIS Core Instruction Interface
 - CMSIS SIMD Intrinsics
 - Stm32f4xx
 - Stm32f4xx_system

Functions

- void **NMI_Handler** (void)
- void **HardFault_Handler** (void)
- void **MemManage_Handler** (void)
- void **BusFault_Handler** (void)
- void **UsageFault_Handler** (void)
- void **SVC_Handler** (void)
- void **DebugMon_Handler** (void)
- void **PendSV_Handler** (void)
- void **SysTick_Handler** (void)

SysTick interrupt handler.

Variables

- `_IO uint32_t ADC_TypeDef::SR`
- `_IO uint32_t ADC_TypeDef::CR1`
- `_IO uint32_t ADC_TypeDef::CR2`
- `_IO uint32_t ADC_TypeDef::SMPR1`
- `_IO uint32_t ADC_TypeDef::SMPR2`
- `_IO uint32_t ADC_TypeDef::JOFR1`
- `_IO uint32_t ADC_TypeDef::JOFR2`
- `_IO uint32_t ADC_TypeDef::JOFR3`
- `_IO uint32_t ADC_TypeDef::JOFR4`
- `_IO uint32_t ADC_TypeDef::HTR`
- `_IO uint32_t ADC_TypeDef::LTR`
- `_IO uint32_t ADC_TypeDef::SQR1`
- `_IO uint32_t ADC_TypeDef::SQR2`
- `_IO uint32_t ADC_TypeDef::SQR3`
- `_IO uint32_t ADC_TypeDef::JSQR`
- `_IO uint32_t ADC_TypeDef::JDR1`
- `_IO uint32_t ADC_TypeDef::JDR2`
- `_IO uint32_t ADC_TypeDef::JDR3`
- `_IO uint32_t ADC_TypeDef::JDR4`
- `_IO uint32_t ADC_TypeDef::DR`
- `_IO uint32_t ADC_Common_TypeDef::CSR`
- `_IO uint32_t ADC_Common_TypeDef::CCR`
- `_IO uint32_t ADC_Common_TypeDef::CDR`
- `_IO uint32_t CAN_TxMailBox_TypeDef::TIR`
- `_IO uint32_t CAN_TxMailBox_TypeDef::TDTR`
- `_IO uint32_t CAN_TxMailBox_TypeDef::TDLR`
- `_IO uint32_t CAN_TxMailBox_TypeDef::TDHR`
- `_IO uint32_t CAN_FIFOMailBox_TypeDef::RIR`
- `_IO uint32_t CAN_FIFOMailBox_TypeDef::RDTR`
- `_IO uint32_t CAN_FIFOMailBox_TypeDef::RDLR`
- `_IO uint32_t CAN_FIFOMailBox_TypeDef::RDHR`
- `_IO uint32_t CAN_FilterRegister_TypeDef::FR1`
- `_IO uint32_t CAN_FilterRegister_TypeDef::FR2`
- `_IO uint32_t CAN_TypeDef::MCR`
- `_IO uint32_t CAN_TypeDef::MSR`
- `_IO uint32_t CAN_TypeDef::TSR`
- `_IO uint32_t CAN_TypeDef::RF0R`
- `_IO uint32_t CAN_TypeDef::RF1R`
- `_IO uint32_t CAN_TypeDef::IER`

- `__IO uint32_t CAN_TypeDef::ESR`
- `__IO uint32_t CAN_TypeDef::BTR`
- `uint32_t CAN_TypeDef::RESERVED0 [88]`
- `CAN_TxMailBox_TypeDef CAN_TypeDef::sTxMailBox [3]`
- `CAN_FIFOMailBox_TypeDef CAN_TypeDef::sFIFOMailBox [2]`
- `uint32_t CAN_TypeDef::RESERVED1 [12]`
- `__IO uint32_t CAN_TypeDef::FMR`
- `__IO uint32_t CAN_TypeDef::FM1R`
- `uint32_t CAN_TypeDef::RESERVED2`
- `__IO uint32_t CAN_TypeDef::FS1R`
- `uint32_t CAN_TypeDef::RESERVED3`
- `__IO uint32_t CAN_TypeDef::FFA1R`
- `uint32_t CAN_TypeDef::RESERVED4`
- `__IO uint32_t CAN_TypeDef::FA1R`
- `uint32_t CAN_TypeDef::RESERVED5 [8]`
- `CAN_FilterRegister_TypeDef CAN_TypeDef::sFilterRegister [28]`
- `__IO uint32_t CRC_TypeDef::DR`
- `__IO uint8_t CRC_TypeDef::IDR`
- `uint8_t CRC_TypeDef::RESERVED0`
- `uint16_t CRC_TypeDef::RESERVED1`
- `__IO uint32_t CRC_TypeDef::CR`
- `__IO uint32_t DAC_TypeDef::CR`
- `__IO uint32_t DAC_TypeDef::SWTRIGR`
- `__IO uint32_t DAC_TypeDef::DHR12R1`
- `__IO uint32_t DAC_TypeDef::DHR12L1`
- `__IO uint32_t DAC_TypeDef::DHR8R1`
- `__IO uint32_t DAC_TypeDef::DHR12R2`
- `__IO uint32_t DAC_TypeDef::DHR12L2`
- `__IO uint32_t DAC_TypeDef::DHR8R2`
- `__IO uint32_t DAC_TypeDef::DHR12RD`
- `__IO uint32_t DAC_TypeDef::DHR12LD`
- `__IO uint32_t DAC_TypeDef::DHR8RD`
- `__IO uint32_t DAC_TypeDef::DOR1`
- `__IO uint32_t DAC_TypeDef::DOR2`
- `__IO uint32_t DAC_TypeDef::SR`
- `__IO uint32_t DBGMCU_TypeDef::IDCODE`
- `__IO uint32_t DBGMCU_TypeDef::CR`
- `__IO uint32_t DBGMCU_TypeDef::APB1FZ`
- `__IO uint32_t DBGMCU_TypeDef::APB2FZ`
- `__IO uint32_t DCMI_TypeDef::CR`
- `__IO uint32_t DCMI_TypeDef::SR`
- `__IO uint32_t DCMI_TypeDef::RISR`
- `__IO uint32_t DCMI_TypeDef::IER`
- `__IO uint32_t DCMI_TypeDef::MISR`
- `__IO uint32_t DCMI_TypeDef::ICR`
- `__IO uint32_t DCMI_TypeDef::ESCR`
- `__IO uint32_t DCMI_TypeDef::ESUR`
- `__IO uint32_t DCMI_TypeDef::CWSTRTR`
- `__IO uint32_t DCMI_TypeDef::CWSIZER`
- `__IO uint32_t DCMI_TypeDef::DR`
- `__IO uint32_t DMA_Stream_TypeDef::CR`
- `__IO uint32_t DMA_Stream_TypeDef::NDTR`
- `__IO uint32_t DMA_Stream_TypeDef::PAR`
- `__IO uint32_t DMA_Stream_TypeDef::M0AR`
- `__IO uint32_t DMA_Stream_TypeDef::M1AR`

- `__IO uint32_t DMA_Stream_TypeDef::FCR`
- `__IO uint32_t DMA_TypeDef::LISR`
- `__IO uint32_t DMA_TypeDef::HISR`
- `__IO uint32_t DMA_TypeDef::LIFCR`
- `__IO uint32_t DMA_TypeDef::HIFCR`
- `__IO uint32_t DMA2D_TypeDef::CR`
- `__IO uint32_t DMA2D_TypeDef::ISR`
- `__IO uint32_t DMA2D_TypeDef::IFCR`
- `__IO uint32_t DMA2D_TypeDef::FGMAR`
- `__IO uint32_t DMA2D_TypeDef::FGOR`
- `__IO uint32_t DMA2D_TypeDef::BGMAR`
- `__IO uint32_t DMA2D_TypeDef::BGOR`
- `__IO uint32_t DMA2D_TypeDef::FGPFCCR`
- `__IO uint32_t DMA2D_TypeDef::FGCOLR`
- `__IO uint32_t DMA2D_TypeDef::BGPFCCR`
- `__IO uint32_t DMA2D_TypeDef::BGCOLR`
- `__IO uint32_t DMA2D_TypeDef::FGCMAR`
- `__IO uint32_t DMA2D_TypeDef::BGCMAR`
- `__IO uint32_t DMA2D_TypeDef::OPFCCR`
- `__IO uint32_t DMA2D_TypeDef::OCOLR`
- `__IO uint32_t DMA2D_TypeDef::OMAR`
- `__IO uint32_t DMA2D_TypeDef::OOR`
- `__IO uint32_t DMA2D_TypeDef::NLR`
- `__IO uint32_t DMA2D_TypeDef::LWR`
- `__IO uint32_t DMA2D_TypeDef::AMTCR`
- `uint32_t DMA2D_TypeDef::RESERVED [236]`
- `__IO uint32_t DMA2D_TypeDef::FGCLUT [256]`
- `__IO uint32_t DMA2D_TypeDef::BGCLUT [256]`
- `__IO uint32_t ETH_TypeDef::MACCR`
- `__IO uint32_t ETH_TypeDef::MACFFR`
- `__IO uint32_t ETH_TypeDef::MACHTHR`
- `__IO uint32_t ETH_TypeDef::MACHTLR`
- `__IO uint32_t ETH_TypeDef::MACMIIAR`
- `__IO uint32_t ETH_TypeDef::MACMIIDR`
- `__IO uint32_t ETH_TypeDef::MACFCR`
- `__IO uint32_t ETH_TypeDef::MACVLANTR`
- `uint32_t ETH_TypeDef::RESERVED0 [2]`
- `__IO uint32_t ETH_TypeDef::MACRWUFFR`
- `__IO uint32_t ETH_TypeDef::MACPMTCSR`
- `uint32_t ETH_TypeDef::RESERVED1 [2]`
- `__IO uint32_t ETH_TypeDef::MACSR`
- `__IO uint32_t ETH_TypeDef::MACIMR`
- `__IO uint32_t ETH_TypeDef::MACA0HR`
- `__IO uint32_t ETH_TypeDef::MACA0LR`
- `__IO uint32_t ETH_TypeDef::MACA1HR`
- `__IO uint32_t ETH_TypeDef::MACA1LR`
- `__IO uint32_t ETH_TypeDef::MACA2HR`
- `__IO uint32_t ETH_TypeDef::MACA2LR`
- `__IO uint32_t ETH_TypeDef::MACA3HR`
- `__IO uint32_t ETH_TypeDef::MACA3LR`
- `uint32_t ETH_TypeDef::RESERVED2 [40]`
- `__IO uint32_t ETH_TypeDef::MMCCR`
- `__IO uint32_t ETH_TypeDef::MMCRIR`
- `__IO uint32_t ETH_TypeDef::MMCTIR`
- `__IO uint32_t ETH_TypeDef::MMCRIMR`

- `__IO uint32_t ETH_TypeDef::MMCTIMR`
- `uint32_t ETH_TypeDef::RESERVED3 [14]`
- `__IO uint32_t ETH_TypeDef::MMCTGFSCCR`
- `__IO uint32_t ETH_TypeDef::MMCTGFMSCCR`
- `uint32_t ETH_TypeDef::RESERVED4 [5]`
- `__IO uint32_t ETH_TypeDef::MMCTGFCR`
- `uint32_t ETH_TypeDef::RESERVED5 [10]`
- `__IO uint32_t ETH_TypeDef::MMCRFCECR`
- `__IO uint32_t ETH_TypeDef::MMCRFAECR`
- `uint32_t ETH_TypeDef::RESERVED6 [10]`
- `__IO uint32_t ETH_TypeDef::MMCRGUFCR`
- `uint32_t ETH_TypeDef::RESERVED7 [334]`
- `__IO uint32_t ETH_TypeDef::PTPTSCR`
- `__IO uint32_t ETH_TypeDef::PTPSSIR`
- `__IO uint32_t ETH_TypeDef::PTPTSHR`
- `__IO uint32_t ETH_TypeDef::PTPTSLR`
- `__IO uint32_t ETH_TypeDef::PTPTSHUR`
- `__IO uint32_t ETH_TypeDef::PTPTSLUR`
- `__IO uint32_t ETH_TypeDef::PTPTSAR`
- `__IO uint32_t ETH_TypeDef::PTPTTHR`
- `__IO uint32_t ETH_TypeDef::PTPTTLR`
- `__IO uint32_t ETH_TypeDef::RESERVED8`
- `__IO uint32_t ETH_TypeDef::PTPTSSR`
- `uint32_t ETH_TypeDef::RESERVED9 [565]`
- `__IO uint32_t ETH_TypeDef::DMABMR`
- `__IO uint32_t ETH_TypeDef::DMATPDR`
- `__IO uint32_t ETH_TypeDef::DMARPDR`
- `__IO uint32_t ETH_TypeDef::DMARDLAR`
- `__IO uint32_t ETH_TypeDef::DMATDLAR`
- `__IO uint32_t ETH_TypeDef::DMASR`
- `__IO uint32_t ETH_TypeDef::DMAOMR`
- `__IO uint32_t ETH_TypeDef::DMAIER`
- `__IO uint32_t ETH_TypeDef::DMAMFBOCR`
- `__IO uint32_t ETH_TypeDef::DMARSWTR`
- `uint32_t ETH_TypeDef::RESERVED10 [8]`
- `__IO uint32_t ETH_TypeDef::DMACHTDR`
- `__IO uint32_t ETH_TypeDef::DMACHRDR`
- `__IO uint32_t ETH_TypeDef::DMACHTBAR`
- `__IO uint32_t ETH_TypeDef::DMACHRBAR`
- `__IO uint32_t EXTI_TypeDef::IMR`
- `__IO uint32_t EXTI_TypeDef::EMR`
- `__IO uint32_t EXTI_TypeDef::RTSR`
- `__IO uint32_t EXTI_TypeDef::FTSR`
- `__IO uint32_t EXTI_TypeDef::SWIER`
- `__IO uint32_t EXTI_TypeDef::PR`
- `__IO uint32_t FLASH_TypeDef::ACR`
- `__IO uint32_t FLASH_TypeDef::KEYR`
- `__IO uint32_t FLASH_TypeDef::OPTKEYR`
- `__IO uint32_t FLASH_TypeDef::SR`
- `__IO uint32_t FLASH_TypeDef::CR`
- `__IO uint32_t FLASH_TypeDef::OPTCR`
- `__IO uint32_t FLASH_TypeDef::OPTCR1`
- `__IO uint32_t GPIO_TypeDef::MODER`
- `__IO uint32_t GPIO_TypeDef::OTYPER`
- `__IO uint32_t GPIO_TypeDef::OSPEEDR`

- `__IO uint32_t GPIO_TypeDef::PUPDR`
- `__IO uint32_t GPIO_TypeDef::IDR`
- `__IO uint32_t GPIO_TypeDef::ODR`
- `__IO uint16_t GPIO_TypeDef::BSRRL`
- `__IO uint16_t GPIO_TypeDef::BSRRH`
- `__IO uint32_t GPIO_TypeDef::LCKR`
- `__IO uint32_t GPIO_TypeDef::AFR [2]`
- `__IO uint32_t SYSCFG_TypeDef::MEMRMP`
- `__IO uint32_t SYSCFG_TypeDef::PMC`
- `__IO uint32_t SYSCFG_TypeDef::EXTICR [4]`
- `uint32_t SYSCFG_TypeDef::RESERVED [2]`
- `__IO uint32_t SYSCFG_TypeDef::CMPCR`
- `__IO uint16_t I2C_TypeDef::CR1`
- `uint16_t I2C_TypeDef::RESERVED0`
- `__IO uint16_t I2C_TypeDef::CR2`
- `uint16_t I2C_TypeDef::RESERVED1`
- `__IO uint16_t I2C_TypeDef::OAR1`
- `uint16_t I2C_TypeDef::RESERVED2`
- `__IO uint16_t I2C_TypeDef::OAR2`
- `uint16_t I2C_TypeDef::RESERVED3`
- `__IO uint16_t I2C_TypeDef::DR`
- `uint16_t I2C_TypeDef::RESERVED4`
- `__IO uint16_t I2C_TypeDef::SR1`
- `uint16_t I2C_TypeDef::RESERVED5`
- `__IO uint16_t I2C_TypeDef::SR2`
- `uint16_t I2C_TypeDef::RESERVED6`
- `__IO uint16_t I2C_TypeDef::CCR`
- `uint16_t I2C_TypeDef::RESERVED7`
- `__IO uint16_t I2C_TypeDef::TRISE`
- `uint16_t I2C_TypeDef::RESERVED8`
- `__IO uint16_t I2C_TypeDef::FLTR`
- `uint16_t I2C_TypeDef::RESERVED9`
- `__IO uint32_t IWDG_TypeDef::KR`
- `__IO uint32_t IWDG_TypeDef::PR`
- `__IO uint32_t IWDG_TypeDef::RLR`
- `__IO uint32_t IWDG_TypeDef::SR`
- `uint32_t LTDC_TypeDef::RESERVED0 [2]`
- `__IO uint32_t LTDC_TypeDef::SSCR`
- `__IO uint32_t LTDC_TypeDef::BPCR`
- `__IO uint32_t LTDC_TypeDef::AWCR`
- `__IO uint32_t LTDC_TypeDef::TWCR`
- `__IO uint32_t LTDC_TypeDef::GCR`
- `uint32_t LTDC_TypeDef::RESERVED1 [2]`
- `__IO uint32_t LTDC_TypeDef::SRCR`
- `uint32_t LTDC_TypeDef::RESERVED2 [1]`
- `__IO uint32_t LTDC_TypeDef::BCCR`
- `uint32_t LTDC_TypeDef::RESERVED3 [1]`
- `__IO uint32_t LTDC_TypeDef::IER`
- `__IO uint32_t LTDC_TypeDef::ISR`
- `__IO uint32_t LTDC_TypeDef::ICR`
- `__IO uint32_t LTDC_TypeDef::LIPCR`
- `__IO uint32_t LTDC_TypeDef::CPSR`
- `__IO uint32_t LTDC_TypeDef::CDSR`
- `__IO uint32_t LTDC_Layer_TypeDef::CR`
- `__IO uint32_t LTDC_Layer_TypeDef::WHPCR`

- `_IO uint32_t LTDC_Layer_TypeDef::WVPCR`
- `_IO uint32_t LTDC_Layer_TypeDef::CKCR`
- `_IO uint32_t LTDC_Layer_TypeDef::PFCR`
- `_IO uint32_t LTDC_Layer_TypeDef::CACR`
- `_IO uint32_t LTDC_Layer_TypeDef::DCCR`
- `_IO uint32_t LTDC_Layer_TypeDef::BFCR`
- `uint32_t LTDC_Layer_TypeDef::RESERVED0 [2]`
- `_IO uint32_t LTDC_Layer_TypeDef::CFBAR`
- `_IO uint32_t LTDC_Layer_TypeDef::CFBLR`
- `_IO uint32_t LTDC_Layer_TypeDef::CFBLNR`
- `uint32_t LTDC_Layer_TypeDef::RESERVED1 [3]`
- `_IO uint32_t LTDC_Layer_TypeDef::CLUTWR`
- `_IO uint32_t PWR_TypeDef::CR`
- `_IO uint32_t PWR_TypeDef::CSR`
- `_IO uint32_t RCC_TypeDef::CR`
- `_IO uint32_t RCC_TypeDef::PLLCFGR`
- `_IO uint32_t RCC_TypeDef::CFGGR`
- `_IO uint32_t RCC_TypeDef::CIR`
- `_IO uint32_t RCC_TypeDef::AHB1RSTR`
- `_IO uint32_t RCC_TypeDef::AHB2RSTR`
- `_IO uint32_t RCC_TypeDef::AHB3RSTR`
- `uint32_t RCC_TypeDef::RESERVED0`
- `_IO uint32_t RCC_TypeDef::APB1RSTR`
- `_IO uint32_t RCC_TypeDef::APB2RSTR`
- `uint32_t RCC_TypeDef::RESERVED1 [2]`
- `_IO uint32_t RCC_TypeDef::AHB1ENR`
- `_IO uint32_t RCC_TypeDef::AHB2ENR`
- `_IO uint32_t RCC_TypeDef::AHB3ENR`
- `uint32_t RCC_TypeDef::RESERVED2`
- `_IO uint32_t RCC_TypeDef::APB1ENR`
- `_IO uint32_t RCC_TypeDef::APB2ENR`
- `uint32_t RCC_TypeDef::RESERVED3 [2]`
- `_IO uint32_t RCC_TypeDef::AHB1LPENR`
- `_IO uint32_t RCC_TypeDef::AHB2LPENR`
- `_IO uint32_t RCC_TypeDef::AHB3LPENR`
- `uint32_t RCC_TypeDef::RESERVED4`
- `_IO uint32_t RCC_TypeDef::APB1LPENR`
- `_IO uint32_t RCC_TypeDef::APB2LPENR`
- `uint32_t RCC_TypeDef::RESERVED5 [2]`
- `_IO uint32_t RCC_TypeDef::BDCR`
- `_IO uint32_t RCC_TypeDef::CSR`
- `uint32_t RCC_TypeDef::RESERVED6 [2]`
- `_IO uint32_t RCC_TypeDef::SSCGR`
- `_IO uint32_t RCC_TypeDef::PLLSAICFGR`
- `_IO uint32_t RCC_TypeDef::DCKCFGR`
- `_IO uint32_t RCC_TypeDef::CKGATENR`
- `_IO uint32_t RCC_TypeDef::DCKCFGR2`
- `_IO uint32_t RTC_TypeDef::TR`
- `_IO uint32_t RTC_TypeDef::DR`
- `_IO uint32_t RTC_TypeDef::CR`
- `_IO uint32_t RTC_TypeDef::ISR`
- `_IO uint32_t RTC_TypeDef::PRER`
- `_IO uint32_t RTC_TypeDef::WUTR`
- `_IO uint32_t RTC_TypeDef::CALIBR`

- `__IO uint32_t RTC_TypeDef::ALRMAR`
- `__IO uint32_t RTC_TypeDef::ALRMBR`
- `__IO uint32_t RTC_TypeDef::WPR`
- `__IO uint32_t RTC_TypeDef::SSR`
- `__IO uint32_t RTC_TypeDef::SHIFTR`
- `__IO uint32_t RTC_TypeDef::TSTR`
- `__IO uint32_t RTC_TypeDef::TSDR`
- `__IO uint32_t RTC_TypeDef::TSSSR`
- `__IO uint32_t RTC_TypeDef::CALR`
- `__IO uint32_t RTC_TypeDef::TAFCR`
- `__IO uint32_t RTC_TypeDef::ALRMASSR`
- `__IO uint32_t RTC_TypeDef::ALRMBSSR`
- `uint32_t RTC_TypeDef::RESERVED7`
- `__IO uint32_t RTC_TypeDef::BKP0R`
- `__IO uint32_t RTC_TypeDef::BKP1R`
- `__IO uint32_t RTC_TypeDef::BKP2R`
- `__IO uint32_t RTC_TypeDef::BKP3R`
- `__IO uint32_t RTC_TypeDef::BKP4R`
- `__IO uint32_t RTC_TypeDef::BKP5R`
- `__IO uint32_t RTC_TypeDef::BKP6R`
- `__IO uint32_t RTC_TypeDef::BKP7R`
- `__IO uint32_t RTC_TypeDef::BKP8R`
- `__IO uint32_t RTC_TypeDef::BKP9R`
- `__IO uint32_t RTC_TypeDef::BKP10R`
- `__IO uint32_t RTC_TypeDef::BKP11R`
- `__IO uint32_t RTC_TypeDef::BKP12R`
- `__IO uint32_t RTC_TypeDef::BKP13R`
- `__IO uint32_t RTC_TypeDef::BKP14R`
- `__IO uint32_t RTC_TypeDef::BKP15R`
- `__IO uint32_t RTC_TypeDef::BKP16R`
- `__IO uint32_t RTC_TypeDef::BKP17R`
- `__IO uint32_t RTC_TypeDef::BKP18R`
- `__IO uint32_t RTC_TypeDef::BKP19R`
- `__IO uint32_t SAI_TypeDef::GCR`
- `__IO uint32_t SAI_Block_TypeDef::CR1`
- `__IO uint32_t SAI_Block_TypeDef::CR2`
- `__IO uint32_t SAI_Block_TypeDef::FRCR`
- `__IO uint32_t SAI_Block_TypeDef::SLOTR`
- `__IO uint32_t SAI_Block_TypeDef::IMR`
- `__IO uint32_t SAI_Block_TypeDef::SR`
- `__IO uint32_t SAI_Block_TypeDef::CLRFR`
- `__IO uint32_t SAI_Block_TypeDef::DR`
- `__IO uint32_t SDIO_TypeDef::POWER`
- `__IO uint32_t SDIO_TypeDef::CLKCR`
- `__IO uint32_t SDIO_TypeDef::ARG`
- `__IO uint32_t SDIO_TypeDef::CMD`
- `__I uint32_t SDIO_TypeDef::RESPCMD`
- `__I uint32_t SDIO_TypeDef::RESP1`
- `__I uint32_t SDIO_TypeDef::RESP2`
- `__I uint32_t SDIO_TypeDef::RESP3`
- `__I uint32_t SDIO_TypeDef::RESP4`
- `__IO uint32_t SDIO_TypeDef::DTIMER`
- `__IO uint32_t SDIO_TypeDef::DLEN`
- `__IO uint32_t SDIO_TypeDef::DCTRL`
- `__I uint32_t SDIO_TypeDef::DCOUNT`

- `__I uint32_t SDIO_TypeDef::STA`
- `__IO uint32_t SDIO_TypeDef::ICR`
- `__IO uint32_t SDIO_TypeDef::MASK`
- `uint32_t SDIO_TypeDef::RESERVED0 [2]`
- `__I uint32_t SDIO_TypeDef::FIFOCNT`
- `uint32_t SDIO_TypeDef::RESERVED1 [13]`
- `__IO uint32_t SPI_TypeDef::FIFO`
- `__IO uint16_t SPI_TypeDef::CR1`
- `uint16_t SPI_TypeDef::RESERVED0`
- `__IO uint16_t SPI_TypeDef::CR2`
- `uint16_t SPI_TypeDef::RESERVED1`
- `__IO uint16_t SPI_TypeDef::SR`
- `uint16_t SPI_TypeDef::RESERVED2`
- `__IO uint16_t SPI_TypeDef::DR`
- `uint16_t SPI_TypeDef::RESERVED3`
- `__IO uint16_t SPI_TypeDef::CRCPR`
- `uint16_t SPI_TypeDef::RESERVED4`
- `__IO uint16_t SPI_TypeDef::RXCCR`
- `uint16_t SPI_TypeDef::RESERVED5`
- `__IO uint16_t SPI_TypeDef::TXCCR`
- `uint16_t SPI_TypeDef::RESERVED6`
- `__IO uint16_t SPI_TypeDef::I2SCFGR`
- `uint16_t SPI_TypeDef::RESERVED7`
- `__IO uint16_t SPI_TypeDef::I2SPR`
- `uint16_t SPI_TypeDef::RESERVED8`
- `__IO uint16_t TIM_TypeDef::CR1`
- `uint16_t TIM_TypeDef::RESERVED0`
- `__IO uint16_t TIM_TypeDef::CR2`
- `uint16_t TIM_TypeDef::RESERVED1`
- `__IO uint16_t TIM_TypeDef::SMCR`
- `uint16_t TIM_TypeDef::RESERVED2`
- `__IO uint16_t TIM_TypeDef::DIER`
- `uint16_t TIM_TypeDef::RESERVED3`
- `__IO uint16_t TIM_TypeDef::SR`
- `uint16_t TIM_TypeDef::RESERVED4`
- `__IO uint16_t TIM_TypeDef::EGR`
- `uint16_t TIM_TypeDef::RESERVED5`
- `__IO uint16_t TIM_TypeDef::CCMR1`
- `uint16_t TIM_TypeDef::RESERVED6`
- `__IO uint16_t TIM_TypeDef::CCMR2`
- `uint16_t TIM_TypeDef::RESERVED7`
- `__IO uint16_t TIM_TypeDef::CCER`
- `uint16_t TIM_TypeDef::RESERVED8`
- `__IO uint32_t TIM_TypeDef::CNT`
- `__IO uint16_t TIM_TypeDef::PSC`
- `uint16_t TIM_TypeDef::RESERVED9`
- `__IO uint32_t TIM_TypeDef::ARR`
- `__IO uint16_t TIM_TypeDef::RCR`
- `uint16_t TIM_TypeDef::RESERVED10`
- `__IO uint32_t TIM_TypeDef::CCR1`
- `__IO uint32_t TIM_TypeDef::CCR2`
- `__IO uint32_t TIM_TypeDef::CCR3`
- `__IO uint32_t TIM_TypeDef::CCR4`
- `__IO uint16_t TIM_TypeDef::BDTR`
- `uint16_t TIM_TypeDef::RESERVED11`

- `__IO uint16_t TIM_TypeDef::DCR`
- `uint16_t TIM_TypeDef::RESERVED12`
- `__IO uint16_t TIM_TypeDef::DMAR`
- `uint16_t TIM_TypeDef::RESERVED13`
- `__IO uint16_t TIM_TypeDef::OR`
- `uint16_t TIM_TypeDef::RESERVED14`
- `__IO uint16_t USART_TypeDef::SR`
- `uint16_t USART_TypeDef::RESERVED0`
- `__IO uint16_t USART_TypeDef::DR`
- `uint16_t USART_TypeDef::RESERVED1`
- `__IO uint16_t USART_TypeDef::BRR`
- `uint16_t USART_TypeDef::RESERVED2`
- `__IO uint16_t USART_TypeDef::CR1`
- `uint16_t USART_TypeDef::RESERVED3`
- `__IO uint16_t USART_TypeDef::CR2`
- `uint16_t USART_TypeDef::RESERVED4`
- `__IO uint16_t USART_TypeDef::CR3`
- `uint16_t USART_TypeDef::RESERVED5`
- `__IO uint16_t USART_TypeDef::GTPR`
- `uint16_t USART_TypeDef::RESERVED6`
- `__IO uint32_t WWDG_TypeDef::CR`
- `__IO uint32_t WWDG_TypeDef::CFR`
- `__IO uint32_t WWDG_TypeDef::SR`
- `__IO uint32_t CRYP_TypeDef::CR`
- `__IO uint32_t CRYP_TypeDef::SR`
- `__IO uint32_t CRYP_TypeDef::DR`
- `__IO uint32_t CRYP_TypeDef::DOUT`
- `__IO uint32_t CRYP_TypeDef::DMACR`
- `__IO uint32_t CRYP_TypeDef::IMSCR`
- `__IO uint32_t CRYP_TypeDef::RISR`
- `__IO uint32_t CRYP_TypeDef::MISR`
- `__IO uint32_t CRYP_TypeDef::K0LR`
- `__IO uint32_t CRYP_TypeDef::K0RR`
- `__IO uint32_t CRYP_TypeDef::K1LR`
- `__IO uint32_t CRYP_TypeDef::K1RR`
- `__IO uint32_t CRYP_TypeDef::K2LR`
- `__IO uint32_t CRYP_TypeDef::K2RR`
- `__IO uint32_t CRYP_TypeDef::K3LR`
- `__IO uint32_t CRYP_TypeDef::K3RR`
- `__IO uint32_t CRYP_TypeDef::IV0LR`
- `__IO uint32_t CRYP_TypeDef::IV0RR`
- `__IO uint32_t CRYP_TypeDef::IV1LR`
- `__IO uint32_t CRYP_TypeDef::IV1RR`
- `__IO uint32_t CRYP_TypeDef::CSGCMCCM0R`
- `__IO uint32_t CRYP_TypeDef::CSGCMCCM1R`
- `__IO uint32_t CRYP_TypeDef::CSGCMCCM2R`
- `__IO uint32_t CRYP_TypeDef::CSGCMCCM3R`
- `__IO uint32_t CRYP_TypeDef::CSGCMCCM4R`
- `__IO uint32_t CRYP_TypeDef::CSGCMCCM5R`
- `__IO uint32_t CRYP_TypeDef::CSGCMCCM6R`
- `__IO uint32_t CRYP_TypeDef::CSGCMCCM7R`
- `__IO uint32_t CRYP_TypeDef::CSGCM0R`
- `__IO uint32_t CRYP_TypeDef::CSGCM1R`
- `__IO uint32_t CRYP_TypeDef::CSGCM2R`
- `__IO uint32_t CRYP_TypeDef::CSGCM3R`

- `__IO uint32_t CRYP_TypeDef::CSGCM4R`
- `__IO uint32_t CRYP_TypeDef::CSGCM5R`
- `__IO uint32_t CRYP_TypeDef::CSGCM6R`
- `__IO uint32_t CRYP_TypeDef::CSGCM7R`
- `__IO uint32_t HASH_TypeDef::CR`
- `__IO uint32_t HASH_TypeDef::DIN`
- `__IO uint32_t HASH_TypeDef::STR`
- `__IO uint32_t HASH_TypeDef::HR [5]`
- `__IO uint32_t HASH_TypeDef::IMR`
- `__IO uint32_t HASH_TypeDef::SR`
- `uint32_t HASH_TypeDef::RESERVED [52]`
- `__IO uint32_t HASH_TypeDef::CSR [54]`
- `__IO uint32_t HASH_DIGEST_TypeDef::HR [8]`
- `__IO uint32_t RNG_TypeDef::CR`
- `__IO uint32_t RNG_TypeDef::SR`
- `__IO uint32_t RNG_TypeDef::DR`

- `volatile int32_t ITM_RxBuffer`
- `__STATIC_INLINE void NVIC_SetPriorityGrouping (uint32_t PriorityGroup)`

Set Priority Grouping.
- `__STATIC_INLINE uint32_t NVIC_GetPriorityGrouping (void)`

Get Priority Grouping.
- `__STATIC_INLINE void NVIC_EnableIRQ (IRQn_Type IRQn)`

Enable External Interrupt.
- `__STATIC_INLINE void NVIC_DisableIRQ (IRQn_Type IRQn)`

Disable External Interrupt.
- `__STATIC_INLINE uint32_t NVIC_GetPendingIRQ (IRQn_Type IRQn)`

Get Pending Interrupt.
- `__STATIC_INLINE void NVIC_SetPendingIRQ (IRQn_Type IRQn)`

Set Pending Interrupt.
- `__STATIC_INLINE void NVIC_ClearPendingIRQ (IRQn_Type IRQn)`

Clear Pending Interrupt.
- `__STATIC_INLINE uint32_t NVIC_GetActive (IRQn_Type IRQn)`

Get Active Interrupt.
- `__STATIC_INLINE void NVIC_SetPriority (IRQn_Type IRQn, uint32_t priority)`

Set Interrupt Priority.
- `__STATIC_INLINE uint32_t NVIC_GetPriority (IRQn_Type IRQn)`

Get Interrupt Priority.
- `__STATIC_INLINE uint32_t NVIC_EncodePriority (uint32_t PriorityGroup, uint32_t PreemptPriority, uint32_t SubPriority)`

Encode Priority.
- `__STATIC_INLINE void NVIC_DecodePriority (uint32_t Priority, uint32_t PriorityGroup, uint32_t *pPreemptPriority, uint32_t *pSubPriority)`

Decode Priority.
- `__STATIC_INLINE void NVIC_SystemReset (void)`

System Reset.
- `__STATIC_INLINE uint32_t SysTick_Config (uint32_t ticks)`

System Tick Configuration.
- `__STATIC_INLINE uint32_t ITM_SendChar (uint32_t ch)`

ITM Send Character.
- `__STATIC_INLINE int32_t ITM_ReceiveChar (void)`

ITM Receive Character.

- `__STATIC_INLINE int32_t ITM_CheckChar (void)`
ITM Check Character.
- `#define __CM4_CMSIS_VERSION_MAIN (0x04)`
- `#define __CM4_CMSIS_VERSION_SUB (0x00)`
- `#define __CM4_CMSIS_VERSION`
- `#define __CORTEX_M (0x04)`
- `#define __CORE_CM4_H_DEPENDANT`
- `#define __I volatile const`
- `#define __O volatile`
- `#define __IO volatile`
- `#define APSR_N_Pos 31`
- `#define APSR_N_Msk (1UL << APSR_N_Pos)`
- `#define APSR_Z_Pos 30`
- `#define APSR_Z_Msk (1UL << APSR_Z_Pos)`
- `#define APSR_C_Pos 29`
- `#define APSR_C_Msk (1UL << APSR_C_Pos)`
- `#define APSR_V_Pos 28`
- `#define APSR_V_Msk (1UL << APSR_V_Pos)`
- `#define APSR_Q_Pos 27`
- `#define APSR_Q_Msk (1UL << APSR_Q_Pos)`
- `#define APSR_GE_Pos 16`
- `#define APSR_GE_Msk (0xFUL << APSR_GE_Pos)`
- `#define IPSR_ISR_Pos 0`
- `#define IPSR_ISR_Msk (0x1FFUL /*<< IPSR_ISR_Pos*/)`
- `#define xPSR_N_Pos 31`
- `#define xPSR_N_Msk (1UL << xPSR_N_Pos)`
- `#define xPSR_Z_Pos 30`
- `#define xPSR_Z_Msk (1UL << xPSR_Z_Pos)`
- `#define xPSR_C_Pos 29`
- `#define xPSR_C_Msk (1UL << xPSR_C_Pos)`
- `#define xPSR_V_Pos 28`
- `#define xPSR_V_Msk (1UL << xPSR_V_Pos)`
- `#define xPSR_Q_Pos 27`
- `#define xPSR_Q_Msk (1UL << xPSR_Q_Pos)`
- `#define xPSR_IT_Pos 25`
- `#define xPSR_IT_Msk (3UL << xPSR_IT_Pos)`
- `#define xPSR_T_Pos 24`
- `#define xPSR_T_Msk (1UL << xPSR_T_Pos)`
- `#define xPSR_GE_Pos 16`
- `#define xPSR_GE_Msk (0xFUL << xPSR_GE_Pos)`
- `#define xPSR_ISR_Pos 0`
- `#define xPSR_ISR_Msk (0x1FFUL /*<< xPSR_ISR_Pos*/)`
- `#define CONTROL_FPCA_Pos 2`
- `#define CONTROL_FPCA_Msk (1UL << CONTROL_FPCA_Pos)`
- `#define CONTROL_SPSEL_Pos 1`
- `#define CONTROL_SPSEL_Msk (1UL << CONTROL_SPSEL_Pos)`
- `#define CONTROL_nPRIV_Pos 0`
- `#define CONTROL_nPRIV_Msk (1UL /*<< CONTROL_nPRIV_Pos*/)`
- `#define NVIC_STIR_INTID_Pos 0`
- `#define NVIC_STIR_INTID_Msk (0x1FFUL /*<< NVIC_STIR_INTID_Pos*/)`
- `#define SCB_CPUID_IMPLEMENTER_Pos 24`
- `#define SCB_CPUID_IMPLEMENTER_Msk (0xFFUL << SCB_CPUID_IMPLEMENTER_Pos)`
- `#define SCB_CPUID_VARIANT_Pos 20`
- `#define SCB_CPUID_VARIANT_Msk (0xFUL << SCB_CPUID_VARIANT_Pos)`

- #define SCB_CPUID_ARCHITECTURE_Pos 16
- #define SCB_CPUID_ARCHITECTURE_Msk (0xFUL << SCB_CPUID_ARCHITECTURE_Pos)
- #define SCB_CPUID_PARTNO_Pos 4
- #define SCB_CPUID_PARTNO_Msk (0xFFFFUL << SCB_CPUID_PARTNO_Pos)
- #define SCB_CPUID_REVISION_Pos 0
- #define SCB_CPUID_REVISION_Msk (0xFUL /*<< SCB_CPUID_REVISION_Pos*/)
- #define SCB_ICSR_NMIPENDSET_Pos 31
- #define SCB_ICSR_NMIPENDSET_Msk (1UL << SCB_ICSR_NMIPENDSET_Pos)
- #define SCB_ICSR_PENDSVSET_Pos 28
- #define SCB_ICSR_PENDSVSET_Msk (1UL << SCB_ICSR_PENDSVSET_Pos)
- #define SCB_ICSR_PENDSVCLR_Pos 27
- #define SCB_ICSR_PENDSVCLR_Msk (1UL << SCB_ICSR_PENDSVCLR_Pos)
- #define SCB_ICSR_PENDSTSET_Pos 26
- #define SCB_ICSR_PENDSTSET_Msk (1UL << SCB_ICSR_PENDSTSET_Pos)
- #define SCB_ICSR_PENDSTCLR_Pos 25
- #define SCB_ICSR_PENDSTCLR_Msk (1UL << SCB_ICSR_PENDSTCLR_Pos)
- #define SCB_ICSR_ISRPREEMPT_Pos 23
- #define SCB_ICSR_ISRPREEMPT_Msk (1UL << SCB_ICSR_ISRPREEMPT_Pos)
- #define SCB_ICSR_ISRPENDING_Pos 22
- #define SCB_ICSR_ISRPENDING_Msk (1UL << SCB_ICSR_ISRPENDING_Pos)
- #define SCB_ICSR_VECTPENDING_Pos 12
- #define SCB_ICSR_VECTPENDING_Msk (0x1FFUL << SCB_ICSR_VECTPENDING_Pos)
- #define SCB_ICSR_RETTOBASE_Pos 11
- #define SCB_ICSR_RETTOBASE_Msk (1UL << SCB_ICSR_RETTOBASE_Pos)
- #define SCB_ICSR_VECTACTIVE_Pos 0
- #define SCB_ICSR_VECTACTIVE_Msk (0x1FFUL /*<< SCB_ICSR_VECTACTIVE_Pos*/)
- #define SCB_VTOR_TBLOFF_Pos 7
- #define SCB_VTOR_TBLOFF_Msk (0x1FFFFFFFUL << SCB_VTOR_TBLOFF_Pos)
- #define SCB_AIRCR_VECTKEY_Pos 16
- #define SCB_AIRCR_VECTKEY_Msk (0xFFFFUL << SCB_AIRCR_VECTKEY_Pos)
- #define SCB_AIRCR_VECTKEYSTAT_Pos 16
- #define SCB_AIRCR_VECTKEYSTAT_Msk (0xFFFFUL << SCB_AIRCR_VECTKEYSTAT_Pos)
- #define SCB_AIRCR_ENDIANESS_Pos 15
- #define SCB_AIRCR_ENDIANESS_Msk (1UL << SCB_AIRCR_ENDIANESS_Pos)
- #define SCB_AIRCR_PRIGROUP_Pos 8
- #define SCB_AIRCR_PRIGROUP_Msk (7UL << SCB_AIRCR_PRIGROUP_Pos)
- #define SCB_AIRCR_SYSRESETREQ_Pos 2
- #define SCB_AIRCR_SYSRESETREQ_Msk (1UL << SCB_AIRCR_SYSRESETREQ_Pos)
- #define SCB_AIRCR_VECTCLRACTIVE_Pos 1
- #define SCB_AIRCR_VECTCLRACTIVE_Msk (1UL << SCB_AIRCR_VECTCLRACTIVE_Pos)
- #define SCB_AIRCR_VECTRESET_Pos 0
- #define SCB_AIRCR_VECTRESET_Msk (1UL /*<< SCB_AIRCR_VECTRESET_Pos*/)
- #define SCB_SCR_SEVONPEND_Pos 4
- #define SCB_SCR_SEVONPEND_Msk (1UL << SCB_SCR_SEVONPEND_Pos)
- #define SCB_SCR_SLEEPDEEP_Pos 2
- #define SCB_SCR_SLEEPDEEP_Msk (1UL << SCB_SCR_SLEEPDEEP_Pos)
- #define SCB_SCR_SLEEPONEXIT_Pos 1
- #define SCB_SCR_SLEEPONEXIT_Msk (1UL << SCB_SCR_SLEEPONEXIT_Pos)
- #define SCB_CCR_STKALIGN_Pos 9
- #define SCB_CCR_STKALIGN_Msk (1UL << SCB_CCR_STKALIGN_Pos)
- #define SCB_CCR_BFHFMIGN_Pos 8
- #define SCB_CCR_BFHFMIGN_Msk (1UL << SCB_CCR_BFHFMIGN_Pos)
- #define SCB_CCR_DIV_0_TRP_Pos 4
- #define SCB_CCR_DIV_0_TRP_Msk (1UL << SCB_CCR_DIV_0_TRP_Pos)
- #define SCB_CCR_UNALIGN_TRP_Pos 3

- #define SCB_CCR_UNALIGN_TRP_Msk (1UL << SCB_CCR_UNALIGN_TRP_Pos)
- #define SCB_CCR_USERSETMPEND_Pos 1
- #define SCB_CCR_USERSETMPEND_Msk (1UL << SCB_CCR_USERSETMPEND_Pos)
- #define SCB_CCR_NONBASETHRDENA_Pos 0
- #define SCB_CCR_NONBASETHRDENA_Msk (1UL /*<< SCB_CCR_NONBASETHRDENA_Pos*/)
- #define SCB_SHCSR_USGFAULTENA_Pos 18
- #define SCB_SHCSR_USGFAULTENA_Msk (1UL << SCB_SHCSR_USGFAULTENA_Pos)
- #define SCB_SHCSR_BUSFAULTENA_Pos 17
- #define SCB_SHCSR_BUSFAULTENA_Msk (1UL << SCB_SHCSR_BUSFAULTENA_Pos)
- #define SCB_SHCSR_MEMFAULTENA_Pos 16
- #define SCB_SHCSR_MEMFAULTENA_Msk (1UL << SCB_SHCSR_MEMFAULTENA_Pos)
- #define SCB_SHCSR_SVCALLPENDED_Pos 15
- #define SCB_SHCSR_SVCALLPENDED_Msk (1UL << SCB_SHCSR_SVCALLPENDED_Pos)
- #define SCB_SHCSR_BUSFAULTPENDED_Pos 14
- #define SCB_SHCSR_BUSFAULTPENDED_Msk (1UL << SCB_SHCSR_BUSFAULTPENDED_Pos)
- #define SCB_SHCSR_MEMFAULTPENDED_Pos 13
- #define SCB_SHCSR_MEMFAULTPENDED_Msk (1UL << SCB_SHCSR_MEMFAULTPENDED_Pos)
- #define SCB_SHCSR_USGFAULTPENDED_Pos 12
- #define SCB_SHCSR_USGFAULTPENDED_Msk (1UL << SCB_SHCSR_USGFAULTPENDED_Pos)
- #define SCB_SHCSR_SYSTICKACT_Pos 11
- #define SCB_SHCSR_SYSTICKACT_Msk (1UL << SCB_SHCSR_SYSTICKACT_Pos)
- #define SCB_SHCSR_PENDSVACT_Pos 10
- #define SCB_SHCSR_PENDSVACT_Msk (1UL << SCB_SHCSR_PENDSVACT_Pos)
- #define SCB_SHCSR_MONITORACT_Pos 8
- #define SCB_SHCSR_MONITORACT_Msk (1UL << SCB_SHCSR_MONITORACT_Pos)
- #define SCB_SHCSR_SVCALLACT_Pos 7
- #define SCB_SHCSR_SVCALLACT_Msk (1UL << SCB_SHCSR_SVCALLACT_Pos)
- #define SCB_SHCSR_USGFAULTACT_Pos 3
- #define SCB_SHCSR_USGFAULTACT_Msk (1UL << SCB_SHCSR_USGFAULTACT_Pos)
- #define SCB_SHCSR_BUSFAULTACT_Pos 1
- #define SCB_SHCSR_BUSFAULTACT_Msk (1UL << SCB_SHCSR_BUSFAULTACT_Pos)
- #define SCB_SHCSR_MEMFAULTACT_Pos 0
- #define SCB_SHCSR_MEMFAULTACT_Msk (1UL /*<< SCB_SHCSR_MEMFAULTACT_Pos*/)
- #define SCB_CFSR_USGFAULTSR_Pos 16
- #define SCB_CFSR_USGFAULTSR_Msk (0xFFFFUL << SCB_CFSR_USGFAULTSR_Pos)
- #define SCB_CFSR_BUSFAULTSR_Pos 8
- #define SCB_CFSR_BUSFAULTSR_Msk (0xFFUL << SCB_CFSR_BUSFAULTSR_Pos)
- #define SCB_CFSR_MEMFAULTSR_Pos 0
- #define SCB_CFSR_MEMFAULTSR_Msk (0xFFUL /*<< SCB_CFSR_MEMFAULTSR_Pos*/)
- #define SCB_HFSR_DEBUGEVT_Pos 31
- #define SCB_HFSR_DEBUGEVT_Msk (1UL << SCB_HFSR_DEBUGEVT_Pos)
- #define SCB_HFSR_FORCED_Pos 30
- #define SCB_HFSR_FORCED_Msk (1UL << SCB_HFSR_FORCED_Pos)
- #define SCB_HFSR_VECTTBL_Pos 1
- #define SCB_HFSR_VECTTBL_Msk (1UL << SCB_HFSR_VECTTBL_Pos)
- #define SCB_DFSR_EXTERNAL_Pos 4
- #define SCB_DFSR_EXTERNAL_Msk (1UL << SCB_DFSR_EXTERNAL_Pos)
- #define SCB_DFSR_VCATCH_Pos 3
- #define SCB_DFSR_VCATCH_Msk (1UL << SCB_DFSR_VCATCH_Pos)
- #define SCB_DFSR_DWTTRAP_Pos 2
- #define SCB_DFSR_DWTTRAP_Msk (1UL << SCB_DFSR_DWTTRAP_Pos)
- #define SCB_DFSR_BKPT_Pos 1
- #define SCB_DFSR_BKPT_Msk (1UL << SCB_DFSR_BKPT_Pos)
- #define SCB_DFSR_HALTED_Pos 0
- #define SCB_DFSR_HALTED_Msk (1UL /*<< SCB_DFSR_HALTED_Pos*/)

- #define SCnSCB_ICTR_INTLINESNUM_Pos 0
- #define SCnSCB_ICTR_INTLINESNUM_Msk (0xFUL /*<< SCnSCB_ICTR_INTLINESNUM_Pos*/)
- #define SCnSCB_ACTLR_DISOOPF_Pos 9
- #define SCnSCB_ACTLR_DISOOPF_Msk (1UL << SCnSCB_ACTLR_DISOOPF_Pos)
- #define SCnSCB_ACTLR_DISFPCA_Pos 8
- #define SCnSCB_ACTLR_DISFPCA_Msk (1UL << SCnSCB_ACTLR_DISFPCA_Pos)
- #define SCnSCB_ACTLR_DISFOLD_Pos 2
- #define SCnSCB_ACTLR_DISFOLD_Msk (1UL << SCnSCB_ACTLR_DISFOLD_Pos)
- #define SCnSCB_ACTLR_DISDEFWBUFF_Pos 1
- #define SCnSCB_ACTLR_DISDEFWBUFF_Msk (1UL << SCnSCB_ACTLR_DISDEFWBUFF_Pos)
- #define SCnSCB_ACTLR_DISMCYCINT_Pos 0
- #define SCnSCB_ACTLR_DISMCYCINT_Msk (1UL /*<< SCnSCB_ACTLR_DISMCYCINT_Pos*/)
- #define SysTick_CTRL_COUNTFLAG_Pos 16
- #define SysTick_CTRL_COUNTFLAG_Msk (1UL << SysTick_CTRL_COUNTFLAG_Pos)
- #define SysTick_CTRL_CLKSOURCE_Pos 2
- #define SysTick_CTRL_CLKSOURCE_Msk (1UL << SysTick_CTRL_CLKSOURCE_Pos)
- #define SysTick_CTRL_TICKINT_Pos 1
- #define SysTick_CTRL_TICKINT_Msk (1UL << SysTick_CTRL_TICKINT_Pos)
- #define SysTick_CTRL_ENABLE_Pos 0
- #define SysTick_CTRL_ENABLE_Msk (1UL /*<< SysTick_CTRL_ENABLE_Pos*/)
- #define SysTick_LOAD_RELOAD_Pos 0
- #define SysTick_LOAD_RELOAD_Msk (0xFFFFFUL /*<< SysTick_LOAD_RELOAD_Pos*/)
- #define SysTick_VAL_CURRENT_Pos 0
- #define SysTick_VAL_CURRENT_Msk (0xFFFFFUL /*<< SysTick_VAL_CURRENT_Pos*/)
- #define SysTick_CALIB_NOREF_Pos 31
- #define SysTick_CALIB_NOREF_Msk (1UL << SysTick_CALIB_NOREF_Pos)
- #define SysTick_CALIB_SKEW_Pos 30
- #define SysTick_CALIB_SKEW_Msk (1UL << SysTick_CALIB_SKEW_Pos)
- #define SysTick_CALIB_TENMS_Pos 0
- #define SysTick_CALIB_TENMS_Msk (0xFFFFFUL /*<< SysTick_CALIB_TENMS_Pos*/)
- #define ITM_TPR_PRIVMASK_Pos 0
- #define ITM_TPR_PRIVMASK_Msk (0xFUL /*<< ITM_TPR_PRIVMASK_Pos*/)
- #define ITM_TCR_BUSY_Pos 23
- #define ITM_TCR_BUSY_Msk (1UL << ITM_TCR_BUSY_Pos)
- #define ITM_TCR_TraceBusID_Pos 16
- #define ITM_TCR_TraceBusID_Msk (0x7FUL << ITM_TCR_TraceBusID_Pos)
- #define ITM_TCR_GTSFREQ_Pos 10
- #define ITM_TCR_GTSFREQ_Msk (3UL << ITM_TCR_GTSFREQ_Pos)
- #define ITM_TCR_TSPrescale_Pos 8
- #define ITM_TCR_TSPrescale_Msk (3UL << ITM_TCR_TSPrescale_Pos)
- #define ITM_TCR_SWOENA_Pos 4
- #define ITM_TCR_SWOENA_Msk (1UL << ITM_TCR_SWOENA_Pos)
- #define ITM_TCR_DWTENA_Pos 3
- #define ITM_TCR_DWTENA_Msk (1UL << ITM_TCR_DWTENA_Pos)
- #define ITM_TCR_SYNCENA_Pos 2
- #define ITM_TCR_SYNCENA_Msk (1UL << ITM_TCR_SYNCENA_Pos)
- #define ITM_TCR_TSENA_Pos 1
- #define ITM_TCR_TSENA_Msk (1UL << ITM_TCR_TSENA_Pos)
- #define ITM_TCR_ITMENA_Pos 0
- #define ITM_TCR_ITMENA_Msk (1UL /*<< ITM_TCR_ITMENA_Pos*/)
- #define ITM_IWR_ATVALIDDM_Pos 0
- #define ITM_IWR_ATVALIDDM_Msk (1UL /*<< ITM_IWR_ATVALIDDM_Pos*/)
- #define ITM_IRR_ATREADYM_Pos 0
- #define ITM_IRR_ATREADYM_Msk (1UL /*<< ITM_IRR_ATREADYM_Pos*/)
- #define ITM_IMCR_INTEGRATION_Pos 0

- #define ITM_IMCR_INTEGRATION_Msk (1UL /*<< ITM_IMCR_INTEGRATION_Pos*/)
- #define ITM_LSR_ByteAcc_Pos 2
- #define ITM_LSR_ByteAcc_Msk (1UL << ITM_LSR_ByteAcc_Pos)
- #define ITM_LSR_Access_Pos 1
- #define ITM_LSR_Access_Msk (1UL << ITM_LSR_Access_Pos)
- #define ITM_LSR_Present_Pos 0
- #define ITM_LSR_Present_Msk (1UL /*<< ITM_LSR_Present_Pos*/)
- #define DWT_CTRL_NUMCOMP_Pos 28
- #define DWT_CTRL_NUMCOMP_Msk (0xFUL << DWT_CTRL_NUMCOMP_Pos)
- #define DWT_CTRL_NOTRCPKT_Pos 27
- #define DWT_CTRL_NOTRCPKT_Msk (0x1UL << DWT_CTRL_NOTRCPKT_Pos)
- #define DWT_CTRL_NOEXTTRIG_Pos 26
- #define DWT_CTRL_NOEXTTRIG_Msk (0x1UL << DWT_CTRL_NOEXTTRIG_Pos)
- #define DWT_CTRL_NOCYCCNT_Pos 25
- #define DWT_CTRL_NOCYCCNT_Msk (0x1UL << DWT_CTRL_NOCYCCNT_Pos)
- #define DWT_CTRL_NOPRFCNT_Pos 24
- #define DWT_CTRL_NOPRFCNT_Msk (0x1UL << DWT_CTRL_NOPRFCNT_Pos)
- #define DWT_CTRL_CYCEVTENA_Pos 22
- #define DWT_CTRL_CYCEVTENA_Msk (0x1UL << DWT_CTRL_CYCEVTENA_Pos)
- #define DWT_CTRL_FOLDEVTENA_Pos 21
- #define DWT_CTRL_FOLDEVTENA_Msk (0x1UL << DWT_CTRL_FOLDEVTENA_Pos)
- #define DWT_CTRL_LSUEVTENA_Pos 20
- #define DWT_CTRL_LSUEVTENA_Msk (0x1UL << DWT_CTRL_LSUEVTENA_Pos)
- #define DWT_CTRL_SLEEPEVTENA_Pos 19
- #define DWT_CTRL_SLEEPEVTENA_Msk (0x1UL << DWT_CTRL_SLEEPEVTENA_Pos)
- #define DWT_CTRL_EXCEVTENA_Pos 18
- #define DWT_CTRL_EXCEVTENA_Msk (0x1UL << DWT_CTRL_EXCEVTENA_Pos)
- #define DWT_CTRL_CPIEVTEA_Pos 17
- #define DWT_CTRL_CPIEVTEA_Msk (0x1UL << DWT_CTRL_CPIEVTEA_Pos)
- #define DWT_CTRL_EXCTRCENA_Pos 16
- #define DWT_CTRL_EXCTRCENA_Msk (0x1UL << DWT_CTRL_EXCTRCENA_Pos)
- #define DWT_CTRL_PCSAMPLENA_Pos 12
- #define DWT_CTRL_PCSAMPLENA_Msk (0x1UL << DWT_CTRL_PCSAMPLENA_Pos)
- #define DWT_CTRL_SYNCTAP_Pos 10
- #define DWT_CTRL_SYNCTAP_Msk (0x3UL << DWT_CTRL_SYNCTAP_Pos)
- #define DWT_CTRL_CYCTAP_Pos 9
- #define DWT_CTRL_CYCTAP_Msk (0x1UL << DWT_CTRL_CYCTAP_Pos)
- #define DWT_CTRL_POSTINIT_Pos 5
- #define DWT_CTRL_POSTINIT_Msk (0xFUL << DWT_CTRL_POSTINIT_Pos)
- #define DWT_CTRL_POSTPRESET_Pos 1
- #define DWT_CTRL_POSTPRESET_Msk (0xFUL << DWT_CTRL_POSTPRESET_Pos)
- #define DWT_CTRL_CYCCNTENA_Pos 0
- #define DWT_CTRL_CYCCNTENA_Msk (0x1UL /*<< DWT_CTRL_CYCCNTENA_Pos*/)
- #define DWT_CPLICNT_CPLICNT_Pos 0
- #define DWT_CPLICNT_CPLICNT_Msk (0xFFUL /*<< DWT_CPLICNT_CPLICNT_Pos*/)
- #define DWT_EXCCNT_EXCCNT_Pos 0
- #define DWT_EXCCNT_EXCCNT_Msk (0xFFUL /*<< DWT_EXCCNT_EXCCNT_Pos*/)
- #define DWT_SLEEPcnt_SLEEPcnt_Pos 0
- #define DWT_SLEEPcnt_SLEEPcnt_Msk (0xFFUL /*<< DWT_SLEEPcnt_SLEEPcnt_Pos*/)
- #define DWT_LSUCNT_LSUCNT_Pos 0
- #define DWT_LSUCNT_LSUCNT_Msk (0xFFUL /*<< DWT_LSUCNT_LSUCNT_Pos*/)
- #define DWT_FOLDCNT_FOLDCNT_Pos 0
- #define DWT_FOLDCNT_FOLDCNT_Msk (0xFFUL /*<< DWT_FOLDCNT_FOLDCNT_Pos*/)
- #define DWT_MASK_MASK_Pos 0
- #define DWT_MASK_MASK_Msk (0x1FUL /*<< DWT_MASK_MASK_Pos*/)

- #define DWT_FUNCTION_MATCHED_Pos 24
- #define DWT_FUNCTION_MATCHED_Msk (0x1UL << DWT_FUNCTION_MATCHED_Pos)
- #define DWT_FUNCTION_DATAVADDR1_Pos 16
- #define DWT_FUNCTION_DATAVADDR1_Msk (0xFUL << DWT_FUNCTION_DATAVADDR1_Pos)
- #define DWT_FUNCTION_DATAVADDR0_Pos 12
- #define DWT_FUNCTION_DATAVADDR0_Msk (0xFUL << DWT_FUNCTION_DATAVADDR0_Pos)
- #define DWT_FUNCTION_DATAVSIZE_Pos 10
- #define DWT_FUNCTION_DATAVSIZE_Msk (0x3UL << DWT_FUNCTION_DATAVSIZE_Pos)
- #define DWT_FUNCTION_LNK1ENA_Pos 9
- #define DWT_FUNCTION_LNK1ENA_Msk (0x1UL << DWT_FUNCTION_LNK1ENA_Pos)
- #define DWT_FUNCTION_DATAVMATCH_Pos 8
- #define DWT_FUNCTION_DATAVMATCH_Msk (0x1UL << DWT_FUNCTION_DATAVMATCH_Pos)
- #define DWT_FUNCTION_CYCMATCH_Pos 7
- #define DWT_FUNCTION_CYCMATCH_Msk (0x1UL << DWT_FUNCTION_CYCMATCH_Pos)
- #define DWT_FUNCTION_EMITRANGE_Pos 5
- #define DWT_FUNCTION_EMITRANGE_Msk (0x1UL << DWT_FUNCTION_EMITRANGE_Pos)
- #define DWT_FUNCTION_FUNCTION_Pos 0
- #define DWT_FUNCTION_FUNCTION_Msk (0xFUL /*<< DWT_FUNCTION_FUNCTION_Pos*/)
- #define TPI_ACPR_PRESCALER_Pos 0
- #define TPI_ACPR_PRESCALER_Msk (0xFFFFUL /*<< TPI_ACPR_PRESCALER_Pos*/)
- #define TPI_SPPR_TXMODE_Pos 0
- #define TPI_SPPR_TXMODE_Msk (0x3UL /*<< TPI_SPPR_TXMODE_Pos*/)
- #define TPI_FFSR_FtNonStop_Pos 3
- #define TPI_FFSR_FtNonStop_Msk (0x1UL << TPI_FFSR_FtNonStop_Pos)
- #define TPI_FFSR_TCPresent_Pos 2
- #define TPI_FFSR_TCPresent_Msk (0x1UL << TPI_FFSR_TCPresent_Pos)
- #define TPI_FFSR_FtStopped_Pos 1
- #define TPI_FFSR_FtStopped_Msk (0x1UL << TPI_FFSR_FtStopped_Pos)
- #define TPI_FFSR_FInProg_Pos 0
- #define TPI_FFSR_FInProg_Msk (0x1UL /*<< TPI_FFSR_FInProg_Pos*/)
- #define TPI_FFCR_TrigIn_Pos 8
- #define TPI_FFCR_TrigIn_Msk (0x1UL << TPI_FFCR_TrigIn_Pos)
- #define TPI_FFCR_EnFCont_Pos 1
- #define TPI_FFCR_EnFCont_Msk (0x1UL << TPI_FFCR_EnFCont_Pos)
- #define TPI_TRIGGER_TRIGGER_Pos 0
- #define TPI_TRIGGER_TRIGGER_Msk (0x1UL /*<< TPI_TRIGGER_TRIGGER_Pos*/)
- #define TPI_FIFO0_ITM_ATVALID_Pos 29
- #define TPI_FIFO0_ITM_ATVALID_Msk (0x3UL << TPI_FIFO0_ITM_ATVALID_Pos)
- #define TPI_FIFO0_ITM_bytectcount_Pos 27
- #define TPI_FIFO0_ITM_bytectcount_Msk (0x3UL << TPI_FIFO0_ITM_bytectcount_Pos)
- #define TPI_FIFO0_ETM_ATVALID_Pos 26
- #define TPI_FIFO0_ETM_ATVALID_Msk (0x3UL << TPI_FIFO0_ETM_ATVALID_Pos)
- #define TPI_FIFO0_ETM_bytectcount_Pos 24
- #define TPI_FIFO0_ETM_bytectcount_Msk (0x3UL << TPI_FIFO0_ETM_bytectcount_Pos)
- #define TPI_FIFO0_ETM2_Pos 16
- #define TPI_FIFO0_ETM2_Msk (0xFFUL << TPI_FIFO0_ETM2_Pos)
- #define TPI_FIFO0_ETM1_Pos 8
- #define TPI_FIFO0_ETM1_Msk (0xFFUL << TPI_FIFO0_ETM1_Pos)
- #define TPI_FIFO0_ETM0_Pos 0
- #define TPI_FIFO0_ETM0_Msk (0xFFUL /*<< TPI_FIFO0_ETM0_Pos*/)
- #define TPI_ITATBCTR2_ATREADY_Pos 0
- #define TPI_ITATBCTR2_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR2_ATREADY_Pos*/)
- #define TPI_FIFO1_ITM_ATVALID_Pos 29
- #define TPI_FIFO1_ITM_ATVALID_Msk (0x3UL << TPI_FIFO1_ITM_ATVALID_Pos)
- #define TPI_FIFO1_ITM_bytectcount_Pos 27

- #define TPI_FIFO1_ITM_bytecount_Msk (0x3UL << TPI_FIFO1_ITM_bytecount_Pos)
- #define TPI_FIFO1_ETM_ATVALID_Pos 26
- #define TPI_FIFO1_ETM_ATVALID_Msk (0x3UL << TPI_FIFO1_ETM_ATVALID_Pos)
- #define TPI_FIFO1_ETM_bytecount_Pos 24
- #define TPI_FIFO1_ETM_bytecount_Msk (0x3UL << TPI_FIFO1_ETM_bytecount_Pos)
- #define TPI_FIFO1_ITM2_Pos 16
- #define TPI_FIFO1_ITM2_Msk (0xFFUL << TPI_FIFO1_ITM2_Pos)
- #define TPI_FIFO1_ITM1_Pos 8
- #define TPI_FIFO1_ITM1_Msk (0xFFUL << TPI_FIFO1_ITM1_Pos)
- #define TPI_FIFO1_ITM0_Pos 0
- #define TPI_FIFO1_ITM0_Msk (0xFFUL /*<< TPI_FIFO1_ITM0_Pos*/)
- #define TPI_ITATBCTR0_ATREADY_Pos 0
- #define TPI_ITATBCTR0_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR0_ATREADY_Pos*/)
- #define TPI_ITCTRL_Mode_Pos 0
- #define TPI_ITCTRL_Mode_Msk (0x1UL /*<< TPI_ITCTRL_Mode_Pos*/)
- #define TPI_DEVID_NRZVALID_Pos 11
- #define TPI_DEVID_NRZVALID_Msk (0x1UL << TPI_DEVID_NRZVALID_Pos)
- #define TPI_DEVID_MANCVALID_Pos 10
- #define TPI_DEVID_MANCVALID_Msk (0x1UL << TPI_DEVID_MANCVALID_Pos)
- #define TPI_DEVID_PTINVALID_Pos 9
- #define TPI_DEVID_PTINVALID_Msk (0x1UL << TPI_DEVID_PTINVALID_Pos)
- #define TPI_DEVID_MinBufSz_Pos 6
- #define TPI_DEVID_MinBufSz_Msk (0x7UL << TPI_DEVID_MinBufSz_Pos)
- #define TPI_DEVID_AsynClkIn_Pos 5
- #define TPI_DEVID_AsynClkIn_Msk (0x1UL << TPI_DEVID_AsynClkIn_Pos)
- #define TPI_DEVID_NrTraceInput_Pos 0
- #define TPI_DEVID_NrTraceInput_Msk (0x1FUL /*<< TPI_DEVID_NrTraceInput_Pos*/)
- #define TPI_DEVTYPE_MajorType_Pos 4
- #define TPI_DEVTYPE_MajorType_Msk (0xFUL << TPI_DEVTYPE_MajorType_Pos)
- #define TPI_DEVTYPE_SubType_Pos 0
- #define TPI_DEVTYPE_SubType_Msk (0xFUL /*<< TPI_DEVTYPE_SubType_Pos*/)
- #define CoreDebug_DHCSR_DBGKEY_Pos 16
- #define CoreDebug_DHCSR_DBGKEY_Msk (0xFFFFUL << CoreDebug_DHCSR_DBGKEY_Pos)
- #define CoreDebug_DHCSR_S_RESET_ST_Pos 25
- #define CoreDebug_DHCSR_S_RESET_ST_Msk (1UL << CoreDebug_DHCSR_S_RESET_ST_Pos)
- #define CoreDebug_DHCSR_S_RETIRE_ST_Pos 24
- #define CoreDebug_DHCSR_S_RETIRE_ST_Msk (1UL << CoreDebug_DHCSR_S_RETIRE_ST_Pos)
- #define CoreDebug_DHCSR_S_LOCKUP_Pos 19
- #define CoreDebug_DHCSR_S_LOCKUP_Msk (1UL << CoreDebug_DHCSR_S_LOCKUP_Pos)
- #define CoreDebug_DHCSR_S_SLEEP_Pos 18
- #define CoreDebug_DHCSR_S_SLEEP_Msk (1UL << CoreDebug_DHCSR_S_SLEEP_Pos)
- #define CoreDebug_DHCSR_S_HALT_Pos 17
- #define CoreDebug_DHCSR_S_HALT_Msk (1UL << CoreDebug_DHCSR_S_HALT_Pos)
- #define CoreDebug_DHCSR_S_REGRDY_Pos 16
- #define CoreDebug_DHCSR_S_REGRDY_Msk (1UL << CoreDebug_DHCSR_S_REGRDY_Pos)
- #define CoreDebug_DHCSR_C_SNAPSTALL_Pos 5
- #define CoreDebug_DHCSR_C_SNAPSTALL_Msk (1UL << CoreDebug_DHCSR_C_SNAPSTALL_Pos)
- #define CoreDebug_DHCSR_C_MASKINTS_Pos 3
- #define CoreDebug_DHCSR_C_MASKINTS_Msk (1UL << CoreDebug_DHCSR_C_MASKINTS_Pos)
- #define CoreDebug_DHCSR_C_STEP_Pos 2
- #define CoreDebug_DHCSR_C_STEP_Msk (1UL << CoreDebug_DHCSR_C_STEP_Pos)
- #define CoreDebug_DHCSR_C_HALT_Pos 1
- #define CoreDebug_DHCSR_C_HALT_Msk (1UL << CoreDebug_DHCSR_C_HALT_Pos)
- #define CoreDebug_DHCSR_C_DEBUGEN_Pos 0
- #define CoreDebug_DHCSR_C_DEBUGEN_Msk (1UL /*<< CoreDebug_DHCSR_C_DEBUGEN_Pos*/)

- #define CoreDebug_DCRSR_REGWnR_Pos 16
- #define CoreDebug_DCRSR_REGWnR_Msk (1UL << CoreDebug_DCRSR_REGWnR_Pos)
- #define CoreDebug_DCRSR_REGSEL_Pos 0
- #define CoreDebug_DCRSR_REGSEL_Msk (0x1FUL /*<< CoreDebug_DCRSR_REGSEL_Pos*/)
- #define CoreDebug_DEMCR_TRCENA_Pos 24
- #define CoreDebug_DEMCR_TRCENA_Msk (1UL << CoreDebug_DEMCR_TRCENA_Pos)
- #define CoreDebug_DEMCR_MON_REQ_Pos 19
- #define CoreDebug_DEMCR_MON_REQ_Msk (1UL << CoreDebug_DEMCR_MON_REQ_Pos)
- #define CoreDebug_DEMCR_MON_STEP_Pos 18
- #define CoreDebug_DEMCR_MON_STEP_Msk (1UL << CoreDebug_DEMCR_MON_STEP_Pos)
- #define CoreDebug_DEMCR_MON_PEND_Pos 17
- #define CoreDebug_DEMCR_MON_PEND_Msk (1UL << CoreDebug_DEMCR_MON_PEND_Pos)
- #define CoreDebug_DEMCR_MON_EN_Pos 16
- #define CoreDebug_DEMCR_MON_EN_Msk (1UL << CoreDebug_DEMCR_MON_EN_Pos)
- #define CoreDebug_DEMCR_VC_HARDERR_Pos 10
- #define CoreDebug_DEMCR_VC_HARDERR_Msk (1UL << CoreDebug_DEMCR_VC_HARDERR_Pos)
- #define CoreDebug_DEMCR_VC_INTERR_Pos 9
- #define CoreDebug_DEMCR_VC_INTERR_Msk (1UL << CoreDebug_DEMCR_VC_INTERR_Pos)
- #define CoreDebug_DEMCR_VC_BUSERR_Pos 8
- #define CoreDebug_DEMCR_VC_BUSERR_Msk (1UL << CoreDebug_DEMCR_VC_BUSERR_Pos)
- #define CoreDebug_DEMCR_VC_STATERR_Pos 7
- #define CoreDebug_DEMCR_VC_STATERR_Msk (1UL << CoreDebug_DEMCR_VC_STATERR_Pos)
- #define CoreDebug_DEMCR_VC_CHKERR_Pos 6
- #define CoreDebug_DEMCR_VC_CHKERR_Msk (1UL << CoreDebug_DEMCR_VC_CHKERR_Pos)
- #define CoreDebug_DEMCR_VC_NOCPERR_Pos 5
- #define CoreDebug_DEMCR_VC_NOCPERR_Msk (1UL << CoreDebug_DEMCR_VC_NOCPERR_Pos)
- #define CoreDebug_DEMCR_VC_MMERR_Pos 4
- #define CoreDebug_DEMCR_VC_MMERR_Msk (1UL << CoreDebug_DEMCR_VC_MMERR_Pos)
- #define CoreDebug_DEMCR_VC_CORERESET_Pos 0
- #define CoreDebug_DEMCR_VC_CORERESET_Msk (1UL /*<< CoreDebug_DEMCR_VC_CORERESET_Pos*/)
- #define SCS_BASE (0xE000E000UL)
- #define ITM_BASE (0xE0000000UL)
- #define DWT_BASE (0xE0001000UL)
- #define TPI_BASE (0xE0040000UL)
- #define CoreDebug_BASE (0xE000EDF0UL)
- #define SysTick_BASE (SCS_BASE + 0x0010UL)
- #define NVIC_BASE (SCS_BASE + 0x0100UL)
- #define SCB_BASE (SCS_BASE + 0x0D00UL)
- #define SCnSCB ((SCnSCB_Type *) SCS_BASE)
- #define SCB ((SCB_Type *) SCB_BASE)
- #define SysTick ((SysTick_Type *) SysTick_BASE)
- #define NVIC ((NVIC_Type *) NVIC_BASE)
- #define ITM ((ITM_Type *) ITM_BASE)
- #define DWT ((DWT_Type *) DWT_BASE)
- #define TPI ((TPI_Type *) TPI_BASE)
- #define CoreDebug ((CoreDebug_Type *) CoreDebug_BASE)
- #define ITM_RXBUFFER_EMPTY 0x5AA55AA5

5.164.1 Detailed Description

5.164.1.1 MISRA-C:2004 Compliance Exceptions

CMSIS violates the following MISRA-C:2004 rules:

- Required Rule 8.5, object/function definition in header file.
Function definitions in header files are used to allow 'inlining'.
- Required Rule 18.4, declaration of union type or object of union type: '{...}'.
Unions are used for effective representation of core registers.
- Advisory Rule 19.7, Function-like macro defined.
Function-like macros are used to allow more efficient code.

5.164.2 Macro Definition Documentation

5.164.2.1 __CM4_CMSIS_VERSION

```
#define __CM4_CMSIS_VERSION
```

Value:

```
((__CM4_CMSIS_VERSION_MAIN << 16) | \
    __CM4_CMSIS_VERSION_SUB)
```

CMSIS HAL version number

5.164.2.2 __CM4_CMSIS_VERSION_MAIN

```
#define __CM4_CMSIS_VERSION_MAIN (0x04)
```

[31:16] CMSIS HAL main version

5.164.2.3 __CM4_CMSIS_VERSION_SUB

```
#define __CM4_CMSIS_VERSION_SUB (0x00)
```

[15:0] CMSIS HAL sub version

5.164.2.4 __CORE_CM4_H_DEPENDANT

```
#define __CORE_CM4_H_DEPENDANT
```

__FPU_USED indicates whether an FPU is used or not. For this, __FPU_PRESENT has to be checked prior to making use of FPU specific registers and functions.

5.164.2.5 __CORTEX_M

```
#define __CORTEX_M (0x04)
```

Cortex-M Core

5.164.2.6 __I

```
#define __I volatile const
```

Defines 'read only' permissions

5.164.2.7 __IO

```
#define __IO volatile
```

Defines 'read / write' permissions

5.164.2.8 __O

```
#define __O volatile
```

Defines 'write only' permissions

5.164.2.9 APSR_C_Msk

```
#define APSR_C_Msk (1UL << APSR_C_Pos)
```

APSR: C Mask

5.164.2.10 APSR_C_Pos

```
#define APSR_C_Pos 29
```

APSR: C Position

5.164.2.11 APSR_GE_Msk

```
#define APSR_GE_Msk (0xFUL << APSR_GE_Pos)
```

APSR: GE Mask

5.164.2.12 APSR_GE_Pos

```
#define APSR_GE_Pos 16
```

APSR: GE Position

5.164.2.13 APSR_N_Msk

```
#define APSR_N_Msk (1UL << APSR_N_Pos)
```

APSR: N Mask

5.164.2.14 APSR_N_Pos

```
#define APSR_N_Pos 31
```

APSR: N Position

5.164.2.15 APSR_Q_Msk

```
#define APSR_Q_Msk (1UL << APSR_Q_Pos)
```

APSR: Q Mask

5.164.2.16 APSR_Q_Pos

```
#define APSR_Q_Pos 27
```

APSR: Q Position

5.164.2.17 APSR_V_Msk

```
#define APSR_V_Msk (1UL << APSR_V_Pos)
```

APSR: V Mask

5.164.2.18 APSR_V_Pos

```
#define APSR_V_Pos 28
```

APSR: V Position

5.164.2.19 APSR_Z_Msk

```
#define APSR_Z_Msk (1UL << APSR_Z_Pos)
```

APSR: Z Mask

5.164.2.20 APSR_Z_Pos

```
#define APSR_Z_Pos 30
```

APSR: Z Position

5.164.2.21 CONTROL_FPCA_Msk

```
#define CONTROL_FPCA_Msk (1UL << CONTROL_FPCA_Pos)
```

CONTROL: FPCA Mask

5.164.2.22 CONTROL_FPCA_Pos

```
#define CONTROL_FPCA_Pos 2
```

CONTROL: FPCA Position

5.164.2.23 CONTROL_nPRIV_Msk

```
#define CONTROL_nPRIV_Msk (1UL /*<< CONTROL_nPRIV_Pos*/)
```

CONTROL: nPRIV Mask

5.164.2.24 CONTROL_nPRIV_Pos

```
#define CONTROL_nPRIV_Pos 0
```

CONTROL: nPRIV Position

5.164.2.25 CONTROL_SPSEL_Msk

```
#define CONTROL_SPSEL_Msk (1UL << CONTROL_SPSEL_Pos)
```

CONTROL: SPSEL Mask

5.164.2.26 CONTROL_SPSEL_Pos

```
#define CONTROL_SPSEL_Pos 1
```

CONTROL: SPSEL Position

5.164.2.27 CoreDebug

```
#define CoreDebug ((CoreDebug_Type *) CoreDebug_BASE)
```

Core Debug configuration struct

5.164.2.28 CoreDebug_BASE

```
#define CoreDebug_BASE (0xE000EDF0UL)
```

Core Debug Base Address

5.164.2.29 CoreDebug_DCRSR_REGSEL_Msk

```
#define CoreDebug_DCRSR_REGSEL_Msk (0x1FUL /*<< CoreDebug_DCRSR_REGSEL_Pos*/)
```

CoreDebug DCRSR: REGSEL Mask

5.164.2.30 CoreDebug_DCRSR_REGSEL_Pos

```
#define CoreDebug_DCRSR_REGSEL_Pos 0
```

CoreDebug DCRSR: REGSEL Position

5.164.2.31 CoreDebug_DCRSR_REGWnR_Msk

```
#define CoreDebug_DCRSR_REGWnR_Msk (1UL << CoreDebug_DCRSR_REGWnR_Pos)
```

CoreDebug DCRSR: REGWnR Mask

5.164.2.32 CoreDebug_DCRSR_REGWnR_Pos

```
#define CoreDebug_DCRSR_REGWnR_Pos 16
```

CoreDebug DCRSR: REGWnR Position

5.164.2.33 CoreDebug_DEMCR_MON_EN_Msk

```
#define CoreDebug_DEMCR_MON_EN_Msk (1UL << CoreDebug_DEMCR_MON_EN_Pos)
```

CoreDebug DEMCR: MON_EN Mask

5.164.2.34 CoreDebug_DEMCR_MON_EN_Pos

```
#define CoreDebug_DEMCR_MON_EN_Pos 16
```

CoreDebug DEMCR: MON_EN Position

5.164.2.35 CoreDebug_DEMCR_MON_PEND_Msk

```
#define CoreDebug_DEMCR_MON_PEND_Msk (1UL << CoreDebug_DEMCR_MON_PEND_Pos)
```

CoreDebug DEMCR: MON_PEND Mask

5.164.2.36 CoreDebug_DEMCR_MON_PEND_Pos

```
#define CoreDebug_DEMCR_MON_PEND_Pos 17
```

CoreDebug DEMCR: MON_PEND Position

5.164.2.37 CoreDebug_DEMCR_MON_REQ_Msk

```
#define CoreDebug_DEMCR_MON_REQ_Msk (1UL << CoreDebug_DEMCR_MON_REQ_Pos)
```

CoreDebug DEMCR: MON_REQ Mask

5.164.2.38 CoreDebug_DEMCR_MON_REQ_Pos

```
#define CoreDebug_DEMCR_MON_REQ_Pos 19
```

CoreDebug DEMCR: MON_REQ Position

5.164.2.39 CoreDebug_DEMCR_MON_STEP_Msk

```
#define CoreDebug_DEMCR_MON_STEP_Msk (1UL << CoreDebug_DEMCR_MON_STEP_Pos)
```

CoreDebug DEMCR: MON_STEP Mask

5.164.2.40 CoreDebug_DEMCR_MON_STEP_Pos

```
#define CoreDebug_DEMCR_MON_STEP_Pos 18
```

CoreDebug DEMCR: MON_STEP Position

5.164.2.41 CoreDebug_DEMCR_TRCENA_Msk

```
#define CoreDebug_DEMCR_TRCENA_Msk (1UL << CoreDebug_DEMCR_TRCENA_Pos)
```

CoreDebug DEMCR: TRCENA Mask

5.164.2.42 CoreDebug_DEMCR_TRCENA_Pos

```
#define CoreDebug_DEMCR_TRCENA_Pos 24
```

CoreDebug DEMCR: TRCENA Position

5.164.2.43 CoreDebug_DEMCR_VC_BUSERR_Msk

```
#define CoreDebug_DEMCR_VC_BUSERR_Msk (1UL << CoreDebug_DEMCR_VC_BUSERR_Pos)
```

CoreDebug DEMCR: VC_BUSERR Mask

5.164.2.44 CoreDebug_DEMCR_VC_BUSERR_Pos

```
#define CoreDebug_DEMCR_VC_BUSERR_Pos 8
```

CoreDebug DEMCR: VC_BUSERR Position

5.164.2.45 CoreDebug_DEMCR_VC_CHKERR_Msk

```
#define CoreDebug_DEMCR_VC_CHKERR_Msk (1UL << CoreDebug_DEMCR_VC_CHKERR_Pos)
```

CoreDebug DEMCR: VC_CHKERR Mask

5.164.2.46 CoreDebug_DEMCR_VC_CHKERR_Pos

```
#define CoreDebug_DEMCR_VC_CHKERR_Pos 6
```

CoreDebug DEMCR: VC_CHKERR Position

5.164.2.47 CoreDebug_DEMCR_VC_CORERESET_Msk

```
#define CoreDebug_DEMCR_VC_CORERESET_Msk (1UL /*<< CoreDebug_DEMCR_VC_CORERESET_Pos*/)
```

CoreDebug DEMCR: VC_CORERESET Mask

5.164.2.48 CoreDebug_DEMCR_VC_CORERESET_Pos

```
#define CoreDebug_DEMCR_VC_CORERESET_Pos 0
```

CoreDebug DEMCR: VC_CORERESET Position

5.164.2.49 CoreDebug_DEMCR_VC_HARDERR_Msk

```
#define CoreDebug_DEMCR_VC_HARDERR_Msk (1UL << CoreDebug_DEMCR_VC_HARDERR_Pos)
```

CoreDebug DEMCR: VC_HARDERR Mask

5.164.2.50 CoreDebug_DEMCR_VC_HARDERR_Pos

```
#define CoreDebug_DEMCR_VC_HARDERR_Pos 10
```

CoreDebug DEMCR: VC_HARDERR Position

5.164.2.51 CoreDebug_DEMCR_VC_INTERR_Msk

```
#define CoreDebug_DEMCR_VC_INTERR_Msk (1UL << CoreDebug_DEMCR_VC_INTERR_Pos)
```

CoreDebug DEMCR: VC_INTERR Mask

5.164.2.52 CoreDebug_DEMCR_VC_INTERR_Pos

```
#define CoreDebug_DEMCR_VC_INTERR_Pos 9
```

CoreDebug DEMCR: VC_INTERR Position

5.164.2.53 CoreDebug_DEMCR_VC_MMERR_Msk

```
#define CoreDebug_DEMCR_VC_MMERR_Msk (1UL << CoreDebug_DEMCR_VC_MMERR_Pos)
```

CoreDebug DEMCR: VC_MMERR Mask

5.164.2.54 CoreDebug_DEMCR_VC_MMERR_Pos

```
#define CoreDebug_DEMCR_VC_MMERR_Pos 4
```

CoreDebug DEMCR: VC_MMERR Position

5.164.2.55 CoreDebug_DEMCR_VC_NOCPERR_Msk

```
#define CoreDebug_DEMCR_VC_NOCPERR_Msk (1UL << CoreDebug_DEMCR_VC_NOCPERR_Pos)
```

CoreDebug DEMCR: VC_NOCPERR Mask

5.164.2.56 CoreDebug_DEMCR_VC_NOCPERR_Pos

```
#define CoreDebug_DEMCR_VC_NOCPERR_Pos 5
```

CoreDebug DEMCR: VC_NOCPERR Position

5.164.2.57 CoreDebug_DEMCR_VC_STATERR_Msk

```
#define CoreDebug_DEMCR_VC_STATERR_Msk (1UL << CoreDebug_DEMCR_VC_STATERR_Pos)
```

CoreDebug DEMCR: VC_STATERR Mask

5.164.2.58 CoreDebug_DEMCR_VC_STATERR_Pos

```
#define CoreDebug_DEMCR_VC_STATERR_Pos 7
```

CoreDebug DEMCR: VC_STATERR Position

5.164.2.59 CoreDebug_DHCSR_C_DEBUGEN_Msk

```
#define CoreDebug_DHCSR_C_DEBUGEN_Msk (1UL /*<< CoreDebug_DHCSR_C_DEBUGEN_Pos*/)
```

CoreDebug DHCSR: C_DEBUGEN Mask

5.164.2.60 CoreDebug_DHCSR_C_DEBUGEN_Pos

```
#define CoreDebug_DHCSR_C_DEBUGEN_Pos 0
```

CoreDebug DHCSR: C_DEBUGEN Position

5.164.2.61 CoreDebug_DHCSR_C_HALT_Msk

```
#define CoreDebug_DHCSR_C_HALT_Msk (1UL << CoreDebug_DHCSR_C_HALT_Pos)
```

CoreDebug DHCSR: C_HALT Mask

5.164.2.62 CoreDebug_DHCSR_C_HALT_Pos

```
#define CoreDebug_DHCSR_C_HALT_Pos 1
```

CoreDebug DHCSR: C_HALT Position

5.164.2.63 CoreDebug_DHCSR_C_MASKINTS_Msk

```
#define CoreDebug_DHCSR_C_MASKINTS_Msk (1UL << CoreDebug_DHCSR_C_MASKINTS_Pos)
```

CoreDebug DHCSR: C_MASKINTS Mask

5.164.2.64 CoreDebug_DHCSR_C_MASKINTS_Pos

```
#define CoreDebug_DHCSR_C_MASKINTS_Pos 3
```

CoreDebug DHCSR: C_MASKINTS Position

5.164.2.65 CoreDebug_DHCSR_C_SNAPSTALL_Msk

```
#define CoreDebug_DHCSR_C_SNAPSTALL_Msk (1UL << CoreDebug_DHCSR_C_SNAPSTALL_Pos)
```

CoreDebug DHCSR: C_SNAPSTALL Mask

5.164.2.66 CoreDebug_DHCSR_C_SNAPSTALL_Pos

```
#define CoreDebug_DHCSR_C_SNAPSTALL_Pos 5
```

CoreDebug DHCSR: C_SNAPSTALL Position

5.164.2.67 CoreDebug_DHCSR_C_STEP_Msk

```
#define CoreDebug_DHCSR_C_STEP_Msk (1UL << CoreDebug_DHCSR_C_STEP_Pos)
```

CoreDebug DHCSR: C_STEP Mask

5.164.2.68 CoreDebug_DHCSR_C_STEP_Pos

```
#define CoreDebug_DHCSR_C_STEP_Pos 2
```

CoreDebug DHCSR: C_STEP Position

5.164.2.69 CoreDebug_DHCSR_DBGKEY_Msk

```
#define CoreDebug_DHCSR_DBGKEY_Msk (0xFFFFUL << CoreDebug_DHCSR_DBGKEY_Pos)
```

CoreDebug DHCSR: DBGKEY Mask

5.164.2.70 CoreDebug_DHCSR_DBGKEY_Pos

```
#define CoreDebug_DHCSR_DBGKEY_Pos 16
```

CoreDebug DHCSR: DBGKEY Position

5.164.2.71 CoreDebug_DHCSR_S_HALT_Msk

```
#define CoreDebug_DHCSR_S_HALT_Msk (1UL << CoreDebug_DHCSR_S_HALT_Pos)
```

CoreDebug DHCSR: S_HALT Mask

5.164.2.72 CoreDebug_DHCSR_S_HALT_Pos

```
#define CoreDebug_DHCSR_S_HALT_Pos 17
```

CoreDebug DHCSR: S_HALT Position

5.164.2.73 CoreDebug_DHCSR_S_LOCKUP_Msk

```
#define CoreDebug_DHCSR_S_LOCKUP_Msk (1UL << CoreDebug_DHCSR_S_LOCKUP_Pos)
```

CoreDebug DHCSR: S_LOCKUP Mask

5.164.2.74 CoreDebug_DHCSR_S_LOCKUP_Pos

```
#define CoreDebug_DHCSR_S_LOCKUP_Pos 19
```

CoreDebug DHCSR: S_LOCKUP Position

5.164.2.75 CoreDebug_DHCSR_S_REGRDY_Msk

```
#define CoreDebug_DHCSR_S_REGRDY_Msk (1UL << CoreDebug_DHCSR_S_REGRDY_Pos)
```

CoreDebug DHCSR: S_REGRDY Mask

5.164.2.76 CoreDebug_DHCSR_S_REGRDY_Pos

```
#define CoreDebug_DHCSR_S_REGRDY_Pos 16
```

CoreDebug DHCSR: S_REGRDY Position

5.164.2.77 CoreDebug_DHCSR_S_RESET_ST_Msk

```
#define CoreDebug_DHCSR_S_RESET_ST_Msk (1UL << CoreDebug_DHCSR_S_RESET_ST_Pos)
```

CoreDebug DHCSR: S_RESET_ST Mask

5.164.2.78 CoreDebug_DHCSR_S_RESET_ST_Pos

```
#define CoreDebug_DHCSR_S_RESET_ST_Pos 25
```

CoreDebug DHCSR: S_RESET_ST Position

5.164.2.79 CoreDebug_DHCSR_S_RETIRE_ST_Msk

```
#define CoreDebug_DHCSR_S_RETIRE_ST_Msk (1UL << CoreDebug_DHCSR_S_RETIRE_ST_Pos)
```

CoreDebug DHCSR: S_RETIRE_ST Mask

5.164.2.80 CoreDebug_DHCSR_S_RETIRE_ST_Pos

```
#define CoreDebug_DHCSR_S_RETIRE_ST_Pos 24
```

CoreDebug DHCSR: S_RETIRE_ST Position

5.164.2.81 CoreDebug_DHCSR_S_SLEEP_Msk

```
#define CoreDebug_DHCSR_S_SLEEP_Msk (1UL << CoreDebug_DHCSR_S_SLEEP_Pos)
```

CoreDebug DHCSR: S_SLEEP Mask

5.164.2.82 CoreDebug_DHCSR_S_SLEEP_Pos

```
#define CoreDebug_DHCSR_S_SLEEP_Pos 18
```

CoreDebug DHCSR: S_SLEEP Position

5.164.2.83 DWT

```
#define DWT ((DWT_Type *) DWT_BASE)
```

DWT configuration struct

5.164.2.84 DWT_BASE

```
#define DWT_BASE (0xE0001000UL)
```

DWT Base Address

5.164.2.85 DWT_CPICNT_CPICNT_Msk

```
#define DWT_CPICNT_CPICNT_Msk (0xFFUL /*<< DWT_CPICNT_CPICNT_Pos*/)
```

DWT CPICNT: CPICNT Mask

5.164.2.86 DWT_CPICNT_CPICNT_Pos

```
#define DWT_CPICNT_CPICNT_Pos 0
```

DWT CPICNT: CPICNT Position

5.164.2.87 DWT_CTRL_CPIEVTENA_Msk

```
#define DWT_CTRL_CPIEVTENA_Msk (0x1UL << DWT_CTRL_CPIEVTENA_Pos)
```

DWT CTRL: CPIEVTENA Mask

5.164.2.88 DWT_CTRL_CPIEVTENA_Pos

```
#define DWT_CTRL_CPIEVTENA_Pos 17
```

DWT CTRL: CPIEVTENA Position

5.164.2.89 DWT_CTRL_CYCCNTENA_Msk

```
#define DWT_CTRL_CYCCNTENA_Msk (0x1UL /*<< DWT_CTRL_CYCCNTENA_Pos*/)
```

DWT CTRL: CYCCNTENA Mask

5.164.2.90 DWT_CTRL_CYCCNTENA_Pos

```
#define DWT_CTRL_CYCCNTENA_Pos 0
```

DWT CTRL: CYCCNTENA Position

5.164.2.91 DWT_CTRL_CYCEVTENA_Msk

```
#define DWT_CTRL_CYCEVTENA_Msk (0x1UL << DWT_CTRL_CYCEVTENA_Pos)
```

DWT CTRL: CYCEVTENA Mask

5.164.2.92 DWT_CTRL_CYCEVTENA_Pos

```
#define DWT_CTRL_CYCEVTENA_Pos 22
```

DWT CTRL: CYCEVTENA Position

5.164.2.93 DWT_CTRL_CYCTAP_Msk

```
#define DWT_CTRL_CYCTAP_Msk (0x1UL << DWT_CTRL_CYCTAP_Pos)
```

DWT CTRL: CYCTAP Mask

5.164.2.94 DWT_CTRL_CYCTAP_Pos

```
#define DWT_CTRL_CYCTAP_Pos 9
```

DWT CTRL: CYCTAP Position

5.164.2.95 DWT_CTRL_EXCEVTENA_Msk

```
#define DWT_CTRL_EXCEVTENA_Msk (0x1UL << DWT_CTRL_EXCEVTENA_Pos)
```

DWT CTRL: EXCEVTENA Mask

5.164.2.96 DWT_CTRL_EXCEVTENA_Pos

```
#define DWT_CTRL_EXCEVTENA_Pos 18
```

DWT CTRL: EXCEVTENA Position

5.164.2.97 DWT_CTRL_EXCTRCENA_Msk

```
#define DWT_CTRL_EXCTRCENA_Msk (0x1UL << DWT_CTRL_EXCTRCENA_Pos)
```

DWT CTRL: EXCTRCENA Mask

5.164.2.98 DWT_CTRL_EXCTRCENA_Pos

```
#define DWT_CTRL_EXCTRCENA_Pos 16
```

DWT CTRL: EXCTRCENA Position

5.164.2.99 DWT_CTRL_FOLDEVTENA_Msk

```
#define DWT_CTRL_FOLDEVTENA_Msk (0x1UL << DWT_CTRL_FOLDEVTENA_Pos)
```

DWT CTRL: FOLDEVTENA Mask

5.164.2.100 DWT_CTRL_FOLDEVTENA_Pos

```
#define DWT_CTRL_FOLDEVTENA_Pos 21
```

DWT CTRL: FOLDEVTENA Position

5.164.2.101 DWT_CTRL_LSUEVTENA_Msk

```
#define DWT_CTRL_LSUEVTENA_Msk (0x1UL << DWT_CTRL_LSUEVTENA_Pos)
```

DWT CTRL: LSUEVTENA Mask

5.164.2.102 DWT_CTRL_LSUEVTENA_Pos

```
#define DWT_CTRL_LSUEVTENA_Pos 20
```

DWT CTRL: LSUEVTENA Position

5.164.2.103 DWT_CTRL_NOCCNT_Msk

```
#define DWT_CTRL_NOCCNT_Msk (0x1UL << DWT_CTRL_NOCCNT_Pos)
```

DWT CTRL: NOCCNT Mask

5.164.2.104 DWT_CTRL_NOCCNT_Pos

```
#define DWT_CTRL_NOCCNT_Pos 25
```

DWT CTRL: NOCCNT Position

5.164.2.105 DWT_CTRL_NOEXTTRIG_Msk

```
#define DWT_CTRL_NOEXTTRIG_Msk (0x1UL << DWT_CTRL_NOEXTTRIG_Pos)
```

DWT CTRL: NOEXTTRIG Mask

5.164.2.106 DWT_CTRL_NOEXTTRIG_Pos

```
#define DWT_CTRL_NOEXTTRIG_Pos 26
```

DWT CTRL: NOEXTTRIG Position

5.164.2.107 DWT_CTRL_NOPRFCNT_Msk

```
#define DWT_CTRL_NOPRFCNT_Msk (0x1UL << DWT_CTRL_NOPRFCNT_Pos)
```

DWT CTRL: NOPRFCNT Mask

5.164.2.108 DWT_CTRL_NOPRFCNT_Pos

```
#define DWT_CTRL_NOPRFCNT_Pos 24
```

DWT CTRL: NOPRFCNT Position

5.164.2.109 DWT_CTRL_NOTRCPKT_Msk

```
#define DWT_CTRL_NOTRCPKT_Msk (0x1UL << DWT_CTRL_NOTRCPKT_Pos)
```

DWT CTRL: NOTRCPKT Mask

5.164.2.110 DWT_CTRL_NOTRCPKT_Pos

```
#define DWT_CTRL_NOTRCPKT_Pos 27
```

DWT CTRL: NOTRCPKT Position

5.164.2.111 DWT_CTRL_NUMCOMP_Msk

```
#define DWT_CTRL_NUMCOMP_Msk (0xFUL << DWT_CTRL_NUMCOMP_Pos)
```

DWT CTRL: NUMCOMP Mask

5.164.2.112 DWT_CTRL_NUMCOMP_Pos

```
#define DWT_CTRL_NUMCOMP_Pos 28
```

DWT CTRL: NUMCOMP Position

5.164.2.113 DWT_CTRL_PCSAMPLENA_Msk

```
#define DWT_CTRL_PCSAMPLENA_Msk (0x1UL << DWT_CTRL_PCSAMPLENA_Pos)
```

DWT CTRL: PCSAMPLENA Mask

5.164.2.114 DWT_CTRL_PCSAMPLENA_Pos

```
#define DWT_CTRL_PCSAMPLENA_Pos 12
```

DWT CTRL: PCSAMPLENA Position

5.164.2.115 DWT_CTRL_POSTINIT_Msk

```
#define DWT_CTRL_POSTINIT_Msk (0xFUL << DWT_CTRL_POSTINIT_Pos)
```

DWT CTRL: POSTINIT Mask

5.164.2.116 DWT_CTRL_POSTINIT_Pos

```
#define DWT_CTRL_POSTINIT_Pos 5
```

DWT CTRL: POSTINIT Position

5.164.2.117 DWT_CTRL_POSTPRESET_Msk

```
#define DWT_CTRL_POSTPRESET_Msk (0xFUL << DWT_CTRL_POSTPRESET_Pos)
```

DWT CTRL: POSTPRESET Mask

5.164.2.118 DWT_CTRL_POSTPRESET_Pos

```
#define DWT_CTRL_POSTPRESET_Pos 1
```

DWT CTRL: POSTPRESET Position

5.164.2.119 DWT_CTRL_SLEEPEVTENA_Msk

```
#define DWT_CTRL_SLEEPEVTENA_Msk (0x1UL << DWT_CTRL_SLEEPEVTENA_Pos)
```

DWT CTRL: SLEEPEVTENA Mask

5.164.2.120 DWT_CTRL_SLEEPEVTENA_Pos

```
#define DWT_CTRL_SLEEPEVTENA_Pos 19
```

DWT CTRL: SLEEPEVTENA Position

5.164.2.121 DWT_CTRL_SYNCTAP_Msk

```
#define DWT_CTRL_SYNCTAP_Msk (0x3UL << DWT_CTRL_SYNCTAP_Pos)
```

DWT CTRL: SYNCTAP Mask

5.164.2.122 DWT_CTRL_SYNCTAP_Pos

```
#define DWT_CTRL_SYNCTAP_Pos 10
```

DWT CTRL: SYNCTAP Position

5.164.2.123 DWT_EXCCNT_EXCCNT_Msk

```
#define DWT_EXCCNT_EXCCNT_Msk (0xFFUL /*<< DWT_EXCCNT_EXCCNT_Pos*/)
```

DWT EXCCNT: EXCCNT Mask

5.164.2.124 DWT_EXCCNT_EXCCNT_Pos

```
#define DWT_EXCCNT_EXCCNT_Pos 0
```

DWT EXCCNT: EXCCNT Position

5.164.2.125 DWT_FOLDCNT_FOLDCNT_Msk

```
#define DWT_FOLDCNT_FOLDCNT_Msk (0xFFUL /*<< DWT_FOLDCNT_FOLDCNT_Pos*/)
```

DWT FOLDCNT: FOLDCNT Mask

5.164.2.126 DWT_FOLDCNT_FOLDCNT_Pos

```
#define DWT_FOLDCNT_FOLDCNT_Pos 0
```

DWT FOLDCNT: FOLDCNT Position

5.164.2.127 DWT_FUNCTION_CYCMATCH_Msk

```
#define DWT_FUNCTION_CYCMATCH_Msk (0x1UL << DWT_FUNCTION_CYCMATCH_Pos)
```

DWT FUNCTION: CYCMATCH Mask

5.164.2.128 DWT_FUNCTION_CYCMATCH_Pos

```
#define DWT_FUNCTION_CYCMATCH_Pos 7
```

DWT FUNCTION: CYCMATCH Position

5.164.2.129 DWT_FUNCTION_DATAVADDR0_Msk

```
#define DWT_FUNCTION_DATAVADDR0_Msk (0xFUL << DWT_FUNCTION_DATAVADDR0_Pos)
```

DWT FUNCTION: DATAVADDR0 Mask

5.164.2.130 DWT_FUNCTION_DATAVADDR0_Pos

```
#define DWT_FUNCTION_DATAVADDR0_Pos 12
```

DWT FUNCTION: DATAVADDR0 Position

5.164.2.131 DWT_FUNCTION_DATAVADDR1_Msk

```
#define DWT_FUNCTION_DATAVADDR1_Msk (0xFUL << DWT_FUNCTION_DATAVADDR1_Pos)
```

DWT FUNCTION: DATAVADDR1 Mask

5.164.2.132 DWT_FUNCTION_DATAVADDR1_Pos

```
#define DWT_FUNCTION_DATAVADDR1_Pos 16
```

DWT FUNCTION: DATAVADDR1 Position

5.164.2.133 DWT_FUNCTION_DATAVMATCH_Msk

```
#define DWT_FUNCTION_DATAVMATCH_Msk (0x1UL << DWT_FUNCTION_DATAVMATCH_Pos)
```

DWT FUNCTION: DATAVMATCH Mask

5.164.2.134 DWT_FUNCTION_DATAVMATCH_Pos

```
#define DWT_FUNCTION_DATAVMATCH_Pos 8
```

DWT FUNCTION: DATAVMATCH Position

5.164.2.135 DWT_FUNCTION_DATAVSIZE_Msk

```
#define DWT_FUNCTION_DATAVSIZE_Msk (0x3UL << DWT_FUNCTION_DATAVSIZE_Pos)
```

DWT FUNCTION: DATAVSIZE Mask

5.164.2.136 DWT_FUNCTION_DATAVSIZE_Pos

```
#define DWT_FUNCTION_DATAVSIZE_Pos 10
```

DWT FUNCTION: DATAVSIZE Position

5.164.2.137 DWT_FUNCTION_EMITRANGE_Msk

```
#define DWT_FUNCTION_EMITRANGE_Msk (0x1UL << DWT_FUNCTION_EMITRANGE_Pos)
```

DWT FUNCTION: EMITRANGE Mask

5.164.2.138 DWT_FUNCTION_EMITRANGE_Pos

```
#define DWT_FUNCTION_EMITRANGE_Pos 5
```

DWT FUNCTION: EMITRANGE Position

5.164.2.139 DWT_FUNCTION_FUNCTION_Msk

```
#define DWT_FUNCTION_FUNCTION_Msk (0xFUL /*<< DWT_FUNCTION_FUNCTION_Pos*/)
```

DWT FUNCTION: FUNCTION Mask

5.164.2.140 DWT_FUNCTION_FUNCTION_Pos

```
#define DWT_FUNCTION_FUNCTION_Pos 0
```

DWT FUNCTION: FUNCTION Position

5.164.2.141 DWT_FUNCTION_LNK1ENA_Msk

```
#define DWT_FUNCTION_LNK1ENA_Msk (0x1UL << DWT_FUNCTION_LNK1ENA_Pos)
```

DWT FUNCTION: LNK1ENA Mask

5.164.2.142 DWT_FUNCTION_LNK1ENA_Pos

```
#define DWT_FUNCTION_LNK1ENA_Pos 9
```

DWT FUNCTION: LNK1ENA Position

5.164.2.143 DWT_FUNCTION_MATCHED_Msk

```
#define DWT_FUNCTION_MATCHED_Msk (0x1UL << DWT_FUNCTION_MATCHED_Pos)
```

DWT FUNCTION: MATCHED Mask

5.164.2.144 DWT_FUNCTION_MATCHED_Pos

```
#define DWT_FUNCTION_MATCHED_Pos 24
```

DWT FUNCTION: MATCHED Position

5.164.2.145 DWT_LSUCNT_LSUCNT_Msk

```
#define DWT_LSUCNT_LSUCNT_Msk (0xFFUL /*<< DWT_LSUCNT_LSUCNT_Pos*/)
```

DWT LSUCNT: LSUCNT Mask

5.164.2.146 DWT_LSUCNT_LSUCNT_Pos

```
#define DWT_LSUCNT_LSUCNT_Pos 0
```

DWT LSUCNT: LSUCNT Position

5.164.2.147 DWT_MASK_MASK_Msk

```
#define DWT_MASK_MASK_Msk (0x1FUL /*<< DWT_MASK_MASK_Pos*/)
```

DWT MASK: MASK Mask

5.164.2.148 DWT_MASK_MASK_Pos

```
#define DWT_MASK_MASK_Pos 0
```

DWT MASK: MASK Position

5.164.2.149 DWT_SLEEPcnt_SLEEPcnt_Msk

```
#define DWT_SLEEPcnt_SLEEPcnt_Msk (0xFFUL /*<< DWT_SLEEPcnt_SLEEPcnt_Pos*/)
```

DWT SLEEPcnt: SLEEPcnt Mask

5.164.2.150 DWT_SLEEPcnt_SLEEPcnt_Pos

```
#define DWT_SLEEPcnt_SLEEPcnt_Pos 0
```

DWT SLEEPcnt: SLEEPcnt Position

5.164.2.151 IPSR_ISR_Msk

```
#define IPSR_ISR_Msk (0x1FFUL /*<< IPSR_ISR_Pos*/)
```

IPSR: ISR Mask

5.164.2.152 IPSR_ISR_Pos

```
#define IPSR_ISR_Pos 0
```

IPSR: ISR Position

5.164.2.153 ITM

```
#define ITM ((ITM_Type *) ITM_BASE)
```

ITM configuration struct

5.164.2.154 ITM_BASE

```
#define ITM_BASE (0xE0000000UL)
```

ITM Base Address

5.164.2.155 ITM_IMCR_INTEGRATION_Msk

```
#define ITM_IMCR_INTEGRATION_Msk (1UL /*<< ITM_IMCR_INTEGRATION_Pos*/)
```

ITM IMCR: INTEGRATION Mask

5.164.2.156 ITM_IMCR_INTEGRATION_Pos

```
#define ITM_IMCR_INTEGRATION_Pos 0
```

ITM IMCR: INTEGRATION Position

5.164.2.157 ITMIRR_ATREADYM_Msk

```
#define ITMIRR_ATREADYM_Msk (1UL /*<< ITMIRR_ATREADYM_Pos*/)
```

ITM IRR: ATREADYM Mask

5.164.2.158 ITMIRR_ATREADYM_Pos

```
#define ITMIRR_ATREADYM_Pos 0
```

ITM IRR: ATREADYM Position

5.164.2.159 ITMIWR_ATVALIDDM_Msk

```
#define ITMIWR_ATVALIDDM_Msk (1UL /*<< ITMIWR_ATVALIDDM_Pos*/)
```

ITM IWR: ATVALIDDM Mask

5.164.2.160 ITM_IWR_ATVALIDM_Pos

```
#define ITM_IWR_ATVALIDM_Pos 0
```

ITM IWR: ATVALIDM Position

5.164.2.161 ITM_LSR_Access_Msk

```
#define ITM_LSR_Access_Msk (1UL << ITM_LSR_Access_Pos)
```

ITM LSR: Access Mask

5.164.2.162 ITM_LSR_Access_Pos

```
#define ITM_LSR_Access_Pos 1
```

ITM LSR: Access Position

5.164.2.163 ITM_LSR_ByteAcc_Msk

```
#define ITM_LSR_ByteAcc_Msk (1UL << ITM_LSR_ByteAcc_Pos)
```

ITM LSR: ByteAcc Mask

5.164.2.164 ITM_LSR_ByteAcc_Pos

```
#define ITM_LSR_ByteAcc_Pos 2
```

ITM LSR: ByteAcc Position

5.164.2.165 ITM_LSR_Present_Msk

```
#define ITM_LSR_Present_Msk (1UL /*<< ITM_LSR_Present_Pos*/)
```

ITM LSR: Present Mask

5.164.2.166 ITM_LSR_Present_Pos

```
#define ITM_LSR_Present_Pos 0
```

ITM LSR: Present Position

5.164.2.167 ITM_RXBUFFER_EMPTY

```
#define ITM_RXBUFFER_EMPTY 0x5AA55AA5
```

Value identifying [ITM_RxBuffer](#) is ready for next character.

5.164.2.168 **ITM_TCR_BUSY_Msk**

```
#define ITM_TCR_BUSY_Msk (1UL << ITM_TCR_BUSY_Pos)
```

ITM TCR: BUSY Mask

5.164.2.169 **ITM_TCR_BUSY_Pos**

```
#define ITM_TCR_BUSY_Pos 23
```

ITM TCR: BUSY Position

5.164.2.170 **ITM_TCR_DWTENA_Msk**

```
#define ITM_TCR_DWTENA_Msk (1UL << ITM_TCR_DWTENA_Pos)
```

ITM TCR: DWTENA Mask

5.164.2.171 **ITM_TCR_DWTENA_Pos**

```
#define ITM_TCR_DWTENA_Pos 3
```

ITM TCR: DWTENA Position

5.164.2.172 **ITM_TCR_GTSFREQ_Msk**

```
#define ITM_TCR_GTSFREQ_Msk (3UL << ITM_TCR_GTSFREQ_Pos)
```

ITM TCR: Global timestamp frequency Mask

5.164.2.173 **ITM_TCR_GTSFREQ_Pos**

```
#define ITM_TCR_GTSFREQ_Pos 10
```

ITM TCR: Global timestamp frequency Position

5.164.2.174 **ITM_TCR_ITMENA_Msk**

```
#define ITM_TCR_ITMENA_Msk (1UL /*<< ITM_TCR_ITMENA_Pos*/)
```

ITM TCR: ITM Enable bit Mask

5.164.2.175 **ITM_TCR_ITMENA_Pos**

```
#define ITM_TCR_ITMENA_Pos 0
```

ITM TCR: ITM Enable bit Position

5.164.2.176 ITM_TCR_SWOENA_Msk

```
#define ITM_TCR_SWOENA_Msk (1UL << ITM_TCR_SWOENA_Pos)
```

ITM TCR: SWOENA Mask

5.164.2.177 ITM_TCR_SWOENA_Pos

```
#define ITM_TCR_SWOENA_Pos 4
```

ITM TCR: SWOENA Position

5.164.2.178 ITM_TCR_SYNCENA_Msk

```
#define ITM_TCR_SYNCENA_Msk (1UL << ITM_TCR_SYNCENA_Pos)
```

ITM TCR: SYNCENA Mask

5.164.2.179 ITM_TCR_SYNCENA_Pos

```
#define ITM_TCR_SYNCENA_Pos 2
```

ITM TCR: SYNCENA Position

5.164.2.180 ITM_TCR_TraceBusID_Msk

```
#define ITM_TCR_TraceBusID_Msk (0x7FUL << ITM_TCR_TraceBusID_Pos)
```

ITM TCR: ATBID Mask

5.164.2.181 ITM_TCR_TraceBusID_Pos

```
#define ITM_TCR_TraceBusID_Pos 16
```

ITM TCR: ATBID Position

5.164.2.182 ITM_TCR_TSENA_Msk

```
#define ITM_TCR_TSENA_Msk (1UL << ITM_TCR_TSENA_Pos)
```

ITM TCR: TSENA Mask

5.164.2.183 ITM_TCR_TSENA_Pos

```
#define ITM_TCR_TSENA_Pos 1
```

ITM TCR: TSENA Position

5.164.2.184 ITM_TCR_TSPrescale_Msk

```
#define ITM_TCR_TSPrescale_Msk (3UL << ITM_TCR_TSPrescale_Pos)
```

ITM TCR: TSPrescale Mask

5.164.2.185 ITM_TCR_TSPrescale_Pos

```
#define ITM_TCR_TSPrescale_Pos 8
```

ITM TCR: TSPrescale Position

5.164.2.186 ITM_TPR_PRIVMASK_Msk

```
#define ITM_TPR_PRIVMASK_Msk (0xFUL /*<< ITM_TPR_PRIVMASK_Pos*/)
```

ITM TPR: PRIVMASK Mask

5.164.2.187 ITM_TPR_PRIVMASK_Pos

```
#define ITM_TPR_PRIVMASK_Pos 0
```

ITM TPR: PRIVMASK Position

5.164.2.188 NVIC

```
#define NVIC ((NVIC_Type *) NVIC_BASE )
```

NVIC configuration struct

5.164.2.189 NVIC_BASE

```
#define NVIC_BASE (SCS_BASE + 0x0100UL)
```

NVIC Base Address

5.164.2.190 NVIC_STIR_INTID_Msk

```
#define NVIC_STIR_INTID_Msk (0x1FFUL /*<< NVIC_STIR_INTID_Pos*/)
```

STIR: INTLINESNUM Mask

5.164.2.191 NVIC_STIR_INTID_Pos

```
#define NVIC_STIR_INTID_Pos 0
```

STIR: INTLINESNUM Position

5.164.2.192 SCB

```
#define SCB ((SCB_Type *) SCB_BASE)
```

SCB configuration struct

5.164.2.193 SCB_AIRCR_ENDIANESS_Msk

```
#define SCB_AIRCR_ENDIANESS_Msk (1UL << SCB_AIRCR_ENDIANESS_Pos)
```

SCB AIRCR: ENDIANESS Mask

5.164.2.194 SCB_AIRCR_ENDIANESS_Pos

```
#define SCB_AIRCR_ENDIANESS_Pos 15
```

SCB AIRCR: ENDIANESS Position

5.164.2.195 SCB_AIRCR_PRIGROUP_Msk

```
#define SCB_AIRCR_PRIGROUP_Msk (7UL << SCB_AIRCR_PRIGROUP_Pos)
```

SCB AIRCR: PRIGROUP Mask

5.164.2.196 SCB_AIRCR_PRIGROUP_Pos

```
#define SCB_AIRCR_PRIGROUP_Pos 8
```

SCB AIRCR: PRIGROUP Position

5.164.2.197 SCB_AIRCR_SYSRESETREQ_Msk

```
#define SCB_AIRCR_SYSRESETREQ_Msk (1UL << SCB_AIRCR_SYSRESETREQ_Pos)
```

SCB AIRCR: SYSRESETREQ Mask

5.164.2.198 SCB_AIRCR_SYSRESETREQ_Pos

```
#define SCB_AIRCR_SYSRESETREQ_Pos 2
```

SCB AIRCR: SYSRESETREQ Position

5.164.2.199 SCB_AIRCR_VECTCLRACTIVE_Msk

```
#define SCB_AIRCR_VECTCLRACTIVE_Msk (1UL << SCB_AIRCR_VECTCLRACTIVE_Pos)
```

SCB AIRCR: VECTCLRACTIVE Mask

5.164.2.200 SCB_AIRCR_VECTCLRACTIVE_Pos

```
#define SCB_AIRCR_VECTCLRACTIVE_Pos 1
```

SCB AIRCR: VECTCLRACTIVE Position

5.164.2.201 SCB_AIRCR_VECTKEY_Msk

```
#define SCB_AIRCR_VECTKEY_Msk (0xFFFFUL << SCB_AIRCR_VECTKEY_Pos)
```

SCB AIRCR: VECTKEY Mask

5.164.2.202 SCB_AIRCR_VECTKEY_Pos

```
#define SCB_AIRCR_VECTKEY_Pos 16
```

SCB AIRCR: VECTKEY Position

5.164.2.203 SCB_AIRCR_VECTKEYSTAT_Msk

```
#define SCB_AIRCR_VECTKEYSTAT_Msk (0xFFFFUL << SCB_AIRCR_VECTKEYSTAT_Pos)
```

SCB AIRCR: VECTKEYSTAT Mask

5.164.2.204 SCB_AIRCR_VECTKEYSTAT_Pos

```
#define SCB_AIRCR_VECTKEYSTAT_Pos 16
```

SCB AIRCR: VECTKEYSTAT Position

5.164.2.205 SCB_AIRCR_VECTRESET_Msk

```
#define SCB_AIRCR_VECTRESET_Msk (1UL /*<< SCB_AIRCR_VECTRESET_Pos*/)
```

SCB AIRCR: VECTRESET Mask

5.164.2.206 SCB_AIRCR_VECTRESET_Pos

```
#define SCB_AIRCR_VECTRESET_Pos 0
```

SCB AIRCR: VECTRESET Position

5.164.2.207 SCB_BASE

```
#define SCB_BASE (SCS_BASE + 0x0D00UL)
```

System Control Block Base Address

5.164.2.208 SCB_CCR_BFHFNIGN_Msk

```
#define SCB_CCR_BFHFNIGN_Msk (1UL << SCB_CCR_BFHFNIGN_Pos)
```

SCB CCR: BFHFNMIGN Mask

5.164.2.209 SCB_CCR_BFHFNIGN_Pos

```
#define SCB_CCR_BFHFNIGN_Pos 8
```

SCB CCR: BFHFNMIGN Position

5.164.2.210 SCB_CCR_DIV_0_TRP_Msk

```
#define SCB_CCR_DIV_0_TRP_Msk (1UL << SCB_CCR_DIV_0_TRP_Pos)
```

SCB CCR: DIV_0_TRP Mask

5.164.2.211 SCB_CCR_DIV_0_TRP_Pos

```
#define SCB_CCR_DIV_0_TRP_Pos 4
```

SCB CCR: DIV_0_TRP Position

5.164.2.212 SCB_CCR_NONBASETHRDENA_Msk

```
#define SCB_CCR_NONBASETHRDENA_Msk (1UL /*<< SCB_CCR_NONBASETHRDENA_Pos*/)
```

SCB CCR: NONBASETHRDENA Mask

5.164.2.213 SCB_CCR_NONBASETHRDENA_Pos

```
#define SCB_CCR_NONBASETHRDENA_Pos 0
```

SCB CCR: NONBASETHRDENA Position

5.164.2.214 SCB_CCR_STKALIGN_Msk

```
#define SCB_CCR_STKALIGN_Msk (1UL << SCB_CCR_STKALIGN_Pos)
```

SCB CCR: STKALIGN Mask

5.164.2.215 SCB_CCR_STKALIGN_Pos

```
#define SCB_CCR_STKALIGN_Pos 9
```

SCB CCR: STKALIGN Position

5.164.2.216 SCB_CCR_UNALIGN_TRP_Msk

```
#define SCB_CCR_UNALIGN_TRP_Msk (1UL << SCB_CCR_UNALIGN_TRP_Pos)
```

SCB CCR: UNALIGN_TRP Mask

5.164.2.217 SCB_CCR_UNALIGN_TRP_Pos

```
#define SCB_CCR_UNALIGN_TRP_Pos 3
```

SCB CCR: UNALIGN_TRP Position

5.164.2.218 SCB_CCR_USERSETMPEND_Msk

```
#define SCB_CCR_USERSETMPEND_Msk (1UL << SCB_CCR_USERSETMPEND_Pos)
```

SCB CCR: USERSETMPEND Mask

5.164.2.219 SCB_CCR_USERSETMPEND_Pos

```
#define SCB_CCR_USERSETMPEND_Pos 1
```

SCB CCR: USERSETMPEND Position

5.164.2.220 SCB_CFSR_BUSFAULTSR_Msk

```
#define SCB_CFSR_BUSFAULTSR_Msk (0xFFUL << SCB_CFSR_BUSFAULTSR_Pos)
```

SCB CFSR: Bus Fault Status Register Mask

5.164.2.221 SCB_CFSR_BUSFAULTSR_Pos

```
#define SCB_CFSR_BUSFAULTSR_Pos 8
```

SCB CFSR: Bus Fault Status Register Position

5.164.2.222 SCB_CFSR_MEMFAULTSR_Msk

```
#define SCB_CFSR_MEMFAULTSR_Msk (0xFFUL /*<< SCB_CFSR_MEMFAULTSR_Pos*/)
```

SCB CFSR: Memory Manage Fault Status Register Mask

5.164.2.223 SCB_CFSR_MEMFAULTSR_Pos

```
#define SCB_CFSR_MEMFAULTSR_Pos 0
```

SCB CFSR: Memory Manage Fault Status Register Position

5.164.2.224 SCB_CFSR_USGFAULTSR_Msk

```
#define SCB_CFSR_USGFAULTSR_Msk (0xFFFFUL << SCB_CFSR_USGFAULTSR_Pos)
```

SCB CFSR: Usage Fault Status Register Mask

5.164.2.225 SCB_CFSR_USGFAULTSR_Pos

```
#define SCB_CFSR_USGFAULTSR_Pos 16
```

SCB CFSR: Usage Fault Status Register Position

5.164.2.226 SCB_CPUID_ARCHITECTURE_Msk

```
#define SCB_CPUID_ARCHITECTURE_Msk (0xFUL << SCB_CPUID_ARCHITECTURE_Pos)
```

SCB CPUID: ARCHITECTURE Mask

5.164.2.227 SCB_CPUID_ARCHITECTURE_Pos

```
#define SCB_CPUID_ARCHITECTURE_Pos 16
```

SCB CPUID: ARCHITECTURE Position

5.164.2.228 SCB_CPUID_IMPLEMENTER_Msk

```
#define SCB_CPUID_IMPLEMENTER_Msk (0xFFUL << SCB_CPUID_IMPLEMENTER_Pos)
```

SCB CPUID: IMPLEMENTER Mask

5.164.2.229 SCB_CPUID_IMPLEMENTER_Pos

```
#define SCB_CPUID_IMPLEMENTER_Pos 24
```

SCB CPUID: IMPLEMENTER Position

5.164.2.230 SCB_CPUID_PARTNO_Msk

```
#define SCB_CPUID_PARTNO_Msk (0xFFFFUL << SCB_CPUID_IMPLEMENTER_Pos)
```

SCB CPUID: PARTNO Mask

5.164.2.231 SCB_CPUID_PARTNO_Pos

```
#define SCB_CPUID_PARTNO_Pos 4
```

SCB CPUID: PARTNO Position

5.164.2.232 SCB_CPUID_REVISION_Msk

```
#define SCB_CPUID_REVISION_Msk (0xFUL /*<< SCB_CPUID_REVISION_Pos*/)
```

SCB CPUID: REVISION Mask

5.164.2.233 SCB_CPUID_REVISION_Pos

```
#define SCB_CPUID_REVISION_Pos 0
```

SCB CPUID: REVISION Position

5.164.2.234 SCB_CPUID_VARIANT_Msk

```
#define SCB_CPUID_VARIANT_Msk (0xFUL << SCB_CPUID_VARIANT_Pos)
```

SCB CPUID: VARIANT Mask

5.164.2.235 SCB_CPUID_VARIANT_Pos

```
#define SCB_CPUID_VARIANT_Pos 20
```

SCB CPUID: VARIANT Position

5.164.2.236 SCB_DFSR_BKPT_Msk

```
#define SCB_DFSR_BKPT_Msk (1UL << SCB_DFSR_BKPT_Pos)
```

SCB DFSR: BKPT Mask

5.164.2.237 SCB_DFSR_BKPT_Pos

```
#define SCB_DFSR_BKPT_Pos 1
```

SCB DFSR: BKPT Position

5.164.2.238 SCB_DFSR_DWTTRAP_Msk

```
#define SCB_DFSR_DWTTRAP_Msk (1UL << SCB_DFSR_DWTTRAP_Pos)
```

SCB DFSR: DWTTRAP Mask

5.164.2.239 SCB_DFSR_DWTTRAP_Pos

```
#define SCB_DFSR_DWTTRAP_Pos 2
```

SCB DFSR: DWTTRAP Position

5.164.2.240 SCB_DFSR_EXTERNAL_Msk

```
#define SCB_DFSR_EXTERNAL_Msk (1UL << SCB_DFSR_EXTERNAL_Pos)
```

SCB DFSR: EXTERNAL Mask

5.164.2.241 SCB_DFSR_EXTERNAL_Pos

```
#define SCB_DFSR_EXTERNAL_Pos 4
```

SCB DFSR: EXTERNAL Position

5.164.2.242 SCB_DFSR_HALTED_Msk

```
#define SCB_DFSR_HALTED_Msk (1UL /*<< SCB_DFSR_HALTED_Pos*/)
```

SCB DFSR: HALTED Mask

5.164.2.243 SCB_DFSR_HALTED_Pos

```
#define SCB_DFSR_HALTED_Pos 0
```

SCB DFSR: HALTED Position

5.164.2.244 SCB_DFSR_VCATCH_Msk

```
#define SCB_DFSR_VCATCH_Msk (1UL << SCB_DFSR_VCATCH_Pos)
```

SCB DFSR: VCATCH Mask

5.164.2.245 SCB_DFSR_VCATCH_Pos

```
#define SCB_DFSR_VCATCH_Pos 3
```

SCB DFSR: VCATCH Position

5.164.2.246 SCB_HFSR_DEBUGEV_T_Msk

```
#define SCB_HFSR_DEBUGEV_T_Msk (1UL << SCB_HFSR_DEBUGEV_T_Pos)
```

SCB HFSR: DEBUGEV_T Mask

5.164.2.247 SCB_HFSR_DEBUGEV_T_Pos

```
#define SCB_HFSR_DEBUGEV_T_Pos 31
```

SCB HFSR: DEBUGEV_T Position

5.164.2.248 SCB_HFSR_FORCED_Msk

```
#define SCB_HFSR_FORCED_Msk (1UL << SCB_HFSR_FORCED_Pos)
```

SCB HFSR: FORCED Mask

5.164.2.249 SCB_HFSR_FORCED_Pos

```
#define SCB_HFSR_FORCED_Pos 30
```

SCB HFSR: FORCED Position

5.164.2.250 SCB_HFSR_VECTTBL_Msk

```
#define SCB_HFSR_VECTTBL_Msk (1UL << SCB_HFSR_VECTTBL_Pos)
```

SCB HFSR: VECTTBL Mask

5.164.2.251 SCB_HFSR_VECTTBL_Pos

```
#define SCB_HFSR_VECTTBL_Pos 1
```

SCB HFSR: VECTTBL Position

5.164.2.252 SCB_ICSR_ISRPENDING_Msk

```
#define SCB_ICSR_ISRPENDING_Msk (1UL << SCB_ICSR_ISRPENDING_Pos)
```

SCB ICSR: ISRPENDING Mask

5.164.2.253 SCB_ICSR_ISRPENDING_Pos

```
#define SCB_ICSR_ISRPENDING_Pos 22
```

SCB ICSR: ISRPENDING Position

5.164.2.254 SCB_ICSR_ISRPREEMPT_Msk

```
#define SCB_ICSR_ISRPREEMPT_Msk (1UL << SCB_ICSR_ISRPREEMPT_Pos)
```

SCB ICSR: ISRPREEMPT Mask

5.164.2.255 SCB_ICSR_ISRPREEMPT_Pos

```
#define SCB_ICSR_ISRPREEMPT_Pos 23
```

SCB ICSR: ISRPREEMPT Position

5.164.2.256 SCB_ICSR_NMIPENDSET_Msk

```
#define SCB_ICSR_NMIPENDSET_Msk (1UL << SCB_ICSR_NMIPENDSET_Pos)
```

SCB ICSR: NMIPENDSET Mask

5.164.2.257 SCB_ICSR_NMIPENDSET_Pos

```
#define SCB_ICSR_NMIPENDSET_Pos 31
```

SCB ICSR: NMIPENDSET Position

5.164.2.258 SCB_ICSR_PENDSTCLR_Msk

```
#define SCB_ICSR_PENDSTCLR_Msk (1UL << SCB_ICSR_PENDSTCLR_Pos)
```

SCB ICSR: PENDSTCLR Mask

5.164.2.259 SCB_ICSR_PENDSTCLR_Pos

```
#define SCB_ICSR_PENDSTCLR_Pos 25
```

SCB ICSR: PENDSTCLR Position

5.164.2.260 SCB_ICSR_PENDSTSET_Msk

```
#define SCB_ICSR_PENDSTSET_Msk (1UL << SCB_ICSR_PENDSTSET_Pos)
```

SCB ICSR: PENDSTSET Mask

5.164.2.261 SCB_ICSR_PENDSTSET_Pos

```
#define SCB_ICSR_PENDSTSET_Pos 26
```

SCB ICSR: PENDSTSET Position

5.164.2.262 SCB_ICSR_PENDSVCLR_Msk

```
#define SCB_ICSR_PENDSVCLR_Msk (1UL << SCB_ICSR_PENDSVCLR_Pos)
```

SCB ICSR: PENDSVCLR Mask

5.164.2.263 SCB_ICSR_PENDSVCLR_Pos

```
#define SCB_ICSR_PENDSVCLR_Pos 27
```

SCB ICSR: PENDSVCLR Position

5.164.2.264 SCB_ICSR_PENDSVSET_Msk

```
#define SCB_ICSR_PENDSVSET_Msk (1UL << SCB_ICSR_PENDSVSET_Pos)
```

SCB ICSR: PENDSVSET Mask

5.164.2.265 SCB_ICSR_PENDSVSET_Pos

```
#define SCB_ICSR_PENDSVSET_Pos 28
```

SCB ICSR: PENDSVSET Position

5.164.2.266 SCB_ICSR_RETTOBASE_Msk

```
#define SCB_ICSR_RETTOBASE_Msk (1UL << SCB_ICSR_RETTOBASE_Pos)
```

SCB ICSR: RETTOBASE Mask

5.164.2.267 SCB_ICSR_RETTOBASE_Pos

```
#define SCB_ICSR_RETTOBASE_Pos 11
```

SCB ICSR: RETTOBASE Position

5.164.2.268 SCB_ICSR_VECTACTIVE_Msk

```
#define SCB_ICSR_VECTACTIVE_Msk (0x1FFUL /*<< SCB_ICSR_VECTACTIVE_Pos*/)
```

SCB ICSR: VECTACTIVE Mask

5.164.2.269 SCB_ICSR_VECTACTIVE_Pos

```
#define SCB_ICSR_VECTACTIVE_Pos 0
```

SCB ICSR: VECTACTIVE Position

5.164.2.270 SCB_ICSR_VECTPENDING_Msk

```
#define SCB_ICSR_VECTPENDING_Msk (0x1FFUL << SCB_ICSR_VECTPENDING_Pos)
```

SCB ICSR: VECTPENDING Mask

5.164.2.271 SCB_ICSR_VECTPENDING_Pos

```
#define SCB_ICSR_VECTPENDING_Pos 12
```

SCB ICSR: VECTPENDING Position

5.164.2.272 SCB_SCR_SEVONPEND_Msk

```
#define SCB_SCR_SEVONPEND_Msk (1UL << SCB_SCR_SEVONPEND_Pos)
```

SCB SCR: SEVONPEND Mask

5.164.2.273 SCB_SCR_SEVONPEND_Pos

```
#define SCB_SCR_SEVONPEND_Pos 4
```

SCB SCR: SEVONPEND Position

5.164.2.274 SCB_SCR_SLEEPDEEP_Msk

```
#define SCB_SCR_SLEEPDEEP_Msk (1UL << SCB_SCR_SLEEPDEEP_Pos)
```

SCB SCR: SLEEPDEEP Mask

5.164.2.275 SCB_SCR_SLEEPDEEP_Pos

```
#define SCB_SCR_SLEEPDEEP_Pos 2
```

SCB SCR: SLEEPDEEP Position

5.164.2.276 SCB_SCR_SLEEPONEXIT_Msk

```
#define SCB_SCR_SLEEPONEXIT_Msk (1UL << SCB_SCR_SLEEPONEXIT_Pos)
```

SCB SCR: SLEEPONEXIT Mask

5.164.2.277 SCB_SCR_SLEEPONEXIT_Pos

```
#define SCB_SCR_SLEEPONEXIT_Pos 1
```

SCB SCR: SLEEPONEXIT Position

5.164.2.278 SCB_SHCSR_BUSFAULTACT_Msk

```
#define SCB_SHCSR_BUSFAULTACT_Msk (1UL << SCB_SHCSR_BUSFAULTACT_Pos)
```

SCB SHCSR: BUSFAULTACT Mask

5.164.2.279 SCB_SHCSR_BUSFAULTACT_Pos

```
#define SCB_SHCSR_BUSFAULTACT_Pos 1
```

SCB SHCSR: BUSFAULTACT Position

5.164.2.280 SCB_SHCSR_BUSFAULTENA_Msk

```
#define SCB_SHCSR_BUSFAULTENA_Msk (1UL << SCB_SHCSR_BUSFAULTENA_Pos)
```

SCB SHCSR: BUSFAULTENA Mask

5.164.2.281 SCB_SHCSR_BUSFAULTENA_Pos

```
#define SCB_SHCSR_BUSFAULTENA_Pos 17
```

SCB SHCSR: BUSFAULTENA Position

5.164.2.282 SCB_SHCSR_BUSFAULTPENDED_Msk

```
#define SCB_SHCSR_BUSFAULTPENDED_Msk (1UL << SCB_SHCSR_BUSFAULTPENDED_Pos)
```

SCB SHCSR: BUSFAULTPENDED Mask

5.164.2.283 SCB_SHCSR_BUSFAULTPENDED_Pos

```
#define SCB_SHCSR_BUSFAULTPENDED_Pos 14
```

SCB SHCSR: BUSFAULTPENDED Position

5.164.2.284 SCB_SHCSR_MEMFAULTACT_Msk

```
#define SCB_SHCSR_MEMFAULTACT_Msk (1UL /*<< SCB_SHCSR_MEMFAULTACT_Pos*/)
```

SCB SHCSR: MEMFAULTACT Mask

5.164.2.285 SCB_SHCSR_MEMFAULTACT_Pos

```
#define SCB_SHCSR_MEMFAULTACT_Pos 0
```

SCB SHCSR: MEMFAULTACT Position

5.164.2.286 SCB_SHCSR_MEMFAULTENA_Msk

```
#define SCB_SHCSR_MEMFAULTENA_Msk (1UL << SCB_SHCSR_MEMFAULTENA_Pos)
```

SCB SHCSR: MEMFAULTENA Mask

5.164.2.287 SCB_SHCSR_MEMFAULTENA_Pos

```
#define SCB_SHCSR_MEMFAULTENA_Pos 16
```

SCB SHCSR: MEMFAULTENA Position

5.164.2.288 SCB_SHCSR_MEMFAULTPENDED_Msk

```
#define SCB_SHCSR_MEMFAULTPENDED_Msk (1UL << SCB_SHCSR_MEMFAULTPENDED_Pos)
```

SCB SHCSR: MEMFAULTPENDED Mask

5.164.2.289 SCB_SHCSR_MEMFAULTPENDED_Pos

```
#define SCB_SHCSR_MEMFAULTPENDED_Pos 13
```

SCB SHCSR: MEMFAULTPENDED Position

5.164.2.290 SCB_SHCSR_MONITORACT_Msk

```
#define SCB_SHCSR_MONITORACT_Msk (1UL << SCB_SHCSR_MONITORACT_Pos)
```

SCB SHCSR: MONITORACT Mask

5.164.2.291 SCB_SHCSR_MONITORACT_Pos

```
#define SCB_SHCSR_MONITORACT_Pos 8
```

SCB SHCSR: MONITORACT Position

5.164.2.292 SCB_SHCSR_PENDSVACT_Msk

```
#define SCB_SHCSR_PENDSVACT_Msk (1UL << SCB_SHCSR_PENDSVACT_Pos)
```

SCB SHCSR: PENDSVACT Mask

5.164.2.293 SCB_SHCSR_PENDSVACT_Pos

```
#define SCB_SHCSR_PENDSVACT_Pos 10
```

SCB SHCSR: PENDSVACT Position

5.164.2.294 SCB_SHCSR_SVCALLACT_Msk

```
#define SCB_SHCSR_SVCALLACT_Msk (1UL << SCB_SHCSR_SVCALLACT_Pos)
```

SCB SHCSR: SVCALLACT Mask

5.164.2.295 SCB_SHCSR_SVCALLACT_Pos

```
#define SCB_SHCSR_SVCALLACT_Pos 7
```

SCB SHCSR: SVCALLACT Position

5.164.2.296 SCB_SHCSR_SVCALLPENDED_Msk

```
#define SCB_SHCSR_SVCALLPENDED_Msk (1UL << SCB_SHCSR_SVCALLPENDED_Pos)
```

SCB SHCSR: SVCALLPENDED Mask

5.164.2.297 SCB_SHCSR_SVCALLPENDED_Pos

```
#define SCB_SHCSR_SVCALLPENDED_Pos 15
```

SCB SHCSR: SVCALLPENDED Position

5.164.2.298 SCB_SHCSR_SYSTICKACT_Msk

```
#define SCB_SHCSR_SYSTICKACT_Msk (1UL << SCB_SHCSR_SYSTICKACT_Pos)
```

SCB SHCSR: SYSTICKACT Mask

5.164.2.299 SCB_SHCSR_SYSTICKACT_Pos

```
#define SCB_SHCSR_SYSTICKACT_Pos 11
```

SCB SHCSR: SYSTICKACT Position

5.164.2.300 SCB_SHCSR_USGFAULTACT_Msk

```
#define SCB_SHCSR_USGFAULTACT_Msk (1UL << SCB_SHCSR_USGFAULTACT_Pos)
```

SCB SHCSR: USGFAULTACT Mask

5.164.2.301 SCB_SHCSR_USGFAULTACT_Pos

```
#define SCB_SHCSR_USGFAULTACT_Pos 3
```

SCB SHCSR: USGFAULTACT Position

5.164.2.302 SCB_SHCSR_USGFAULTENA_Msk

```
#define SCB_SHCSR_USGFAULTENA_Msk (1UL << SCB_SHCSR_USGFAULTENA_Pos)
```

SCB SHCSR: USGFAULTENA Mask

5.164.2.303 SCB_SHCSR_USGFAULTENA_Pos

```
#define SCB_SHCSR_USGFAULTENA_Pos 18
```

SCB SHCSR: USGFAULTENA Position

5.164.2.304 SCB_SHCSR_USGFAULTPENDED_Msk

```
#define SCB_SHCSR_USGFAULTPENDED_Msk (1UL << SCB_SHCSR_USGFAULTPENDED_Pos)
```

SCB SHCSR: USGFAULTPENDED Mask

5.164.2.305 SCB_SHCSR_USGFAULTPENDED_Pos

```
#define SCB_SHCSR_USGFAULTPENDED_Pos 12
```

SCB SHCSR: USGFAULTPENDED Position

5.164.2.306 SCB_VTOR_TBLOFF_Msk

```
#define SCB_VTOR_TBLOFF_Msk (0x1FFFFFFUL << SCB_VTOR_TBLOFF_Pos)
```

SCB VTOR: TBLOFF Mask

5.164.2.307 SCB_VTOR_TBLOFF_Pos

```
#define SCB_VTOR_TBLOFF_Pos 7
```

SCB VTOR: TBLOFF Position

5.164.2.308 SCnSCB

```
#define SCnSCB ((SCnSCB_Type *) SCS_BASE )
```

System control Register not in SCB

5.164.2.309 SCnSCB_ACTLR_DISDEFWBUF_Msk

```
#define SCnSCB_ACTLR_DISDEFWBUF_Msk (1UL << SCnSCB_ACTLR_DISDEFWBUF_Pos)
```

ACTLR: DISDEFWBUF Mask

5.164.2.310 SCnSCB_ACTLR_DISDEFWBUF_Pos

```
#define SCnSCB_ACTLR_DISDEFWBUF_Pos 1
```

ACTLR: DISDEFWBUF Position

5.164.2.311 SCnSCB_ACTLR_DISFOLD_Msk

```
#define SCnSCB_ACTLR_DISFOLD_Msk (1UL << SCnSCB_ACTLR_DISFOLD_Pos)
```

ACTLR: DISFOLD Mask

5.164.2.312 SCnSCB_ACTLR_DISFOLD_Pos

```
#define SCnSCB_ACTLR_DISFOLD_Pos 2
```

ACTLR: DISFOLD Position

5.164.2.313 SCnSCB_ACTLR_DISFPCA_Msk

```
#define SCnSCB_ACTLR_DISFPCA_Msk (1UL << SCnSCB_ACTLR_DISFPCA_Pos)
```

ACTLR: DISFPCA Mask

5.164.2.314 SCnSCB_ACTLR_DISFPCA_Pos

```
#define SCnSCB_ACTLR_DISFPCA_Pos 8
```

ACTLR: DISFPCA Position

5.164.2.315 SCnSCB_ACTLR_DISMCYCINT_Msk

```
#define SCnSCB_ACTLR_DISMCYCINT_Msk (1UL /*<< SCnSCB_ACTLR_DISMCYCINT_Pos*/)
```

ACTLR: DISMCYCINT Mask

5.164.2.316 SCnSCB_ACTLR_DISMCYCINT_Pos

```
#define SCnSCB_ACTLR_DISMCYCINT_Pos 0
```

ACTLR: DISMCYCINT Position

5.164.2.317 SCnSCB_ACTLR_DISOOFP_Msk

```
#define SCnSCB_ACTLR_DISOOFP_Msk (1UL << SCnSCB_ACTLR_DISOOFP_Pos)
```

ACTLR: DISOOFP Mask

5.164.2.318 SCnSCB_ACTLR_DISOOFP_Pos

```
#define SCnSCB_ACTLR_DISOOFP_Pos 9
```

ACTLR: DISOOFP Position

5.164.2.319 SCnSCB_ICTR_INTLINESNUM_Msk

```
#define SCnSCB_ICTR_INTLINESNUM_Msk (0xFUL /*<< SCnSCB_ICTR_INTLINESNUM_Pos*/)
```

ICTR: INTLINESNUM Mask

5.164.2.320 SCnSCB_ICTR_INTLINESNUM_Pos

```
#define SCnSCB_ICTR_INTLINESNUM_Pos 0
```

ICTR: INTLINESNUM Position

5.164.2.321 SCS_BASE

```
#define SCS_BASE (0xE000E000UL)
```

System Control Space Base Address

5.164.2.322 SysTick

```
#define SysTick ((SysTick_Type *) SysTick_BASE )
```

SysTick configuration struct

5.164.2.323 SysTick_BASE

```
#define SysTick_BASE (SCS_BASE + 0x0010UL)
```

SysTick Base Address

5.164.2.324 SysTick_CALIB_NOREF_Msk

```
#define SysTick_CALIB_NOREF_Msk (1UL << SysTick_CALIB_NOREF_Pos)
```

SysTick CALIB: NOREF Mask

5.164.2.325 SysTick_CALIB_NOREF_Pos

```
#define SysTick_CALIB_NOREF_Pos 31
```

SysTick CALIB: NOREF Position

5.164.2.326 SysTick_CALIB_SKEW_Msk

```
#define SysTick_CALIB_SKEW_Msk (1UL << SysTick_CALIB_SKEW_Pos)
```

SysTick CALIB: SKEW Mask

5.164.2.327 SysTick_CALIB_SKEW_Pos

```
#define SysTick_CALIB_SKEW_Pos 30
```

SysTick CALIB: SKEW Position

5.164.2.328 SysTick_CALIB_TENMS_Msk

```
#define SysTick_CALIB_TENMS_Msk (0xFFFFFUL /*<< SysTick_CALIB_TENMS_Pos*/)
```

SysTick CALIB: TENMS Mask

5.164.2.329 SysTick_CALIB_TENMS_Pos

```
#define SysTick_CALIB_TENMS_Pos 0
```

SysTick CALIB: TENMS Position

5.164.2.330 SysTick_CTRL_CLKSOURCE_Msk

```
#define SysTick_CTRL_CLKSOURCE_Msk (1UL << SysTick_CTRL_CLKSOURCE_Pos)
```

SysTick CTRL: CLKSOURCE Mask

5.164.2.331 SysTick_CTRL_CLKSOURCE_Pos

```
#define SysTick_CTRL_CLKSOURCE_Pos 2
```

SysTick CTRL: CLKSOURCE Position

5.164.2.332 SysTick_CTRL_COUNTFLAG_Msk

```
#define SysTick_CTRL_COUNTFLAG_Msk (1UL << SysTick_CTRL_COUNTFLAG_Pos)
```

SysTick CTRL: COUNTFLAG Mask

5.164.2.333 SysTick_CTRL_COUNTFLAG_Pos

```
#define SysTick_CTRL_COUNTFLAG_Pos 16
```

SysTick CTRL: COUNTFLAG Position

5.164.2.334 SysTick_CTRL_ENABLE_Msk

```
#define SysTick_CTRL_ENABLE_Msk (1UL /*<< SysTick_CTRL_ENABLE_Pos*/)
```

SysTick CTRL: ENABLE Mask

5.164.2.335 SysTick_CTRL_ENABLE_Pos

```
#define SysTick_CTRL_ENABLE_Pos 0
```

SysTick CTRL: ENABLE Position

5.164.2.336 SysTick_CTRL_TICKINT_Msk

```
#define SysTick_CTRL_TICKINT_Msk (1UL << SysTick_CTRL_TICKINT_Pos)
```

SysTick CTRL: TICKINT Mask

5.164.2.337 SysTick_CTRL_TICKINT_Pos

```
#define SysTick_CTRL_TICKINT_Pos 1
```

SysTick CTRL: TICKINT Position

5.164.2.338 SysTick_LOAD_RELOAD_Msk

```
#define SysTick_LOAD_RELOAD_Msk (0xFFFFFUL /*<< SysTick_LOAD_RELOAD_Pos*/)
```

SysTick LOAD: RELOAD Mask

5.164.2.339 SysTick_LOAD_RELOAD_Pos

```
#define SysTick_LOAD_RELOAD_Pos 0
```

SysTick LOAD: RELOAD Position

5.164.2.340 SysTick_VAL_CURRENT_Msk

```
#define SysTick_VAL_CURRENT_Msk (0xFFFFFFFFUL /*<< SysTick_VAL_CURRENT_Pos*/)
```

SysTick VAL: CURRENT Mask

5.164.2.341 SysTick_VAL_CURRENT_Pos

```
#define SysTick_VAL_CURRENT_Pos 0
```

SysTick VAL: CURRENT Position

5.164.2.342 TPI

```
#define TPI ((TPI_Type *) TPI_BASE )
```

TPI configuration struct

5.164.2.343 TPI_ACPR_PRESCALER_Msk

```
#define TPI_ACPR_PRESCALER_Msk (0x1FFFUL /*<< TPI_ACPR_PRESCALER_Pos*/)
```

TPI ACPR: PRESCALER Mask

5.164.2.344 TPI_ACPR_PRESCALER_Pos

```
#define TPI_ACPR_PRESCALER_Pos 0
```

TPI ACPR: PRESCALER Position

5.164.2.345 TPI_BASE

```
#define TPI_BASE (0xE0040000UL)
```

TPI Base Address

5.164.2.346 TPI_DEVID_AsynClkIn_Msk

```
#define TPI_DEVID_AsynClkIn_Msk (0x1UL << TPI_DEVID_AsynClkIn_Pos)
```

TPI DEVID: AsynClkIn Mask

5.164.2.347 TPI_DEVID_AsynClkIn_Pos

```
#define TPI_DEVID_AsynClkIn_Pos 5
```

TPI DEVID: AsynClkIn Position

5.164.2.348 TPI_DEVID_MANCVALID_Msk

```
#define TPI_DEVID_MANCVALID_Msk (0x1UL << TPI_DEVID_MANCVALID_Pos)
```

TPI DEVID: MANCVALID Mask

5.164.2.349 TPI_DEVID_MANCVALID_Pos

```
#define TPI_DEVID_MANCVALID_Pos 10
```

TPI DEVID: MANCVALID Position

5.164.2.350 TPI_DEVID_MinBufSz_Msk

```
#define TPI_DEVID_MinBufSz_Msk (0x7UL << TPI_DEVID_MinBufSz_Pos)
```

TPI DEVID: MinBufSz Mask

5.164.2.351 TPI_DEVID_MinBufSz_Pos

```
#define TPI_DEVID_MinBufSz_Pos 6
```

TPI DEVID: MinBufSz Position

5.164.2.352 TPI_DEVID_NrTraceInput_Msk

```
#define TPI_DEVID_NrTraceInput_Msk (0x1FUL /*<< TPI_DEVID_NrTraceInput_Pos*/)
```

TPI DEVID: NrTraceInput Mask

5.164.2.353 TPI_DEVID_NrTraceInput_Pos

```
#define TPI_DEVID_NrTraceInput_Pos 0
```

TPI DEVID: NrTraceInput Position

5.164.2.354 TPI_DEVID_NRZVALID_Msk

```
#define TPI_DEVID_NRZVALID_Msk (0x1UL << TPI_DEVID_NRZVALID_Pos)
```

TPI DEVID: NRZVALID Mask

5.164.2.355 TPI_DEVID_NRZVALID_Pos

```
#define TPI_DEVID_NRZVALID_Pos 11
```

TPI DEVID: NRZVALID Position

5.164.2.356 TPI_DEVID_PTINVALIDID_Msk

```
#define TPI_DEVID_PTINVALIDID_Msk (0x1UL << TPI_DEVID_PTINVALIDID_Pos)
```

TPI DEVID: PTINVALIDID Mask

5.164.2.357 TPI_DEVID_PTINVALIDID_Pos

```
#define TPI_DEVID_PTINVALIDID_Pos 9
```

TPI DEVID: PTINVALIDID Position

5.164.2.358 TPI_DEVTYPE_MajorType_Msk

```
#define TPI_DEVTYPE_MajorType_Msk (0xFUL << TPI_DEVTYPE_MajorType_Pos)
```

TPI DEVTYPE: MajorType Mask

5.164.2.359 TPI_DEVTYPE_MajorType_Pos

```
#define TPI_DEVTYPE_MajorType_Pos 4
```

TPI DEVTYPE: MajorType Position

5.164.2.360 TPI_DEVTYPE_SubType_Msk

```
#define TPI_DEVTYPE_SubType_Msk (0xFUL /*<< TPI_DEVTYPE_SubType_Pos*/)
```

TPI DEVTYPE: SubType Mask

5.164.2.361 TPI_DEVTYPE_SubType_Pos

```
#define TPI_DEVTYPE_SubType_Pos 0
```

TPI DEVTYPE: SubType Position

5.164.2.362 TPI_FFCR_EnFCont_Msk

```
#define TPI_FFCR_EnFCont_Msk (0x1UL << TPI_FFCR_EnFCont_Pos)
```

TPI FFCR: EnFCont Mask

5.164.2.363 TPI_FFCR_EnFCont_Pos

```
#define TPI_FFCR_EnFCont_Pos 1
```

TPI FFCR: EnFCont Position

5.164.2.364 TPI_FFCR_TrigIn_Msk

```
#define TPI_FFCR_TrigIn_Msk (0x1UL << TPI_FFCR_TrigIn_Pos)
```

TPI FFCR: TrigIn Mask

5.164.2.365 TPI_FFCR_TrigIn_Pos

```
#define TPI_FFCR_TrigIn_Pos 8
```

TPI FFCR: TrigIn Position

5.164.2.366 TPI_FFSR_FInProg_Msk

```
#define TPI_FFSR_FInProg_Msk (0x1UL /*<< TPI_FFSR_FInProg_Pos*/)
```

TPI FFSR: FInProg Mask

5.164.2.367 TPI_FFSR_FInProg_Pos

```
#define TPI_FFSR_FInProg_Pos 0
```

TPI FFSR: FInProg Position

5.164.2.368 TPI_FFSR_FtNonStop_Msk

```
#define TPI_FFSR_FtNonStop_Msk (0x1UL << TPI_FFSR_FtNonStop_Pos)
```

TPI FFSR: FtNonStop Mask

5.164.2.369 TPI_FFSR_FtNonStop_Pos

```
#define TPI_FFSR_FtNonStop_Pos 3
```

TPI FFSR: FtNonStop Position

5.164.2.370 TPI_FFSR_FtStopped_Msk

```
#define TPI_FFSR_FtStopped_Msk (0x1UL << TPI_FFSR_FtStopped_Pos)
```

TPI FFSR: FtStopped Mask

5.164.2.371 TPI_FFSR_FtStopped_Pos

```
#define TPI_FFSR_FtStopped_Pos 1
```

TPI FFSR: FtStopped Position

5.164.2.372 TPI_FFSR_TCPresent_Msk

```
#define TPI_FFSR_TCPresent_Msk (0x1UL << TPI_FFSR_TCPresent_Pos)
```

TPI FFSR: TCPresent Mask

5.164.2.373 TPI_FFSR_TCPresent_Pos

```
#define TPI_FFSR_TCPresent_Pos 2
```

TPI FFSR: TCPresent Position

5.164.2.374 TPI_FIFO0_ETM0_Msk

```
#define TPI_FIFO0_ETM0_Msk (0xFFUL /*<< TPI_FIFO0_ETM0_Pos*/)
```

TPI FIFO0: ETM0 Mask

5.164.2.375 TPI_FIFO0_ETM0_Pos

```
#define TPI_FIFO0_ETM0_Pos 0
```

TPI FIFO0: ETM0 Position

5.164.2.376 TPI_FIFO0_ETM1_Msk

```
#define TPI_FIFO0_ETM1_Msk (0xFFUL << TPI_FIFO0_ETM1_Pos)
```

TPI FIFO0: ETM1 Mask

5.164.2.377 TPI_FIFO0_ETM1_Pos

```
#define TPI_FIFO0_ETM1_Pos 8
```

TPI FIFO0: ETM1 Position

5.164.2.378 TPI_FIFO0_ETM2_Msk

```
#define TPI_FIFO0_ETM2_Msk (0xFFUL << TPI_FIFO0_ETM2_Pos)
```

TPI FIFO0: ETM2 Mask

5.164.2.379 TPI_FIFO0_ETM2_Pos

```
#define TPI_FIFO0_ETM2_Pos 16
```

TPI FIFO0: ETM2 Position

5.164.2.380 TPI_FIFO0_ETM_ATVALID_Msk

```
#define TPI_FIFO0_ETM_ATVALID_Msk (0x3UL << TPI_FIFO0_ETM_ATVALID_Pos)
```

TPI FIFO0: ETM_ATVALID Mask

5.164.2.381 TPI_FIFO0_ETM_ATVALID_Pos

```
#define TPI_FIFO0_ETM_ATVALID_Pos 26
```

TPI FIFO0: ETM_ATVALID Position

5.164.2.382 TPI_FIFO0_ETM_bytectcount_Msk

```
#define TPI_FIFO0_ETM_bytectcount_Msk (0x3UL << TPI_FIFO0_ETM_bytectcount_Pos)
```

TPI FIFO0: ETM_bytectcount Mask

5.164.2.383 TPI_FIFO0_ETM_bytectcount_Pos

```
#define TPI_FIFO0_ETM_bytectcount_Pos 24
```

TPI FIFO0: ETM_bytectcount Position

5.164.2.384 TPI_FIFO0_ITM_ATVALID_Msk

```
#define TPI_FIFO0_ITM_ATVALID_Msk (0x3UL << TPI_FIFO0_ITM_ATVALID_Pos)
```

TPI FIFO0: ITM_ATVALID Mask

5.164.2.385 TPI_FIFO0_ITM_ATVALID_Pos

```
#define TPI_FIFO0_ITM_ATVALID_Pos 29
```

TPI FIFO0: ITM_ATVALID Position

5.164.2.386 TPI_FIFO0_ITM_bytectcount_Msk

```
#define TPI_FIFO0_ITM_bytectcount_Msk (0x3UL << TPI_FIFO0_ITM_bytectcount_Pos)
```

TPI FIFO0: ITM_bytectcount Mask

5.164.2.387 TPI_FIFO0_ITM_bytectcount_Pos

```
#define TPI_FIFO0_ITM_bytectcount_Pos 27
```

TPI FIFO0: ITM_bytectcount Position

5.164.2.388 TPI_FIFO1_ETM_ATVALID_Msk

```
#define TPI_FIFO1_ETM_ATVALID_Msk (0x3UL << TPI_FIFO1_ETM_ATVALID_Pos)
```

TPI FIFO1: ETM_ATVALID Mask

5.164.2.389 TPI_FIFO1_ETM_ATVALID_Pos

```
#define TPI_FIFO1_ETM_ATVALID_Pos 26
```

TPI FIFO1: ETM_ATVALID Position

5.164.2.390 TPI_FIFO1_ETM_bytectcount_Msk

```
#define TPI_FIFO1_ETM_bytectcount_Msk (0x3UL << TPI_FIFO1_ETM_bytectcount_Pos)
```

TPI FIFO1: ETM_bytectcount Mask

5.164.2.391 TPI_FIFO1_ETM_bytectcount_Pos

```
#define TPI_FIFO1_ETM_bytectcount_Pos 24
```

TPI FIFO1: ETM_bytectcount Position

5.164.2.392 TPI_FIFO1_ITM0_Msk

```
#define TPI_FIFO1_ITM0_Msk (0xFFUL /*<< TPI_FIFO1_ITM0_Pos*/)
```

TPI FIFO1: ITM0 Mask

5.164.2.393 TPI_FIFO1_ITM0_Pos

```
#define TPI_FIFO1_ITM0_Pos 0
```

TPI FIFO1: ITM0 Position

5.164.2.394 TPI_FIFO1_ITM1_Msk

```
#define TPI_FIFO1_ITM1_Msk (0xFFUL << TPI_FIFO1_ITM1_Pos)
```

TPI FIFO1: ITM1 Mask

5.164.2.395 TPI_FIFO1_ITM1_Pos

```
#define TPI_FIFO1_ITM1_Pos 8
```

TPI FIFO1: ITM1 Position

5.164.2.396 TPI_FIFO1_ITM2_Msk

```
#define TPI_FIFO1_ITM2_Msk (0xFFUL << TPI_FIFO1_ITM2_Pos)
```

TPI FIFO1: ITM2 Mask

5.164.2.397 TPI_FIFO1_ITM2_Pos

```
#define TPI_FIFO1_ITM2_Pos 16
```

TPI FIFO1: ITM2 Position

5.164.2.398 TPI_FIFO1_ITM_ATVALID_Msk

```
#define TPI_FIFO1_ITM_ATVALID_Msk (0x3UL << TPI_FIFO1_ITM_ATVALID_Pos)
```

TPI FIFO1: ITM_ATVALID Mask

5.164.2.399 TPI_FIFO1_ITM_ATVALID_Pos

```
#define TPI_FIFO1_ITM_ATVALID_Pos 29
```

TPI FIFO1: ITM_ATVALID Position

5.164.2.400 TPI_FIFO1_ITM_bytectcount_Msk

```
#define TPI_FIFO1_ITM_bytectcount_Msk (0x3UL << TPI_FIFO1_ITM_bytectcount_Pos)
```

TPI FIFO1: ITM_bytectcount Mask

5.164.2.401 TPI_FIFO1_ITM_bytectcount_Pos

```
#define TPI_FIFO1_ITM_bytectcount_Pos 27
```

TPI FIFO1: ITM_bytectcount Position

5.164.2.402 TPI_ITATBCTR0_ATREADY_Msk

```
#define TPI_ITATBCTR0_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR0_ATREADY_Pos*/)
```

TPI ITATBCTR0: ATREADY Mask

5.164.2.403 TPI_ITATBCTR0_ATREADY_Pos

```
#define TPI_ITATBCTR0_ATREADY_Pos 0
```

TPI ITATBCTR0: ATREADY Position

5.164.2.404 TPI_ITATBCTR2_ATREADY_Msk

```
#define TPI_ITATBCTR2_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR2_ATREADY_Pos*/)
```

TPI ITATBCTR2: ATREADY Mask

5.164.2.405 TPI_ITATBCTR2_ATREADY_Pos

```
#define TPI_ITATBCTR2_ATREADY_Pos 0
```

TPI ITATBCTR2: ATREADY Position

5.164.2.406 TPI_ITCTRL_Mode_Msk

```
#define TPI_ITCTRL_Mode_Msk (0x1UL /*<< TPI_ITCTRL_Mode_Pos*/)
```

TPI ITCTRL: Mode Mask

5.164.2.407 TPI_ITCTRL_Mode_Pos

```
#define TPI_ITCTRL_Mode_Pos 0
```

TPI ITCTRL: Mode Position

5.164.2.408 TPI_SPPR_TXMODE_Msk

```
#define TPI_SPPR_TXMODE_Msk (0x3UL /*<< TPI_SPPR_TXMODE_Pos*/)
```

TPI SPPR: TXMODE Mask

5.164.2.409 TPI_SPPR_TXMODE_Pos

```
#define TPI_SPPR_TXMODE_Pos 0
```

TPI SPPR: TXMODE Position

5.164.2.410 TPI_TRIGGER_TRIGGER_Msk

```
#define TPI_TRIGGER_TRIGGER_Msk (0x1UL /*<< TPI_TRIGGER_TRIGGER_Pos*/)
```

TPI TRIGGER: TRIGGER Mask

5.164.2.411 TPI_TRIGGER_TRIGGER_Pos

```
#define TPI_TRIGGER_TRIGGER_Pos 0
```

TPI TRIGGER: TRIGGER Position

5.164.2.412 xPSR_C_Msk

```
#define xPSR_C_Msk (1UL << xPSR_C_Pos)
```

xPSR: C Mask

5.164.2.413 xPSR_C_Pos

```
#define xPSR_C_Pos 29
```

xPSR: C Position

5.164.2.414 xPSR_GE_Msk

```
#define xPSR_GE_Msk (0xFUL << xPSR_GE_Pos)
```

xPSR: GE Mask

5.164.2.415 xPSR_GE_Pos

```
#define xPSR_GE_Pos 16
```

xPSR: GE Position

5.164.2.416 xPSR_ISR_Msk

```
#define xPSR_ISR_Msk (0x1FFUL /*<< xPSR_ISR_Pos*/)
```

xPSR: ISR Mask

5.164.2.417 xPSR_ISR_Pos

```
#define xPSR_ISR_Pos 0
```

xPSR: ISR Position

5.164.2.418 xPSR_IT_Msk

```
#define xPSR_IT_Msk (3UL << xPSR_IT_Pos)
```

xPSR: IT Mask

5.164.2.419 xPSR_IT_Pos

```
#define xPSR_IT_Pos 25
```

xPSR: IT Position

5.164.2.420 xPSR_N_Msk

```
#define xPSR_N_Msk (1UL << xPSR_N_Pos)
```

xPSR: N Mask

5.164.2.421 xPSR_N_Pos

```
#define xPSR_N_Pos 31
```

xPSR: N Position

5.164.2.422 xPSR_Q_Msk

```
#define xPSR_Q_Msk (1UL << xPSR_Q_Pos)
```

xPSR: Q Mask

5.164.2.423 xPSR_Q_Pos

```
#define xPSR_Q_Pos 27
```

xPSR: Q Position

5.164.2.424 xPSR_T_Msk

```
#define xPSR_T_Msk (1UL << xPSR_T_Pos)
```

xPSR: T Mask

5.164.2.425 xPSR_T_Pos

```
#define xPSR_T_Pos 24
```

xPSR: T Position

5.164.2.426 xPSR_V_Msk

```
#define xPSR_V_Msk (1UL << xPSR_V_Pos)
```

xPSR: V Mask

5.164.2.427 xPSR_V_Pos

```
#define xPSR_V_Pos 28
```

xPSR: V Position

5.164.2.428 xPSR_Z_Msk

```
#define xPSR_Z_Msk (1UL << xPSR_Z_Pos)
```

xPSR: Z Mask

5.164.2.429 xPSR_Z_Pos

```
#define xPSR_Z_Pos 30
```

xPSR: Z Position

5.164.3 Function Documentation

5.164.3.1 ITM_CheckChar()

```
__STATIC_INLINE int32_t ITM_CheckChar (
    void )
```

ITM Check Character.

The function checks whether a character is pending for reading in the variable [ITM_RxBuffer](#).

Returns

- 0 No character available.
- 1 Character available.

5.164.3.2 ITM_ReceiveChar()

```
__STATIC_INLINE int32_t ITM_ReceiveChar (
    void )
```

ITM Receive Character.

The function inputs a character via the external variable [ITM_RxBuffer](#).

Returns

- Received character.
- 1 No character pending.

5.164.3.3 ITM_SendChar()

```
__STATIC_INLINE uint32_t ITM_SendChar (
    uint32_t ch )
```

ITM Send Character.

The function transmits a character via the ITM channel 0, and

- Just returns when no debugger is connected that has booked the output.
- Is blocking when a debugger is connected, but the previous character sent has not been transmitted.

Parameters

in	<i>ch</i>	Character to transmit.
----	-----------	------------------------

Returns

Character to transmit.

5.164.3.4 NVIC_ClearPendingIRQ()

```
__STATIC_INLINE void NVIC_ClearPendingIRQ (
    IRQn_Type IRQn )
```

Clear Pending Interrupt.

The function clears the pending bit of an external interrupt.

Parameters

in	<i>IRQn</i>	External interrupt number. Value cannot be negative.
----	-------------	--

5.164.3.5 NVIC_DecodePriority()

```
__STATIC_INLINE void NVIC_DecodePriority (
    uint32_t Priority,
    uint32_t PriorityGroup,
    uint32_t * pPreemptPriority,
    uint32_t * pSubPriority )
```

Decode Priority.

The function decodes an interrupt priority value with a given priority group to preemptive priority value and subpriority value. In case of a conflict between priority grouping and available priority bits (`__NVIC_PRIO_BITS`) the smallest possible priority group is set.

Parameters

in	<i>Priority</i>	Priority value, which can be retrieved with the function NVIC_GetPriority() .
in	<i>PriorityGroup</i>	Used priority group.
out	<i>pPreemptPriority</i>	Preemptive priority value (starting from 0).
out	<i>pSubPriority</i>	Subpriority value (starting from 0).

5.164.3.6 NVIC_DisableIRQ()

```
__STATIC_INLINE void NVIC_DisableIRQ (
    IRQn_Type IRQn )
```

Disable External Interrupt.

The function disables a device-specific interrupt in the NVIC interrupt controller.

Parameters

in	<i>IRQn</i>	External interrupt number. Value cannot be negative.
----	-------------	--

5.164.3.7 NVIC_EnableIRQ()

```
__STATIC_INLINE void NVIC_EnableIRQ (
    IRQn_Type IRQn )
```

Enable External Interrupt.

The function enables a device-specific interrupt in the NVIC interrupt controller.

Parameters

in	<i>IRQn</i>	External interrupt number. Value cannot be negative.
----	-------------	--

5.164.3.8 NVIC_EncodePriority()

```
__STATIC_INLINE uint32_t NVIC_EncodePriority (
    uint32_t PriorityGroup,
    uint32_t PreemptPriority,
    uint32_t SubPriority )
```

Encode Priority.

The function encodes the priority for an interrupt with the given priority group, preemptive priority value, and sub-priority value. In case of a conflict between priority grouping and available priority bits (`__NVIC_PRIO_BITS`), the smallest possible priority group is set.

Parameters

in	<i>PriorityGroup</i>	Used priority group.
in	<i>PreemptPriority</i>	Preemptive priority value (starting from 0).
in	<i>SubPriority</i>	Subpriority value (starting from 0).

Returns

Encoded priority. Value can be used in the function [NVIC_SetPriority\(\)](#).

5.164.3.9 NVIC_GetActive()

```
__STATIC_INLINE uint32_t NVIC_GetActive (
    IRQn_Type IRQn )
```

Get Active Interrupt.

The function reads the active register in NVIC and returns the active bit.

Parameters

in	<i>IRQn</i>	Interrupt number.
----	-------------	-------------------

Returns

- 0 Interrupt status is not active.
- 1 Interrupt status is active.

5.164.3.10 NVIC_GetPendingIRQ()

```
__STATIC_INLINE uint32_t NVIC_GetPendingIRQ (
    IRQn_Type IRQn )
```

Get Pending Interrupt.

The function reads the pending register in the NVIC and returns the pending bit for the specified interrupt.

Parameters

in	<i>IRQn</i>	Interrupt number.
----	-------------	-------------------

Returns

- 0 Interrupt status is not pending.
- 1 Interrupt status is pending.

5.164.3.11 NVIC_GetPriority()

```
__STATIC_INLINE uint32_t NVIC_GetPriority (
    IRQn_Type IRQn )
```

Get Interrupt Priority.

The function reads the priority of an interrupt. The interrupt number can be positive to specify an external (device specific) interrupt, or negative to specify an internal (core) interrupt.

Parameters

in	<i>IRQn</i>	Interrupt number.
----	-------------	-------------------

Returns

Interrupt Priority. Value is aligned automatically to the implemented priority bits of the microcontroller.

5.164.3.12 NVIC_GetPriorityGrouping()

```
__STATIC_INLINE uint32_t NVIC_GetPriorityGrouping (
    void )
```

Get Priority Grouping.

The function reads the priority grouping field from the NVIC Interrupt Controller.

Returns

Priority grouping field (SCB->AIRCR [10:8] PRIGROUP field).

5.164.3.13 NVIC_SetPendingIRQ()

```
__STATIC_INLINE void NVIC_SetPendingIRQ (
    IRQn_Type IRQn )
```

Set Pending Interrupt.

The function sets the pending bit of an external interrupt.

Parameters

in	<i>IRQn</i>	Interrupt number. Value cannot be negative.
----	-------------	---

5.164.3.14 NVIC_SetPriority()

```
__STATIC_INLINE void NVIC_SetPriority (
    IRQn_Type IRQn,
    uint32_t priority )
```

Set Interrupt Priority.

The function sets the priority of an interrupt.

Note

The priority cannot be set for every core interrupt.

Parameters

in	<i>IRQn</i>	Interrupt number.
in	<i>priority</i>	Priority to set.

5.164.3.15 NVIC_SetPriorityGrouping()

```
__STATIC_INLINE void NVIC_SetPriorityGrouping (
    uint32_t PriorityGroup )
```

Set Priority Grouping.

The function sets the priority grouping field using the required unlock sequence. The parameter *PriorityGroup* is assigned to the field SCB->AIRCR [10:8] PRIGROUP field. Only values from 0..7 are used. In case of a conflict between priority grouping and available priority bits (*__NVIC_PRIO_BITS*), the smallest possible priority group is set.

Parameters

in	<i>PriorityGroup</i>	Priority grouping field.
----	----------------------	--------------------------

5.164.3.16 NVIC_SystemReset()

```
__STATIC_INLINE void NVIC_SystemReset (
    void )
```

System Reset.

The function initiates a system reset request to reset the MCU.

5.164.3.17 **SysTick_Config()**

```
__STATIC_INLINE uint32_t SysTick_Config (
    uint32_t ticks )
```

System Tick Configuration.

The function initializes the System Timer and its interrupt, and starts the System Tick Timer. Counter is in free running mode to generate periodic interrupts.

Parameters

in	<i>ticks</i>	Number of ticks between two interrupts.
----	--------------	---

Returns

0 Function succeeded.

1 Function failed.

Note

When the variable `__Vendor_SysTickConfig` is set to 1, then the function **SysTick_Config** is not included. In this case, the file **device.h** must contain a vendor-specific implementation of this function.

5.164.3.18 **SysTick_Handler()**

```
void SysTick_Handler (
    void )
```

SysTick interrupt handler.

This function is called when the SysTick timer interrupt occurs. It handles the interrupt by calling the user-defined callback function and, if necessary, stopping the timer based on the running mode.

Parameters

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

Returns

None

5.164.4 Variable Documentation

5.164.4.1 ACR

```
__IO uint32_t FLASH_TypeDef::ACR
```

FLASH access control register, Address offset: 0x00

5.164.4.2 AFR

```
__IO uint32_t GPIO_TypeDef::AFR[2]
```

GPIO alternate function registers, Address offset: 0x20-0x24

5.164.4.3 AHB1ENR

```
__IO uint32_t RCC_TypeDef::AHB1ENR
```

RCC AHB1 peripheral clock register, Address offset: 0x30

5.164.4.4 AHB1LPENR

```
__IO uint32_t RCC_TypeDef::AHB1LPENR
```

RCC AHB1 peripheral clock enable in low power mode register, Address offset: 0x50

5.164.4.5 AHB1RSTR

```
__IO uint32_t RCC_TypeDef::AHB1RSTR
```

RCC AHB1 peripheral reset register, Address offset: 0x10

5.164.4.6 AHB2ENR

```
__IO uint32_t RCC_TypeDef::AHB2ENR
```

RCC AHB2 peripheral clock register, Address offset: 0x34

5.164.4.7 AHB2LPENR

`__IO uint32_t RCC_TypeDef::AHB2LPENR`

RCC AHB2 peripheral clock enable in low power mode register, Address offset: 0x54

5.164.4.8 AHB2RSTR

`__IO uint32_t RCC_TypeDef::AHB2RSTR`

RCC AHB2 peripheral reset register, Address offset: 0x14

5.164.4.9 AHB3ENR

`__IO uint32_t RCC_TypeDef::AHB3ENR`

RCC AHB3 peripheral clock register, Address offset: 0x38

5.164.4.10 AHB3LPENR

`__IO uint32_t RCC_TypeDef::AHB3LPENR`

RCC AHB3 peripheral clock enable in low power mode register, Address offset: 0x58

5.164.4.11 AHB3RSTR

`__IO uint32_t RCC_TypeDef::AHB3RSTR`

RCC AHB3 peripheral reset register, Address offset: 0x18

5.164.4.12 ALRMAR

`__IO uint32_t RTC_TypeDef::ALRMAR`

RTC alarm A register, Address offset: 0x1C

5.164.4.13 ALRMASSR

`__IO uint32_t RTC_TypeDef::ALRMASSR`

RTC alarm A sub second register, Address offset: 0x44

5.164.4.14 ALRMBR

`__IO uint32_t RTC_TypeDef::ALRMBR`

RTC alarm B register, Address offset: 0x20

5.164.4.15 ALRMBSSR

```
__IO uint32_t RTC_TypeDef::ALRMBSSR
```

RTC alarm B sub second register, Address offset: 0x48

5.164.4.16 AMTCR

```
__IO uint32_t DMA2D_TypeDef::AMTCR
```

DMA2D AHB Master Timer Configuration Register, Address offset: 0x4C

5.164.4.17 APB1ENR

```
__IO uint32_t RCC_TypeDef::APB1ENR
```

RCC APB1 peripheral clock enable register, Address offset: 0x40

5.164.4.18 APB1FZ

```
__IO uint32_t DBGMCU_TypeDef::APB1FZ
```

Debug MCU APB1 freeze register, Address offset: 0x08

5.164.4.19 APB1LPENR

```
__IO uint32_t RCC_TypeDef::APB1LPENR
```

RCC APB1 peripheral clock enable in low power mode register, Address offset: 0x60

5.164.4.20 APB1RSTR

```
__IO uint32_t RCC_TypeDef::APB1RSTR
```

RCC APB1 peripheral reset register, Address offset: 0x20

5.164.4.21 APB2ENR

```
__IO uint32_t RCC_TypeDef::APB2ENR
```

RCC APB2 peripheral clock enable register, Address offset: 0x44

5.164.4.22 APB2FZ

```
__IO uint32_t DBGMCU_TypeDef::APB2FZ
```

Debug MCU APB2 freeze register, Address offset: 0x0C

5.164.4.23 APB2LPENR

`__IO uint32_t RCC_TypeDef::APB2LPENR`

RCC APB2 peripheral clock enable in low power mode register, Address offset: 0x64

5.164.4.24 APB2RSTR

`__IO uint32_t RCC_TypeDef::APB2RSTR`

RCC APB2 peripheral reset register, Address offset: 0x24

5.164.4.25 ARG

`__IO uint32_t SDIO_TypeDef::ARG`

SDIO argument register, Address offset: 0x08

5.164.4.26 ARR

`__IO uint32_t TIM_TypeDef::ARR`

TIM auto-reload register, Address offset: 0x2C

5.164.4.27 AWCR

`__IO uint32_t LTDC_TypeDef::AWCR`

LTDC Active Width Configuration Register, Address offset: 0x10

5.164.4.28 BCCR

`__IO uint32_t LTDC_TypeDef::BCCR`

LTDC Background Color Configuration Register, Address offset: 0x2C

5.164.4.29 BDCR

`__IO uint32_t RCC_TypeDef::BDCR`

RCC Backup domain control register, Address offset: 0x70

5.164.4.30 BDTR

`__IO uint16_t TIM_TypeDef::BDTR`

TIM break and dead-time register, Address offset: 0x44

5.164.4.31 BFCR

```
__IO uint32_t LTDC_Layer_TypeDef::BFCR
```

LTDC Layerx Blending Factors Configuration Register Address offset: 0xA0

5.164.4.32 BGCLUT

```
__IO uint32_t DMA2D_TypeDef::BGCLUT[256]
```

DMA2D Background CLUT, Address offset: 800-BFF

5.164.4.33 BGCMAR

```
__IO uint32_t DMA2D_TypeDef::BGCMAR
```

DMA2D Background CLUT Memory Address Register, Address offset: 0x30

5.164.4.34 BGCOLR

```
__IO uint32_t DMA2D_TypeDef::BGCOLR
```

DMA2D Background Color Register, Address offset: 0x28

5.164.4.35 BGMAR

```
__IO uint32_t DMA2D_TypeDef::BGMAR
```

DMA2D Background Memory Address Register, Address offset: 0x14

5.164.4.36 BGOR

```
__IO uint32_t DMA2D_TypeDef::BGOR
```

DMA2D Background Offset Register, Address offset: 0x18

5.164.4.37 BGPFCCR

```
__IO uint32_t DMA2D_TypeDef::BGPFCCR
```

DMA2D Background PFC Control Register, Address offset: 0x24

5.164.4.38 BKPOR

```
__IO uint32_t RTC_TypeDef::BKPOR
```

RTC backup register 1, Address offset: 0x50

5.164.4.39 BKP10R

```
__IO uint32_t RTC_TypeDef::BKP10R
```

RTC backup register 10, Address offset: 0x78

5.164.4.40 BKP11R

```
__IO uint32_t RTC_TypeDef::BKP11R
```

RTC backup register 11, Address offset: 0x7C

5.164.4.41 BKP12R

```
__IO uint32_t RTC_TypeDef::BKP12R
```

RTC backup register 12, Address offset: 0x80

5.164.4.42 BKP13R

```
__IO uint32_t RTC_TypeDef::BKP13R
```

RTC backup register 13, Address offset: 0x84

5.164.4.43 BKP14R

```
__IO uint32_t RTC_TypeDef::BKP14R
```

RTC backup register 14, Address offset: 0x88

5.164.4.44 BKP15R

```
__IO uint32_t RTC_TypeDef::BKP15R
```

RTC backup register 15, Address offset: 0x8C

5.164.4.45 BKP16R

```
__IO uint32_t RTC_TypeDef::BKP16R
```

RTC backup register 16, Address offset: 0x90

5.164.4.46 BKP17R

```
__IO uint32_t RTC_TypeDef::BKP17R
```

RTC backup register 17, Address offset: 0x94

5.164.4.47 BKP18R

```
__IO uint32_t RTC_TypeDef::BKP18R
```

RTC backup register 18, Address offset: 0x98

5.164.4.48 BKP19R

```
__IO uint32_t RTC_TypeDef::BKP19R
```

RTC backup register 19, Address offset: 0x9C

5.164.4.49 BKP1R

```
__IO uint32_t RTC_TypeDef::BKP1R
```

RTC backup register 1, Address offset: 0x54

5.164.4.50 BKP2R

```
__IO uint32_t RTC_TypeDef::BKP2R
```

RTC backup register 2, Address offset: 0x58

5.164.4.51 BKP3R

```
__IO uint32_t RTC_TypeDef::BKP3R
```

RTC backup register 3, Address offset: 0x5C

5.164.4.52 BKP4R

```
__IO uint32_t RTC_TypeDef::BKP4R
```

RTC backup register 4, Address offset: 0x60

5.164.4.53 BKP5R

```
__IO uint32_t RTC_TypeDef::BKP5R
```

RTC backup register 5, Address offset: 0x64

5.164.4.54 BKP6R

```
__IO uint32_t RTC_TypeDef::BKP6R
```

RTC backup register 6, Address offset: 0x68

5.164.4.55 BKP7R

`__IO uint32_t RTC_TypeDef::BKP7R`

RTC backup register 7, Address offset: 0x6C

5.164.4.56 BKP8R

`__IO uint32_t RTC_TypeDef::BKP8R`

RTC backup register 8, Address offset: 0x70

5.164.4.57 BKP9R

`__IO uint32_t RTC_TypeDef::BKP9R`

RTC backup register 9, Address offset: 0x74

5.164.4.58 BPCR

`__IO uint32_t LTDC_TypeDef::BPCR`

LTDC Back Porch Configuration Register, Address offset: 0x0C

5.164.4.59 BRR

`__IO uint16_t USART_TypeDef::BRR`

USART Baud rate register, Address offset: 0x08

5.164.4.60 BSRRH

`__IO uint16_t GPIO_TypeDef::BSRRH`

GPIO port bit set/reset high register, Address offset: 0x1A

5.164.4.61 BSRRL

`__IO uint16_t GPIO_TypeDef::BSRRL`

GPIO port bit set/reset low register, Address offset: 0x18

5.164.4.62 BTR

```
__IO uint32_t CAN_TypeDef::BTR
```

CAN bit timing register, Address offset: 0x1C

5.164.4.63 CACR

```
__IO uint32_t LTDC_Layer_TypeDef::CACR
```

LTDC Layerx Constant Alpha Configuration Register Address offset: 0x98

5.164.4.64 CALIBR

```
__IO uint32_t RTC_TypeDef::CALIBR
```

RTC calibration register, Address offset: 0x18

5.164.4.65 CALR

```
__IO uint32_t RTC_TypeDef::CALR
```

RTC calibration register, Address offset: 0x3C

5.164.4.66 CCER

```
__IO uint16_t TIM_TypeDef::CCER
```

TIM capture/compare enable register, Address offset: 0x20

5.164.4.67 CCMR1

```
__IO uint16_t TIM_TypeDef::CCMR1
```

TIM capture/compare mode register 1, Address offset: 0x18

5.164.4.68 CCMR2

```
__IO uint16_t TIM_TypeDef::CCMR2
```

TIM capture/compare mode register 2, Address offset: 0x1C

5.164.4.69 CCR [1/2]

`__IO uint32_t ADC_Common_TypeDef::CCR`

ADC common control register, Address offset: ADC1 base address + 0x304

5.164.4.70 CCR [2/2]

`__IO uint16_t I2C_TypeDef::CCR`

I2C Clock control register, Address offset: 0x1C

5.164.4.71 CCR1

`__IO uint32_t TIM_TypeDef::CCR1`

TIM capture/compare register 1, Address offset: 0x34

5.164.4.72 CCR2

`__IO uint32_t TIM_TypeDef::CCR2`

TIM capture/compare register 2, Address offset: 0x38

5.164.4.73 CCR3

`__IO uint32_t TIM_TypeDef::CCR3`

TIM capture/compare register 3, Address offset: 0x3C

5.164.4.74 CCR4

`__IO uint32_t TIM_TypeDef::CCR4`

TIM capture/compare register 4, Address offset: 0x40

5.164.4.75 CDR

`__IO uint32_t ADC_Common_TypeDef::CDR`

ADC common regular data register for dual AND triple modes, Address offset: ADC1 base address + 0x308

5.164.4.76 CDSR

`__IO uint32_t LTDC_TypeDef::CDSR`

LTDC Current Display Status Register, Address offset: 0x48

5.164.4.77 CFBAR

```
__IO uint32_t LTDC_Layer_TypeDef::CFBAR
```

LTDC Layerx Color Frame Buffer Address Register Address offset: 0xAC

5.164.4.78 CFBLNR

```
__IO uint32_t LTDC_Layer_TypeDef::CFBLNR
```

LTDC Layerx ColorFrame Buffer Line Number Register Address offset: 0xB4

5.164.4.79 CFBLR

```
__IO uint32_t LTDC_Layer_TypeDef::CFBLR
```

LTDC Layerx Color Frame Buffer Length Register Address offset: 0xB0

5.164.4.80 CFGR

```
__IO uint32_t RCC_TypeDef::CFGCR
```

RCC clock configuration register, Address offset: 0x08

5.164.4.81 CFR

```
__IO uint32_t WWDG_TypeDef::CFR
```

WWDG Configuration register, Address offset: 0x04

5.164.4.82 CIR

```
__IO uint32_t RCC_TypeDef::CIR
```

RCC clock interrupt register, Address offset: 0x0C

5.164.4.83 CKCR

```
__IO uint32_t LTDC_Layer_TypeDef::CKCR
```

LTDC Layerx Color Keying Configuration Register Address offset: 0x90

5.164.4.84 CKGATENR

```
__IO uint32_t RCC_TypeDef::CKGATENR
```

RCC Clocks Gated Enable Register, Address offset: 0x90

5.164.4.85 CLKCR

`__IO uint32_t SDIO_TypeDef::CLKCR`

SDI clock control register, Address offset: 0x04

5.164.4.86 CLRFR

`__IO uint32_t SAI_Block_TypeDef::CLRFR`

SAI block x clear flag register, Address offset: 0x1C

5.164.4.87 CLUTWR

`__IO uint32_t LTDC_Layer_TypeDef::CLUTWR`

LTDC Layerx CLUT Write Register Address offset: 0x144

5.164.4.88 CMD

`__IO uint32_t SDIO_TypeDef::CMD`

SDIO command register, Address offset: 0x0C

5.164.4.89 CMPCR

`__IO uint32_t SYSCFG_TypeDef::CMPCR`

SYSCFG Compensation cell control register, Address offset: 0x20

5.164.4.90 CNT

`__IO uint32_t TIM_TypeDef::CNT`

TIM counter register, Address offset: 0x24

5.164.4.91 CPSR

`__IO uint32_t LTDC_TypeDef::CPSR`

LTDC Current Position Status Register, Address offset: 0x44

5.164.4.92 CR [1/15]

```
__IO uint32_t CRC_TypeDef::CR
```

CRC Control register, Address offset: 0x08

5.164.4.93 CR [2/15]

```
__IO uint32_t DAC_TypeDef::CR
```

DAC control register, Address offset: 0x00

5.164.4.94 CR [3/15]

```
__IO uint32_t DBGMCU_TypeDef::CR
```

Debug MCU configuration register, Address offset: 0x04

5.164.4.95 CR [4/15]

```
__IO uint32_t DCMI_TypeDef::CR
```

DCMI control register 1, Address offset: 0x00

5.164.4.96 CR [5/15]

```
__IO uint32_t DMA_Stream_TypeDef::CR
```

DMA stream x configuration register

5.164.4.97 CR [6/15]

```
__IO uint32_t DMA2D_TypeDef::CR
```

DMA2D Control Register, Address offset: 0x00

5.164.4.98 CR [7/15]

```
__IO uint32_t FLASH_TypeDef::CR
```

FLASH control register, Address offset: 0x10

5.164.4.99 CR [8/15]

```
__IO uint32_t LTDC_Layer_TypeDef::CR
```

LTDC Layerx Control Register Address offset: 0x84

5.164.4.100 CR [9/15]

```
__IO uint32_t PWR_TypeDef::CR
```

PWR power control register, Address offset: 0x00

5.164.4.101 CR [10/15]

```
__IO uint32_t RCC_TypeDef::CR
```

RCC clock control register, Address offset: 0x00

5.164.4.102 CR [11/15]

```
__IO uint32_t RTC_TypeDef::CR
```

RTC control register, Address offset: 0x08

5.164.4.103 CR [12/15]

```
__IO uint32_t WWDG_TypeDef::CR
```

WWDG Control register, Address offset: 0x00

5.164.4.104 CR [13/15]

```
__IO uint32_t CRYP_TypeDef::CR
```

CRYP control register, Address offset: 0x00

5.164.4.105 CR [14/15]

```
__IO uint32_t HASH_TypeDef::CR
```

HASH control register, Address offset: 0x00

5.164.4.106 CR [15/15]

```
__IO uint32_t RNG_TypeDef::CR
```

RNG control register, Address offset: 0x00

5.164.4.107 CR1 [1/6]

```
__IO uint32_t ADC_TypeDef::CR1
```

ADC control register 1, Address offset: 0x04

5.164.4.108 CR1 [2/6]

```
__IO uint16_t I2C_TypeDef::CR1
```

I2C Control register 1, Address offset: 0x00

5.164.4.109 CR1 [3/6]

```
__IO uint32_t SAI_Block_TypeDef::CR1
```

SAI block x configuration register 1, Address offset: 0x04

5.164.4.110 CR1 [4/6]

```
__IO uint16_t SPI_TypeDef::CR1
```

SPI control register 1 (not used in I2S mode), Address offset: 0x00

5.164.4.111 CR1 [5/6]

```
__IO uint16_t TIM_TypeDef::CR1
```

TIM control register 1, Address offset: 0x00

5.164.4.112 CR1 [6/6]

```
__IO uint16_t USART_TypeDef::CR1
```

USART Control register 1, Address offset: 0x0C

5.164.4.113 CR2 [1/6]

```
__IO uint32_t ADC_TypeDef::CR2
```

ADC control register 2, Address offset: 0x08

5.164.4.114 CR2 [2/6]

```
__IO uint16_t I2C_TypeDef::CR2
```

I2C Control register 2, Address offset: 0x04

5.164.4.115 CR2 [3/6]

```
__IO uint32_t SAI_Block_TypeDef::CR2
```

SAI block x configuration register 2, Address offset: 0x08

5.164.4.116 CR2 [4/6]

```
__IO uint16_t SPI_TypeDef::CR2
```

SPI control register 2, Address offset: 0x04

5.164.4.117 CR2 [5/6]

```
__IO uint16_t TIM_TypeDef::CR2
```

TIM control register 2, Address offset: 0x04

5.164.4.118 CR2 [6/6]

```
__IO uint16_t USART_TypeDef::CR2
```

USART Control register 2, Address offset: 0x10

5.164.4.119 CR3

```
__IO uint16_t USART_TypeDef::CR3
```

USART Control register 3, Address offset: 0x14

5.164.4.120 CRCPR

```
__IO uint16_t SPI_TypeDef::CRCPR
```

SPI CRC polynomial register (not used in I2S mode), Address offset: 0x10

5.164.4.121 CSGCM0R

```
__IO uint32_t CRYP_TypeDef::CSGCM0R
```

CRYP GCM/GMAC context swap register 0, Address offset: 0x70

5.164.4.122 CSGCM1R

```
__IO uint32_t CRYP_TypeDef::CSGCM1R
```

CRYP GCM/GMAC context swap register 1, Address offset: 0x74

5.164.4.123 CSGCM2R

```
__IO uint32_t CRYP_TypeDef::CSGCM2R
```

CRYP GCM/GMAC context swap register 2, Address offset: 0x78

5.164.4.124 CSGCM3R

```
__IO uint32_t CRYP_TypeDef::CSGCM3R
```

CRYP GCM/GMAC context swap register 3, Address offset: 0x7C

5.164.4.125 CSGCM4R

```
__IO uint32_t CRYP_TypeDef::CSGCM4R
```

CRYP GCM/GMAC context swap register 4, Address offset: 0x80

5.164.4.126 CSGCM5R

```
__IO uint32_t CRYP_TypeDef::CSGCM5R
```

CRYP GCM/GMAC context swap register 5, Address offset: 0x84

5.164.4.127 CSGCM6R

```
__IO uint32_t CRYP_TypeDef::CSGCM6R
```

CRYP GCM/GMAC context swap register 6, Address offset: 0x88

5.164.4.128 CSGCM7R

```
__IO uint32_t CRYP_TypeDef::CSGCM7R
```

CRYP GCM/GMAC context swap register 7, Address offset: 0x8C

5.164.4.129 CSGCMCCM0R

```
__IO uint32_t CRYP_TypeDef::CSGCMCCM0R
```

CRYP GCM/GMAC or CCM/CMAC context swap register 0, Address offset: 0x50

5.164.4.130 CSGCMCCM1R

```
__IO uint32_t CRYP_TypeDef::CSGCMCCM1R
```

CRYP GCM/GMAC or CCM/CMAC context swap register 1, Address offset: 0x54

5.164.4.131 CSGCMCCM2R

```
__IO uint32_t CRYP_TypeDef::CSGCMCCM2R
```

CRYP GCM/GMAC or CCM/CMAC context swap register 2, Address offset: 0x58

5.164.4.132 CSGCMCCM3R

```
__IO uint32_t CRYP_TypeDef::CSGCMCCM3R
```

CRYP GCM/GMAC or CCM/CMAC context swap register 3, Address offset: 0x5C

5.164.4.133 CSGCMCCM4R

```
__IO uint32_t CRYP_TypeDef::CSGCMCCM4R
```

CRYP GCM/GMAC or CCM/CMAC context swap register 4, Address offset: 0x60

5.164.4.134 CSGCMCCM5R

```
__IO uint32_t CRYP_TypeDef::CSGCMCCM5R
```

CRYP GCM/GMAC or CCM/CMAC context swap register 5, Address offset: 0x64

5.164.4.135 CSGCMCCM6R

```
__IO uint32_t CRYP_TypeDef::CSGCMCCM6R
```

CRYP GCM/GMAC or CCM/CMAC context swap register 6, Address offset: 0x68

5.164.4.136 CSGCMCCM7R

```
__IO uint32_t CRYP_TypeDef::CSGCMCCM7R
```

CRYP GCM/GMAC or CCM/CMAC context swap register 7, Address offset: 0x6C

5.164.4.137 CSR [1/4]

```
__IO uint32_t ADC_Common_TypeDef::CSR
```

ADC Common status register, Address offset: ADC1 base address + 0x300

5.164.4.138 CSR [2/4]

```
__IO uint32_t PWR_TypeDef::CSR
```

PWR power control/status register, Address offset: 0x04

5.164.4.139 CSR [3/4]

```
__IO uint32_t RCC_TypeDef::CSR
```

RCC clock control & status register, Address offset: 0x74

5.164.4.140 CSR [4/4]

```
__IO uint32_t HASH_TypeDef::CSR[54]
```

HASH context swap registers, Address offset: 0x0F8-0x1CC

5.164.4.141 CWSIZER

```
__IO uint32_t DCMI_TypeDef::CWSIZER
```

DCMI crop window size, Address offset: 0x24

5.164.4.142 CWSTRTR

```
__IO uint32_t DCMI_TypeDef::CWSTRTR
```

DCMI crop window start, Address offset: 0x20

5.164.4.143 DCCR

```
__IO uint32_t LTDC_Layer_TypeDef::DCCR
```

LTDC Layerx Default Color Configuration Register Address offset: 0x9C

5.164.4.144 DCKCFGR

```
__IO uint32_t RCC_TypeDef::DCKCFGR
```

RCC Dedicated Clocks configuration register, Address offset: 0x8C

5.164.4.145 DCKCFGR2

```
__IO uint32_t RCC_TypeDef::DCKCFGR2
```

RCC Dedicated Clocks configuration register 2, Address offset: 0x94

5.164.4.146 DCOUNT

`__I uint32_t SDIO_TypeDef::DCOUNT`

SDIO data counter register, Address offset: 0x30

5.164.4.147 DCR

`__IO uint16_t TIM_TypeDef::DCR`

TIM DMA control register, Address offset: 0x48

5.164.4.148 DCTRL

`__IO uint32_t SDIO_TypeDef::DCTRL`

SDIO data control register, Address offset: 0x2C

5.164.4.149 DHR12L1

`__IO uint32_t DAC_TypeDef::DHR12L1`

DAC channel1 12-bit left aligned data holding register, Address offset: 0x0C

5.164.4.150 DHR12L2

`__IO uint32_t DAC_TypeDef::DHR12L2`

DAC channel2 12-bit left aligned data holding register, Address offset: 0x18

5.164.4.151 DHR12LD

`__IO uint32_t DAC_TypeDef::DHR12LD`

DUAL DAC 12-bit left aligned data holding register, Address offset: 0x24

5.164.4.152 DHR12R1

`__IO uint32_t DAC_TypeDef::DHR12R1`

DAC channel1 12-bit right-aligned data holding register, Address offset: 0x08

5.164.4.153 DHR12R2

`__IO uint32_t DAC_TypeDef::DHR12R2`

DAC channel2 12-bit right aligned data holding register, Address offset: 0x14

5.164.4.154 DHR12RD

```
__IO uint32_t DAC_TypeDef::DHR12RD
```

Dual DAC 12-bit right-aligned data holding register, Address offset: 0x20

5.164.4.155 DHR8R1

```
__IO uint32_t DAC_TypeDef::DHR8R1
```

DAC channel1 8-bit right aligned data holding register, Address offset: 0x10

5.164.4.156 DHR8R2

```
__IO uint32_t DAC_TypeDef::DHR8R2
```

DAC channel2 8-bit right-aligned data holding register, Address offset: 0x1C

5.164.4.157 DHR8RD

```
__IO uint32_t DAC_TypeDef::DHR8RD
```

DUAL DAC 8-bit right aligned data holding register, Address offset: 0x28

5.164.4.158 DIER

```
__IO uint16_t TIM_TypeDef::DIER
```

TIM DMA/interrupt enable register, Address offset: 0x0C

5.164.4.159 DIN

```
__IO uint32_t HASH_TypeDef::DIN
```

HASH data input register, Address offset: 0x04

5.164.4.160 DLEN

```
__IO uint32_t SDIO_TypeDef::DLEN
```

SDIO data length register, Address offset: 0x28

5.164.4.161 DMACR

`__IO uint32_t CRYP_TypeDef::DMACR`

CRYP DMA control register, Address offset: 0x10

5.164.4.162 DMAR

`__IO uint16_t TIM_TypeDef::DMAR`

TIM DMA address for full transfer, Address offset: 0x4C

5.164.4.163 DOR1

`__IO uint32_t DAC_TypeDef::DOR1`

DAC channel1 data output register, Address offset: 0x2C

5.164.4.164 DOR2

`__IO uint32_t DAC_TypeDef::DOR2`

DAC channel2 data output register, Address offset: 0x30

5.164.4.165 DOUT

`__IO uint32_t CRYP_TypeDef::DOUT`

CRYP data output register, Address offset: 0x0C

5.164.4.166 DR [1/10]

`__IO uint32_t ADC_TypeDef::DR`

ADC regular data register, Address offset: 0x4C

5.164.4.167 DR [2/10]

`__IO uint32_t CRC_TypeDef::DR`

CRC Data register, Address offset: 0x00

5.164.4.168 DR [3/10]

`__IO uint32_t DCMI_TypeDef::DR`

DCMI data register, Address offset: 0x28

5.164.4.169 DR [4/10]

```
__IO uint16_t I2C_TypeDef::DR
```

I2C Data register, Address offset: 0x10

5.164.4.170 DR [5/10]

```
__IO uint32_t RTC_TypeDef::DR
```

RTC date register, Address offset: 0x04

5.164.4.171 DR [6/10]

```
__IO uint32_t SAI_Block_TypeDef::DR
```

SAI block x data register, Address offset: 0x20

5.164.4.172 DR [7/10]

```
__IO uint16_t SPI_TypeDef::DR
```

SPI data register, Address offset: 0x0C

5.164.4.173 DR [8/10]

```
__IO uint16_t USART_TypeDef::DR
```

USART Data register, Address offset: 0x04

5.164.4.174 DR [9/10]

```
__IO uint32_t CRYP_TypeDef::DR
```

CRYP data input register, Address offset: 0x08

5.164.4.175 DR [10/10]

```
__IO uint32_t RNG_TypeDef::DR
```

RNG data register, Address offset: 0x08

5.164.4.176 DTIMER

```
__IO uint32_t SDIO_TypeDef::DTIMER
```

SDIO data timer register, Address offset: 0x24

5.164.4.177 EGR

`__IO uint16_t TIM_TypeDef::EGR`

TIM event generation register, Address offset: 0x14

5.164.4.178 EMR

`__IO uint32_t EXTI_TypeDef::EMR`

EXTI Event mask register, Address offset: 0x04

5.164.4.179 ESCR

`__IO uint32_t DCMI_TypeDef::ESCR`

DCMI embedded synchronization code register, Address offset: 0x18

5.164.4.180 ESR

`__IO uint32_t CAN_TypeDef::ESR`

CAN error status register, Address offset: 0x18

5.164.4.181 ESUR

`__IO uint32_t DCMI_TypeDef::ESUR`

DCMI embedded synchronization unmask register, Address offset: 0x1C

5.164.4.182 EXTICR

`__IO uint32_t SYSCFG_TypeDef::EXTICR[4]`

SYSCFG external interrupt configuration registers, Address offset: 0x08-0x14

5.164.4.183 FA1R

`__IO uint32_t CAN_TypeDef::FA1R`

CAN filter activation register, Address offset: 0x21C

5.164.4.184 FCR

```
__IO uint32_t DMA_Stream_TypeDef::FCR
```

DMA stream x FIFO control register

5.164.4.185 FFA1R

```
__IO uint32_t CAN_TypeDef::FFA1R
```

CAN filter FIFO assignment register, Address offset: 0x214

5.164.4.186 FGCLUT

```
__IO uint32_t DMA2D_TypeDef::FGCLUT[256]
```

DMA2D Foreground CLUT, Address offset: 400-7FF

5.164.4.187 FGCMAR

```
__IO uint32_t DMA2D_TypeDef::FGCMAR
```

DMA2D Foreground CLUT Memory Address Register, Address offset: 0x2C

5.164.4.188 FGCOLR

```
__IO uint32_t DMA2D_TypeDef::FGCOLR
```

DMA2D Foreground Color Register, Address offset: 0x20

5.164.4.189 FGMAR

```
__IO uint32_t DMA2D_TypeDef::FGMAR
```

DMA2D Foreground Memory Address Register, Address offset: 0x0C

5.164.4.190 FGOR

```
__IO uint32_t DMA2D_TypeDef::FGOR
```

DMA2D Foreground Offset Register, Address offset: 0x10

5.164.4.191 FGPFCCR

```
__IO uint32_t DMA2D_TypeDef::FGPFCCR
```

DMA2D Foreground PFC Control Register, Address offset: 0x1C

5.164.4.192 FIFO

```
__IO uint32_t SDIO_TypeDef::FIFO
```

SDIO data FIFO register, Address offset: 0x80

5.164.4.193 FIFOCNT

```
__I uint32_t SDIO_TypeDef::FIFOCNT
```

SDIO FIFO counter register, Address offset: 0x48

5.164.4.194 FLTR

```
__IO uint16_t I2C_TypeDef::FLTR
```

I2C FLTR register, Address offset: 0x24

5.164.4.195 FM1R

```
__IO uint32_t CAN_TypeDef::FM1R
```

CAN filter mode register, Address offset: 0x204

5.164.4.196 FMR

```
__IO uint32_t CAN_TypeDef::FMR
```

CAN filter master register, Address offset: 0x200

5.164.4.197 FR1

```
__IO uint32_t CAN_FilterRegister_TypeDef::FR1
```

CAN Filter bank register 1

5.164.4.198 FR2

```
__IO uint32_t CAN_FilterRegister_TypeDef::FR2
```

CAN Filter bank register 1

5.164.4.199 FRCR

```
__IO uint32_t SAI_Block_TypeDef::FRCR
```

SAI block x frame configuration register, Address offset: 0x0C

5.164.4.200 FS1R

```
__IO uint32_t CAN_TypeDef::FS1R
```

CAN filter scale register, Address offset: 0x20C

5.164.4.201 FTSR

```
__IO uint32_t EXTI_TypeDef::FTSR
```

EXTI Falling trigger selection register, Address offset: 0x0C

5.164.4.202 GCR [1/2]

```
__IO uint32_t LTDC_TypeDef::GCR
```

LTDC Global Control Register, Address offset: 0x18

5.164.4.203 GCR [2/2]

```
__IO uint32_t SAI_TypeDef::GCR
```

SAI global configuration register, Address offset: 0x00

5.164.4.204 GTPR

```
__IO uint16_t USART_TypeDef::GTPR
```

USART Guard time and prescaler register, Address offset: 0x18

5.164.4.205 HIFCR

```
__IO uint32_t DMA_TypeDef::HIFCR
```

DMA high interrupt flag clear register, Address offset: 0x0C

5.164.4.206 HISR

```
__IO uint32_t DMA_TypeDef::HISR
```

DMA high interrupt status register, Address offset: 0x04

5.164.4.207 HR [1/2]

```
__IO uint32_t HASH_TypeDef::HR[5]
```

HASH digest registers, Address offset: 0x0C-0x1C

5.164.4.208 HR [2/2]

```
__IO uint32_t HASH_DIGEST_TypeDef::HR[8]
```

HASH digest registers, Address offset: 0x310-0x32C

5.164.4.209 HTR

```
__IO uint32_t ADC_TypeDef::HTR
```

ADC watchdog higher threshold register, Address offset: 0x24

5.164.4.210 I2SCFGR

```
__IO uint16_t SPI_TypeDef::I2SCFGR
```

SPI_I2S configuration register, Address offset: 0x1C

5.164.4.211 I2SPR

```
__IO uint16_t SPI_TypeDef::I2SPR
```

SPI_I2S prescaler register, Address offset: 0x20

5.164.4.212 ICR [1/3]

```
__IO uint32_t DCMI_TypeDef::ICR
```

DCMI interrupt clear register, Address offset: 0x14

5.164.4.213 ICR [2/3]

```
__IO uint32_t LTDC_TypeDef::ICR
```

LTDC Interrupt Clear Register, Address offset: 0x3C

5.164.4.214 ICR [3/3]

```
__IO uint32_t SDIO_TypeDef::ICR
```

SDIO interrupt clear register, Address offset: 0x38

5.164.4.215 IDCODE

```
__IO uint32_t DBGMCU_TypeDef::IDCODE
```

MCU device ID code, Address offset: 0x00

5.164.4.216 IDR [1/2]

```
__IO uint8_t CRC_TypeDef::IDR
```

CRC Independent data register, Address offset: 0x04

5.164.4.217 IDR [2/2]

```
__IO uint32_t GPIO_TypeDef::IDR
```

GPIO port input data register, Address offset: 0x10

5.164.4.218 IER [1/3]

```
__IO uint32_t CAN_TypeDef::IER
```

CAN interrupt enable register, Address offset: 0x14

5.164.4.219 IER [2/3]

```
__IO uint32_t DCMI_TypeDef::IER
```

DCMI interrupt enable register, Address offset: 0x0C

5.164.4.220 IER [3/3]

```
__IO uint32_t LTDC_TypeDef::IER
```

LTDC Interrupt Enable Register, Address offset: 0x34

5.164.4.221 IFCR

```
__IO uint32_t DMA2D_TypeDef::IFCR
```

DMA2D Interrupt Flag Clear Register, Address offset: 0x08

5.164.4.222 IMR [1/3]

```
__IO uint32_t EXTI_TypeDef::IMR
```

EXTI Interrupt mask register, Address offset: 0x00

5.164.4.223 IMR [2/3]

```
__IO uint32_t SAI_Block_TypeDef::IMR
```

SAI block x interrupt mask register, Address offset: 0x14

5.164.4.224 IMR [3/3]

```
__IO uint32_t HASH_TypeDef::IMR
```

HASH interrupt enable register, Address offset: 0x20

5.164.4.225 IMSCR

```
__IO uint32_t CRYP_TypeDef::IMSCR
```

CRYP interrupt mask set/clear register, Address offset: 0x14

5.164.4.226 ISR [1/3]

```
__IO uint32_t DMA2D_TypeDef::ISR
```

DMA2D Interrupt Status Register, Address offset: 0x04

5.164.4.227 ISR [2/3]

```
__IO uint32_t LTDC_TypeDef::ISR
```

LTDC Interrupt Status Register, Address offset: 0x38

5.164.4.228 ISR [3/3]

```
__IO uint32_t RTC_TypeDef::ISR
```

RTC initialization and status register, Address offset: 0x0C

5.164.4.229 ITM_RxBuffer

```
volatile int32_t ITM_RxBuffer [extern]
```

External variable to receive characters.

5.164.4.230 IV0LR

```
__IO uint32_t CRYP_TypeDef::IV0LR
```

CRYP initialization vector left-word register 0, Address offset: 0x40

5.164.4.231 IV0RR

```
__IO uint32_t CRYP_TypeDef::IV0RR
```

CRYP initialization vector right-word register 0, Address offset: 0x44

5.164.4.232 IV1LR

```
__IO uint32_t CRYP_TypeDef::IV1LR
```

CRYP initialization vector left-word register 1, Address offset: 0x48

5.164.4.233 IV1RR

`__IO uint32_t CRYP_TypeDef::IV1RR`

CRYP initialization vector right-word register 1, Address offset: 0x4C

5.164.4.234 JDR1

`__IO uint32_t ADC_TypeDef::JDR1`

ADC injected data register 1, Address offset: 0x3C

5.164.4.235 JDR2

`__IO uint32_t ADC_TypeDef::JDR2`

ADC injected data register 2, Address offset: 0x40

5.164.4.236 JDR3

`__IO uint32_t ADC_TypeDef::JDR3`

ADC injected data register 3, Address offset: 0x44

5.164.4.237 JDR4

`__IO uint32_t ADC_TypeDef::JDR4`

ADC injected data register 4, Address offset: 0x48

5.164.4.238 JOFR1

`__IO uint32_t ADC_TypeDef::JOFR1`

ADC injected channel data offset register 1, Address offset: 0x14

5.164.4.239 JOFR2

`__IO uint32_t ADC_TypeDef::JOFR2`

ADC injected channel data offset register 2, Address offset: 0x18

5.164.4.240 JOFR3

`__IO uint32_t ADC_TypeDef::JOFR3`

ADC injected channel data offset register 3, Address offset: 0x1C

5.164.4.241 JOFR4

```
__IO uint32_t ADC_TypeDef::JOFR4
```

ADC injected channel data offset register 4, Address offset: 0x20

5.164.4.242 JSQR

```
__IO uint32_t ADC_TypeDef::JSQR
```

ADC injected sequence register, Address offset: 0x38

5.164.4.243 K0LR

```
__IO uint32_t CRYP_TypeDef::K0LR
```

CRYP key left register 0, Address offset: 0x20

5.164.4.244 K0RR

```
__IO uint32_t CRYP_TypeDef::K0RR
```

CRYP key right register 0, Address offset: 0x24

5.164.4.245 K1LR

```
__IO uint32_t CRYP_TypeDef::K1LR
```

CRYP key left register 1, Address offset: 0x28

5.164.4.246 K1RR

```
__IO uint32_t CRYP_TypeDef::K1RR
```

CRYP key right register 1, Address offset: 0x2C

5.164.4.247 K2LR

```
__IO uint32_t CRYP_TypeDef::K2LR
```

CRYP key left register 2, Address offset: 0x30

5.164.4.248 K2RR

```
__IO uint32_t CRYP_TypeDef::K2RR
```

CRYP key right register 2, Address offset: 0x34

5.164.4.249 K3LR

```
__IO uint32_t CRYP_TypeDef::K3LR
```

CRYP key left register 3, Address offset: 0x38

5.164.4.250 K3RR

```
__IO uint32_t CRYP_TypeDef::K3RR
```

CRYP key right register 3, Address offset: 0x3C

5.164.4.251 KEYR

```
__IO uint32_t FLASH_TypeDef::KEYR
```

FLASH key register, Address offset: 0x04

5.164.4.252 KR

```
__IO uint32_t IWDG_TypeDef::KR
```

IWDG Key register, Address offset: 0x00

5.164.4.253 LCKR

```
__IO uint32_t GPIO_TypeDef::LCKR
```

GPIO port configuration lock register, Address offset: 0x1C

5.164.4.254 LIFCR

```
__IO uint32_t DMA_TypeDef::LIFCR
```

DMA low interrupt flag clear register, Address offset: 0x08

5.164.4.255 LIPCR

```
__IO uint32_t LTDC_TypeDef::LIPCR
```

LTDC Line Interrupt Position Configuration Register, Address offset: 0x40

5.164.4.256 LISR

```
__IO uint32_t DMA_TypeDef::LISR
```

DMA low interrupt status register, Address offset: 0x00

5.164.4.257 LTR

```
__IO uint32_t ADC_TypeDef::LTR
```

ADC watchdog lower threshold register, Address offset: 0x28

5.164.4.258 LWR

```
__IO uint32_t DMA2D_TypeDef::LWR
```

DMA2D Line Watermark Register, Address offset: 0x48

5.164.4.259 M0AR

```
__IO uint32_t DMA_Stream_TypeDef::M0AR
```

DMA stream x memory 0 address register

5.164.4.260 M1AR

```
__IO uint32_t DMA_Stream_TypeDef::M1AR
```

DMA stream x memory 1 address register

5.164.4.261 MASK

```
__IO uint32_t SDIO_TypeDef::MASK
```

SDIO mask register, Address offset: 0x3C

5.164.4.262 MCR

```
__IO uint32_t CAN_TypeDef::MCR
```

CAN master control register, Address offset: 0x00

5.164.4.263 MEMRMP

`__IO uint32_t SYSCFG_TypeDef::MEMRMP`

SYSCFG memory remap register, Address offset: 0x00

5.164.4.264 MISR [1/2]

`__IO uint32_t DCMI_TypeDef::MISR`

DCMI masked interrupt status register, Address offset: 0x10

5.164.4.265 MISR [2/2]

`__IO uint32_t CRYP_TypeDef::MISR`

CRYP masked interrupt status register, Address offset: 0x1C

5.164.4.266 MODER

`__IO uint32_t GPIO_TypeDef::MODER`

GPIO port mode register, Address offset: 0x00

5.164.4.267 MSR

`__IO uint32_t CAN_TypeDef::MSR`

CAN master status register, Address offset: 0x04

5.164.4.268 NDTR

`__IO uint32_t DMA_Stream_TypeDef::NDTR`

DMA stream x number of data register

5.164.4.269 NLR

`__IO uint32_t DMA2D_TypeDef::NLR`

DMA2D Number of Line Register, Address offset: 0x44

5.164.4.270 OAR1

```
__IO uint16_t I2C_TypeDef::OAR1
```

I2C Own address register 1, Address offset: 0x08

5.164.4.271 OAR2

```
__IO uint16_t I2C_TypeDef::OAR2
```

I2C Own address register 2, Address offset: 0x0C

5.164.4.272 OCOLR

```
__IO uint32_t DMA2D_TypeDef::OCOLR
```

DMA2D Output Color Register, Address offset: 0x38

5.164.4.273 ODR

```
__IO uint32_t GPIO_TypeDef::ODR
```

GPIO port output data register, Address offset: 0x14

5.164.4.274 OMAR

```
__IO uint32_t DMA2D_TypeDef::OMAR
```

DMA2D Output Memory Address Register, Address offset: 0x3C

5.164.4.275 OOR

```
__IO uint32_t DMA2D_TypeDef::OOR
```

DMA2D Output Offset Register, Address offset: 0x40

5.164.4.276 OPFCCR

```
__IO uint32_t DMA2D_TypeDef::OPFCCR
```

DMA2D Output PFC Control Register, Address offset: 0x34

5.164.4.277 OPTCR

```
__IO uint32_t FLASH_TypeDef::OPTCR
```

FLASH option control register , Address offset: 0x14

5.164.4.278 OPTCR1

```
__IO uint32_t FLASH_TypeDef::OPTCR1
```

FLASH option control register 1, Address offset: 0x18

5.164.4.279 OPTKEYR

```
__IO uint32_t FLASH_TypeDef::OPTKEYR
```

FLASH option key register, Address offset: 0x08

5.164.4.280 OR

```
__IO uint16_t TIM_TypeDef::OR
```

TIM option register, Address offset: 0x50

5.164.4.281 OSPEEDR

```
__IO uint32_t GPIO_TypeDef::OSPEEDR
```

GPIO port output speed register, Address offset: 0x08

5.164.4.282 OTYPER

```
__IO uint32_t GPIO_TypeDef::OTYPER
```

GPIO port output type register, Address offset: 0x04

5.164.4.283 PAR

```
__IO uint32_t DMA_Stream_TypeDef::PAR
```

DMA stream x peripheral address register

5.164.4.284 PFCR

```
__IO uint32_t LTDC_Layer_TypeDef::PFCR
```

LTDC Layerx Pixel Format Configuration Register Address offset: 0x94

5.164.4.285 PLLCFGR

```
__IO uint32_t RCC_TypeDef::PLLCFGR
```

RCC PLL configuration register, Address offset: 0x04

5.164.4.286 PLLI2SCFGR

```
__IO uint32_t RCC_TypeDef::PLLI2SCFGR
```

RCC PLLI2S configuration register, Address offset: 0x84

5.164.4.287 PLLSAICFGR

```
__IO uint32_t RCC_TypeDef::PLLSAICFGR
```

RCC PLLSAI configuration register, Address offset: 0x88

5.164.4.288 PMC

```
__IO uint32_t SYSCFG_TypeDef::PMC
```

SYSCFG peripheral mode configuration register, Address offset: 0x04

5.164.4.289 POWER

```
__IO uint32_t SDIO_TypeDef::POWER
```

SDIO power control register, Address offset: 0x00

5.164.4.290 PR [1/2]

```
__IO uint32_t EXTI_TypeDef::PR
```

EXTI Pending register, Address offset: 0x14

5.164.4.291 PR [2/2]

```
__IO uint32_t IWDG_TypeDef::PR
```

IWDG Prescaler register, Address offset: 0x04

5.164.4.292 PRER

```
__IO uint32_t RTC_TypeDef::PRER
```

RTC prescaler register, Address offset: 0x10

5.164.4.293 PSC

```
__IO uint16_t TIM_TypeDef::PSC
```

TIM prescaler, Address offset: 0x28

5.164.4.294 PUPDR

```
__IO uint32_t GPIO_TypeDef::PUPDR
```

GPIO port pull-up/pull-down register, Address offset: 0x0C

5.164.4.295 RCR

```
__IO uint16_t TIM_TypeDef::RCR
```

TIM repetition counter register, Address offset: 0x30

5.164.4.296 RDHR

```
__IO uint32_t CAN_FIFOMailBox_TypeDef::RDHR
```

CAN receive FIFO mailbox data high register

5.164.4.297 RDLR

```
__IO uint32_t CAN_FIFOMailBox_TypeDef::RDLR
```

CAN receive FIFO mailbox data low register

5.164.4.298 RDTR

```
__IO uint32_t CAN_FIFOMailBox_TypeDef::RDTR
```

CAN receive FIFO mailbox data length control and time stamp register

5.164.4.299 RESERVED [1/3]

```
uint32_t DMA2D_TypeDef::RESERVED[236]
```

Reserved, 0x50-0x3FF

5.164.4.300 RESERVED [2/3]

```
uint32_t SYSCFG_TypeDef::RESERVED[2]
```

Reserved, 0x18-0x1C

5.164.4.301 RESERVED [3/3]

```
uint32_t HASH_TypeDef::RESERVED[52]
```

Reserved, 0x28-0xF4

5.164.4.302 RESERVED0 [1/10]

```
uint16_t I2C_TypeDef::RESERVED0
```

Reserved, 0x02

5.164.4.303 RESERVED0 [2/10]

```
uint32_t LTDC_TypeDef::RESERVED0[2]
```

Reserved, 0x00-0x04

5.164.4.304 RESERVED0 [3/10]

```
uint32_t LTDC_Layer_TypeDef::RESERVED0[2]
```

Reserved

5.164.4.305 RESERVED0 [4/10]

```
uint32_t SDIO_TypeDef::RESERVED0[2]
```

Reserved, 0x40-0x44

5.164.4.306 RESERVED0 [5/10]

```
uint32_t CAN_TypeDef::RESERVED0[88]
```

Reserved, 0x020 - 0x17F

5.164.4.307 RESERVED0 [6/10]

```
uint8_t CRC_TypeDef::RESERVED0
```

Reserved, 0x05

5.164.4.308 RESERVED0 [7/10]

```
uint32_t RCC_TypeDef::RESERVED0
```

Reserved, 0x1C

5.164.4.309 RESERVED0 [8/10]

```
uint16_t SPI_TypeDef::RESERVED0
```

Reserved, 0x02

5.164.4.310 RESERVED0 [9/10]

```
uint16_t TIM_TypeDef::RESERVED0
```

Reserved, 0x02

5.164.4.311 RESERVED0 [10/10]

```
uint16_t USART_TypeDef::RESERVED0
```

Reserved, 0x02

5.164.4.312 RESERVED1 [1/10]

```
uint32_t CAN_TypeDef::RESERVED1[12]
```

Reserved, 0x1D0 - 0x1FF

5.164.4.313 RESERVED1 [2/10]

```
uint16_t CRC_TypeDef::RESERVED1
```

Reserved, 0x06

5.164.4.314 RESERVED1 [3/10]

```
uint32_t SDIO_TypeDef::RESERVED1[13]
```

Reserved, 0x4C-0x7C

5.164.4.315 RESERVED1 [4/10]

```
uint16_t I2C_TypeDef::RESERVED1
```

Reserved, 0x06

5.164.4.316 RESERVED1 [5/10]

```
uint32_t LTDC_TypeDef::RESERVED1[2]
```

Reserved, 0x1C-0x20

5.164.4.317 RESERVED1 [6/10]

```
uint32_t RCC_TypeDef::RESERVED1[2]
```

Reserved, 0x28-0x2C

5.164.4.318 RESERVED1 [7/10]

```
uint32_t LTDC_Layer_TypeDef::RESERVED1[3]
```

Reserved

5.164.4.319 RESERVED1 [8/10]

```
uint16_t SPI_TypeDef::RESERVED1
```

Reserved, 0x06

5.164.4.320 RESERVED1 [9/10]

```
uint16_t TIM_TypeDef::RESERVED1
```

Reserved, 0x06

5.164.4.321 RESERVED1 [10/10]

```
uint16_t USART_TypeDef::RESERVED1
```

Reserved, 0x06

5.164.4.322 RESERVED10

```
uint16_t TIM_TypeDef::RESERVED10
```

Reserved, 0x32

5.164.4.323 RESERVED11

```
uint16_t TIM_TypeDef::RESERVED11
```

Reserved, 0x46

5.164.4.324 RESERVED12

```
uint16_t TIM_TypeDef::RESERVED12
```

Reserved, 0x4A

5.164.4.325 RESERVED13

```
uint16_t TIM_TypeDef::RESERVED13
```

Reserved, 0x4E

5.164.4.326 RESERVED14

```
uint16_t TIM_TypeDef::RESERVED14
```

Reserved, 0x52

5.164.4.327 RESERVED2 [1/7]

```
uint32_t CAN_TypeDef::RESERVED2
```

Reserved, 0x208

5.164.4.328 RESERVED2 [2/7]

```
uint32_t LTDC_TypeDef::RESERVED2[1]
```

Reserved, 0x28

5.164.4.329 RESERVED2 [3/7]

```
uint16_t I2C_TypeDef::RESERVED2
```

Reserved, 0x0A

5.164.4.330 RESERVED2 [4/7]

```
uint32_t RCC_TypeDef::RESERVED2
```

Reserved, 0x3C

5.164.4.331 RESERVED2 [5/7]

```
uint16_t SPI_TypeDef::RESERVED2
```

Reserved, 0x0A

5.164.4.332 RESERVED2 [6/7]

```
uint16_t TIM_TypeDef::RESERVED2
```

Reserved, 0x0A

5.164.4.333 RESERVED2 [7/7]

```
uint16_t USART_TypeDef::RESERVED2
```

Reserved, 0x0A

5.164.4.334 RESERVED3 [1/7]

```
uint32_t CAN_TypeDef::RESERVED3
```

Reserved, 0x210

5.164.4.335 RESERVED3 [2/7]

```
uint16_t I2C_TypeDef::RESERVED3
```

Reserved, 0x0E

5.164.4.336 RESERVED3 [3/7]

```
uint32_t LTDC_TypeDef::RESERVED3[1]
```

Reserved, 0x30

5.164.4.337 RESERVED3 [4/7]

```
uint32_t RCC_TypeDef::RESERVED3[2]
```

Reserved, 0x48-0x4C

5.164.4.338 RESERVED3 [5/7]

```
uint16_t SPI_TypeDef::RESERVED3
```

Reserved, 0x0E

5.164.4.339 RESERVED3 [6/7]

```
uint16_t TIM_TypeDef::RESERVED3
```

Reserved, 0x0E

5.164.4.340 RESERVED3 [7/7]

```
uint16_t USART_TypeDef::RESERVED3
```

Reserved, 0x0E

5.164.4.341 RESERVED4 [1/6]

```
uint32_t CAN_TypeDef::RESERVED4
```

Reserved, 0x218

5.164.4.342 RESERVED4 [2/6]

```
uint16_t I2C_TypeDef::RESERVED4
```

Reserved, 0x12

5.164.4.343 RESERVED4 [3/6]

```
uint32_t RCC_TypeDef::RESERVED4
```

Reserved, 0x5C

5.164.4.344 RESERVED4 [4/6]

```
uint16_t SPI_TypeDef::RESERVED4
```

Reserved, 0x12

5.164.4.345 RESERVED4 [5/6]

```
uint16_t TIM_TypeDef::RESERVED4
```

Reserved, 0x12

5.164.4.346 RESERVED4 [6/6]

```
uint16_t USART_TypeDef::RESERVED4
```

Reserved, 0x12

5.164.4.347 RESERVED5 [1/6]

```
uint32_t CAN_TypeDef::RESERVED5[8]
```

Reserved, 0x220-0x23F

5.164.4.348 RESERVED5 [2/6]

```
uint16_t I2C_TypeDef::RESERVED5
```

Reserved, 0x16

5.164.4.349 RESERVED5 [3/6]

```
uint32_t RCC_TypeDef::RESERVED5[2]
```

Reserved, 0x68-0x6C

5.164.4.350 RESERVED5 [4/6]

```
uint16_t SPI_TypeDef::RESERVED5
```

Reserved, 0x16

5.164.4.351 RESERVED5 [5/6]

```
uint16_t TIM_TypeDef::RESERVED5
```

Reserved, 0x16

5.164.4.352 RESERVED5 [6/6]

```
uint16_t USART_TypeDef::RESERVED5
```

Reserved, 0x16

5.164.4.353 RESERVED6 [1/5]

```
uint16_t I2C_TypeDef::RESERVED6
```

Reserved, 0x1A

5.164.4.354 RESERVED6 [2/5]

```
uint32_t RCC_TypeDef::RESERVED6[2]
```

Reserved, 0x78-0x7C

5.164.4.355 RESERVED6 [3/5]

```
uint16_t SPI_TypeDef::RESERVED6
```

Reserved, 0x1A

5.164.4.356 RESERVED6 [4/5]

```
uint16_t TIM_TypeDef::RESERVED6
```

Reserved, 0x1A

5.164.4.357 RESERVED6 [5/5]

```
uint16_t USART_TypeDef::RESERVED6
```

Reserved, 0x1A

5.164.4.358 RESERVED7 [1/4]

```
uint16_t I2C_TypeDef::RESERVED7
```

Reserved, 0x1E

5.164.4.359 RESERVED7 [2/4]

```
uint32_t RTC_TypeDef::RESERVED7
```

Reserved, 0x4C

5.164.4.360 RESERVED7 [3/4]

```
uint16_t SPI_TypeDef::RESERVED7
```

Reserved, 0x1E

5.164.4.361 RESERVED7 [4/4]

```
uint16_t TIM_TypeDef::RESERVED7
```

Reserved, 0x1E

5.164.4.362 RESERVED8 [1/3]

```
uint16_t I2C_TypeDef::RESERVED8
```

Reserved, 0x22

5.164.4.363 RESERVED8 [2/3]

```
uint16_t SPI_TypeDef::RESERVED8
```

Reserved, 0x22

5.164.4.364 RESERVED8 [3/3]

```
uint16_t TIM_TypeDef::RESERVED8
```

Reserved, 0x22

5.164.4.365 RESERVED9 [1/2]

```
uint16_t I2C_TypeDef::RESERVED9
```

Reserved, 0x26

5.164.4.366 RESERVED9 [2/2]

```
uint16_t TIM_TypeDef::RESERVED9
```

Reserved, 0x2A

5.164.4.367 RESP1

I uint32_t SDIO_TypeDef::RESP1

SDIO response 1 register, Address offset: 0x14

5.164.4.368 RESP2

```
__I uint32_t SDIO_TypeDef::RESP2
```

SDIO response 2 register, Address offset: 0x18

5.164.4.369 RESP3

```
__I uint32_t SDIO_TypeDef::RESP3
```

SDIO response 3 register, Address offset: 0x1C

5.164.4.370 RESP4

```
__I uint32_t SDIO_TypeDef::RESP4
```

SDIO response 4 register, Address offset: 0x20

5.164.4.371 RESPCMD

```
__I uint32_t SDIO_TypeDef::RESPCMD
```

SDIO command response register, Address offset: 0x10

5.164.4.372 RF0R

```
__IO uint32_t CAN_TypeDef::RF0R
```

CAN receive FIFO 0 register, Address offset: 0x0C

5.164.4.373 RF1R

```
__IO uint32_t CAN_TypeDef::RF1R
```

CAN receive FIFO 1 register, Address offset: 0x10

5.164.4.374 RIR

```
__IO uint32_t CAN_FIFOMailBox_TypeDef::RIR
```

CAN receive FIFO mailbox identifier register

5.164.4.375 RISR [1/2]

```
__IO uint32_t DCMI_TypeDef::RISR
```

DCMI raw interrupt status register, Address offset: 0x08

5.164.4.376 RISR [2/2]

```
__IO uint32_t CRYP_TypeDef::RISR
```

CRYP raw interrupt status register, Address offset: 0x18

5.164.4.377 RLR

```
__IO uint32_t IWDG_TypeDef::RLR
```

IWDG Reload register, Address offset: 0x08

5.164.4.378 RTSR

```
__IO uint32_t EXTI_TypeDef::RTSR
```

EXTI Rising trigger selection register, Address offset: 0x08

5.164.4.379 RXCRCR

```
__IO uint16_t SPI_TypeDef::RXCRCR
```

SPI RX CRC register (not used in I2S mode), Address offset: 0x14

5.164.4.380 sFIFOMailBox

```
CAN_FIFOMailBox_TypeDef CAN_TypeDef::sFIFOMailBox[2]
```

CAN FIFO MailBox, Address offset: 0x1B0 - 0x1CC

5.164.4.381 sFilterRegister

```
CAN_FilterRegister_TypeDef CAN_TypeDef::sFilterRegister[28]
```

CAN Filter Register, Address offset: 0x240-0x31C

5.164.4.382 SHIFTR

```
__IO uint32_t RTC_TypeDef::SHIFTR
```

RTC shift control register, Address offset: 0x2C

5.164.4.383 SLOTR

```
__IO uint32_t SAI_Block_TypeDef::SLOTR
```

SAI block x slot register, Address offset: 0x10

5.164.4.384 SMCR

```
__IO uint16_t TIM_TypeDef::SMCR
```

TIM slave mode control register, Address offset: 0x08

5.164.4.385 SMPR1

```
__IO uint32_t ADC_TypeDef::SMPR1
```

ADC sample time register 1, Address offset: 0x0C

5.164.4.386 SMPR2

```
__IO uint32_t ADC_TypeDef::SMPR2
```

ADC sample time register 2, Address offset: 0x10

5.164.4.387 SQR1

```
__IO uint32_t ADC_TypeDef::SQR1
```

ADC regular sequence register 1, Address offset: 0x2C

5.164.4.388 SQR2

```
__IO uint32_t ADC_TypeDef::SQR2
```

ADC regular sequence register 2, Address offset: 0x30

5.164.4.389 SQR3

```
__IO uint32_t ADC_TypeDef::SQR3
```

ADC regular sequence register 3, Address offset: 0x34

5.164.4.390 SR [1/13]

```
__IO uint32_t ADC_TypeDef::SR
```

ADC status register, Address offset: 0x00

5.164.4.391 SR [2/13]

```
__IO uint32_t DAC_TypeDef::SR
```

DAC status register, Address offset: 0x34

5.164.4.392 SR [3/13]

```
__IO uint32_t DCMI_TypeDef::SR
```

DCMI status register, Address offset: 0x04

5.164.4.393 SR [4/13]

```
__IO uint32_t FLASH_TypeDef::SR
```

FLASH status register, Address offset: 0x0C

5.164.4.394 SR [5/13]

```
__IO uint32_t IWDG_TypeDef::SR
```

IWDG Status register, Address offset: 0x0C

5.164.4.395 SR [6/13]

```
__IO uint32_t SAI_Block_TypeDef::SR
```

SAI block x status register, Address offset: 0x18

5.164.4.396 SR [7/13]

```
__IO uint16_t SPI_TypeDef::SR
```

SPI status register, Address offset: 0x08

5.164.4.397 SR [8/13]

```
__IO uint16_t TIM_TypeDef::SR
```

TIM status register, Address offset: 0x10

5.164.4.398 SR [9/13]

```
__IO uint16_t USART_TypeDef::SR
```

USART Status register, Address offset: 0x00

5.164.4.399 SR [10/13]

```
__IO uint32_t WWDG_TypeDef::SR
```

WWDG Status register, Address offset: 0x08

5.164.4.400 SR [11/13]

```
__IO uint32_t CRYP_TypeDef::SR
```

CRYP status register, Address offset: 0x04

5.164.4.401 SR [12/13]

```
__IO uint32_t HASH_TypeDef::SR
```

HASH status register, Address offset: 0x24

5.164.4.402 SR [13/13]

```
__IO uint32_t RNG_TypeDef::SR
```

RNG status register, Address offset: 0x04

5.164.4.403 SR1

```
__IO uint16_t I2C_TypeDef::SR1
```

I2C Status register 1, Address offset: 0x14

5.164.4.404 SR2

```
__IO uint16_t I2C_TypeDef::SR2
```

I2C Status register 2, Address offset: 0x18

5.164.4.405 SRCR

`__IO uint32_t LTDC_TypeDef::SRCR`

LTDC Shadow Reload Configuration Register, Address offset: 0x24

5.164.4.406 SSCGR

`__IO uint32_t RCC_TypeDef::SSCGR`

RCC spread spectrum clock generation register, Address offset: 0x80

5.164.4.407 SSCR

`__IO uint32_t LTDC_TypeDef::SSCR`

LTDC Synchronization Size Configuration Register, Address offset: 0x08

5.164.4.408 SSR

`__IO uint32_t RTC_TypeDef::SSR`

RTC sub second register, Address offset: 0x28

5.164.4.409 STA

`__I uint32_t SDIO_TypeDef::STA`

SDIO status register, Address offset: 0x34

5.164.4.410 STR

`__IO uint32_t HASH_TypeDef::STR`

HASH start register, Address offset: 0x08

5.164.4.411 sTxMailBox

`CAN_TxMailBox_TypeDef CAN_TypeDef::sTxMailBox[3]`

CAN Tx MailBox, Address offset: 0x180 - 0x1AC

5.164.4.412 SWIER

```
__IO uint32_t EXTI_TypeDef::SWIER
```

EXTI Software interrupt event register, Address offset: 0x10

5.164.4.413 SWTRIGR

```
__IO uint32_t DAC_TypeDef::SWTRIGR
```

DAC software trigger register, Address offset: 0x04

5.164.4.414 TAFCR

```
__IO uint32_t RTC_TypeDef::TAFCR
```

RTC tamper and alternate function configuration register, Address offset: 0x40

5.164.4.415 TDHR

```
__IO uint32_t CAN_TxMailBox_TypeDef::TDHR
```

CAN mailbox data high register

5.164.4.416 TDLR

```
__IO uint32_t CAN_TxMailBox_TypeDef::TDLR
```

CAN mailbox data low register

5.164.4.417 TDTR

```
__IO uint32_t CAN_TxMailBox_TypeDef::TDTR
```

CAN mailbox data length control and time stamp register

5.164.4.418 TIR

```
__IO uint32_t CAN_TxMailBox_TypeDef::TIR
```

CAN TX mailbox identifier register

5.164.4.419 TR

```
__IO uint32_t RTC_TypeDef::TR
```

RTC time register, Address offset: 0x00

5.164.4.420 TRISE

`__IO uint16_t I2C_TypeDef::TRISE`

I2C TRISE register, Address offset: 0x20

5.164.4.421 TSDR

`__IO uint32_t RTC_TypeDef::TSDR`

RTC time stamp date register, Address offset: 0x34

5.164.4.422 TSR

`__IO uint32_t CAN_TypeDef::TSR`

CAN transmit status register, Address offset: 0x08

5.164.4.423 TSSSR

`__IO uint32_t RTC_TypeDef::TSSSR`

RTC time-stamp sub second register, Address offset: 0x38

5.164.4.424 TSTR

`__IO uint32_t RTC_TypeDef::TSTR`

RTC time stamp time register, Address offset: 0x30

5.164.4.425 TWCR

`__IO uint32_t LTDC_TypeDef::TWCR`

LTDC Total Width Configuration Register, Address offset: 0x14

5.164.4.426 TXCRCR

`__IO uint16_t SPI_TypeDef::TXCRCR`

SPI TX CRC register (not used in I2S mode), Address offset: 0x18

5.164.4.427 WHPCR

```
__IO uint32_t LTDC_Layer_TypeDef::WHPCR
```

LTDC Layerx Window Horizontal Position Configuration Register Address offset: 0x88

5.164.4.428 WPR

```
__IO uint32_t RTC_TypeDef::WPR
```

RTC write protection register, Address offset: 0x24

5.164.4.429 WUTR

```
__IO uint32_t RTC_TypeDef::WUTR
```

RTC wakeup timer register, Address offset: 0x14

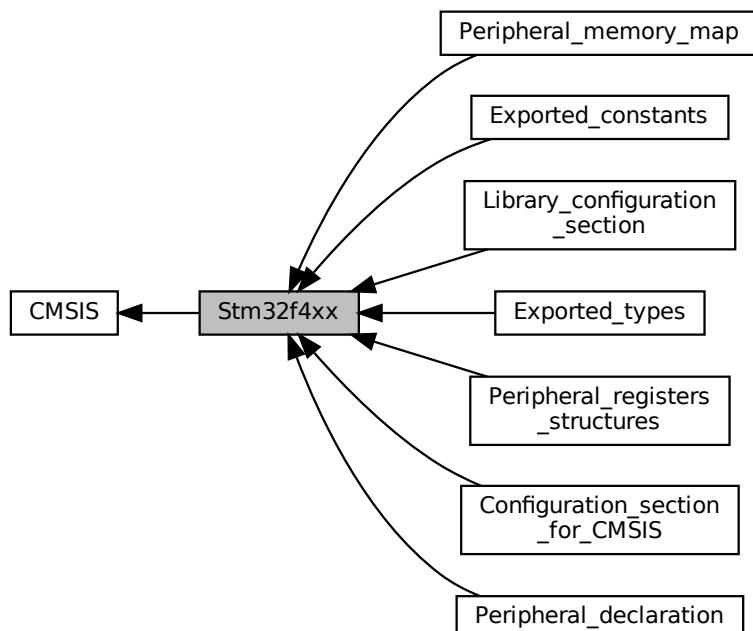
5.164.4.430 WVPCR

```
__IO uint32_t LTDC_Layer_TypeDef::WVPCR
```

LTDC Layerx Window Vertical Position Configuration Register Address offset: 0x8C

5.165 Stm32f4xx

Collaboration diagram for Stm32f4xx:



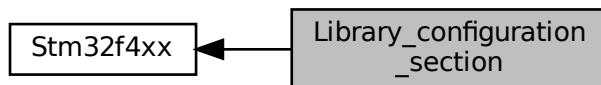
Modules

- [Library_configuration_section](#)
- [Configuration_section_for\CMSIS](#)
- [Exported_types](#)
- [Peripheral_registers_structures](#)
- [Peripheral_memory_map](#)
- [Peripheral_declaration](#)
- [Exported_constants](#)

5.165.1 Detailed Description

5.166 Library_configuration_section

Collaboration diagram for Library_configuration_section:



Macros

- `#define HSE_VALUE ((uint32_t)25000000)`
Comment the line below if you will not use the peripherals drivers. In this case, these drivers will not be included and the application code will be based on direct access to peripherals registers.
- `#define HSE_STARTUP_TIMEOUT ((uint16_t)0x05000)`
In the following line adjust the External High Speed oscillator (HSE) Startup Timeout value.
- `#define HSI_VALUE ((uint32_t)16000000)`
- `#define __STM32F4XX_STDPERIPH_VERSION_MAIN (0x01)`
STM32F4XX Standard Peripherals Library version number V1.8.0.
- `#define __STM32F4XX_STDPERIPH_VERSION_SUB1 (0x08)`
- `#define __STM32F4XX_STDPERIPH_VERSION_SUB2 (0x00)`
- `#define __STM32F4XX_STDPERIPH_VERSION_RC (0x00)`
- `#define __STM32F4XX_STDPERIPH_VERSION`

5.166.1 Detailed Description

5.166.2 Macro Definition Documentation

5.166.2.1 __STM32F4XX_STDPERIPH_VERSION

```
#define __STM32F4XX_STDPERIPH_VERSION
```

Value:

```
((__STM32F4XX_STDPERIPH_VERSION_MAIN << 24) \  
| (__STM32F4XX_STDPERIPH_VERSION_SUB1 << 16) \  
| (__STM32F4XX_STDPERIPH_VERSION_SUB2 << 8) \  
| (__STM32F4XX_STDPERIPH_VERSION_RC))
```

5.166.2.2 __STM32F4XX_STDPERIPH_VERSION_MAIN

```
#define __STM32F4XX_STDPERIPH_VERSION_MAIN (0x01)
```

STM32F4XX Standard Peripherals Library version number V1.8.0.

[31:24] main version

5.166.2.3 __STM32F4XX_STDPERIPH_VERSION_RC

```
#define __STM32F4XX_STDPERIPH_VERSION_RC (0x00)
```

[7:0] release candidate

5.166.2.4 __STM32F4XX_STDPERIPH_VERSION_SUB1

```
#define __STM32F4XX_STDPERIPH_VERSION_SUB1 (0x08)
```

[23:16] sub1 version

5.166.2.5 __STM32F4XX_STDPERIPH_VERSION_SUB2

```
#define __STM32F4XX_STDPERIPH_VERSION_SUB2 (0x00)
```

[15:8] sub2 version

5.166.2.6 HSE_STARTUP_TIMEOUT

```
#define HSE_STARTUP_TIMEOUT ((uint16_t)0x05000)
```

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

Time out for HSE start up

5.166.2.7 HSE_VALUE

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

Comment the line below if you will not use the peripherals drivers. In this case, these drivers will not be included and the application code will be based on direct access to peripherals registers.

In the following line adjust the value of External High Speed oscillator (HSE) used in your application

Tip: To avoid modifying this file each time you need to use different HSE, you can define the HSE value in your toolchain compiler preprocessor. Value of the External oscillator in Hz

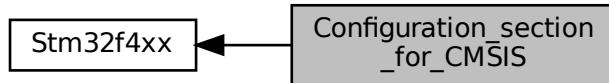
5.166.2.8 HSI_VALUE

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

Value of the Internal oscillator in Hz

5.167 Configuration_section_for_CMSIS

Collaboration diagram for Configuration_section_for_CMSIS:



Macros

- #define __CM4_REV 0x0001
Configuration of the Cortex-M4 Processor and Core Peripherals.
- #define __MPU_PRESENT 1
- #define __NVIC_PRIO_BITS 4
- #define __Vendor_SysTickConfig 0
- #define __FPU_PRESENT 1

Typedefs

- typedef enum IRQn IRQn_Type
STM32F4XX Interrupt Number Definition, according to the selected device in [Library_configuration_section](#).

Enumerations

- enum IRQn {
 NonMaskableInt_IRQn = -14 , MemoryManagement_IRQn = -12 , BusFault_IRQn = -11 , UsageFault_IRQn = -10 ,
 SVCall_IRQn = -5 , DebugMonitor_IRQn = -4 , PendSV_IRQn = -2 , SysTick_IRQn = -1 ,
 WWDG_IRQn = 0 , PVD_IRQn = 1 , TAMP_STAMP_IRQn = 2 , RTC_WKUP_IRQn = 3 ,
 FLASH_IRQn = 4 , RCC_IRQn = 5 , EXTI0_IRQn = 6 , EXTI1_IRQn = 7 ,
 EXTI2_IRQn = 8 , EXTI3_IRQn = 9 , EXTI4_IRQn = 10 , DMA1_Stream0_IRQn = 11 ,
 DMA1_Stream1_IRQn = 12 , DMA1_Stream2_IRQn = 13 , DMA1_Stream3_IRQn = 14 , DMA1_Stream4_IRQn = 15 ,
 DMA1_Stream5_IRQn = 16 , DMA1_Stream6_IRQn = 17 , ADC_IRQn = 18 , EXTI9_5_IRQn = 23 ,
 TIM1_BRK_TIM9_IRQn = 24 , TIM1_UP_TIM10_IRQn = 25 , TIM1_TRG_COM_TIM11_IRQn = 26 ,
 TIM1_CC_IRQn = 27 ,
 TIM2_IRQn = 28 , TIM3_IRQn = 29 , TIM4_IRQn = 30 , I2C1_EV_IRQn = 31 ,
 I2C1_ER_IRQn = 32 , I2C2_EV_IRQn = 33 , I2C2_ER_IRQn = 34 , SPI1_IRQn = 35 ,
 SPI2_IRQn = 36 , USART1_IRQn = 37 , USART2_IRQn = 38 , EXTI15_10_IRQn = 40 ,
 RTC_Alarm_IRQn = 41 , OTG_FS_WKUP_IRQn = 42 , DMA1_Stream7_IRQn = 47 , SDIO_IRQn = 49 ,
 TIM5_IRQn = 50 , SPI3_IRQn = 51 , DMA2_Stream0_IRQn = 56 , DMA2_Stream1_IRQn = 57 ,
 DMA2_Stream2_IRQn = 58 , DMA2_Stream3_IRQn = 59 , DMA2_Stream4_IRQn = 60 , OTG_FS_IRQn = 67 ,
 DMA2_Stream5_IRQn = 68 , DMA2_Stream6_IRQn = 69 , DMA2_Stream7_IRQn = 70 , USART6_IRQn = 71 ,
 I2C3_EV_IRQn = 72 , I2C3_ER_IRQn = 73 , FPU_IRQn = 81 , SPI4_IRQn = 84 }

STM32F4XX Interrupt Number Definition, according to the selected device in [Library configuration section](#).

5.167.1 Detailed Description

5.167.2 Macro Definition Documentation

5.167.2.1 __CM4_REV

```
#define __CM4_REV 0x0001
```

Configuration of the Cortex-M4 Processor and Core Peripherals.

Core revision r0p1

5.167.2.2 __FPU_PRESENT

```
#define __FPU_PRESENT 1
```

FPU present

5.167.2.3 __MPU_PRESENT

```
#define __MPU_PRESENT 1
```

STM32F4XX provides an MPU

5.167.2.4 __NVIC_PRIO_BITS

```
#define __NVIC_PRIO_BITS 4
```

STM32F4XX uses 4 Bits for the Priority Levels

5.167.2.5 __Vendor_SysTickConfig

```
#define __Vendor_SysTickConfig 0
```

Set to 1 if different SysTick Config is used

5.167.3 Enumeration Type Documentation

5.167.3.1 IRQn

```
enum IRQn
```

STM32F4XX Interrupt Number Definition, according to the selected device in [Library_configuration_section](#).

Enumerator

NonMaskableInt_IRQn	2 Non Maskable Interrupt
MemoryManagement_IRQn	4 Cortex-M4 Memory Management Interrupt
BusFault_IRQn	5 Cortex-M4 Bus Fault Interrupt
UsageFault_IRQn	6 Cortex-M4 Usage Fault Interrupt
SVCall_IRQn	11 Cortex-M4 SV Call Interrupt
DebugMonitor_IRQn	12 Cortex-M4 Debug Monitor Interrupt
PendSV_IRQn	14 Cortex-M4 Pend SV Interrupt
SysTick_IRQn	15 Cortex-M4 System Tick Interrupt
WWDG_IRQn	Window WatchDog Interrupt

Enumerator

PVD_IRQn	PVD through EXTI Line detection Interrupt
TAMP_STAMP_IRQn	Tamper andTimeStamp interrupts through the EXTI line
RTC_WKUP_IRQn	RTC Wakeup interrupt through the EXTI line
FLASH_IRQn	FLASH global Interrupt
RCC_IRQn	RCC global Interrupt
EXTI0_IRQn	EXTI Line0 Interrupt
EXTI1_IRQn	EXTI Line1 Interrupt
EXTI2_IRQn	EXTI Line2 Interrupt
EXTI3_IRQn	EXTI Line3 Interrupt
EXTI4_IRQn	EXTI Line4 Interrupt
DMA1_Stream0_IRQn	DMA1 Stream 0 global Interrupt
DMA1_Stream1_IRQn	DMA1 Stream 1 global Interrupt
DMA1_Stream2_IRQn	DMA1 Stream 2 global Interrupt
DMA1_Stream3_IRQn	DMA1 Stream 3 global Interrupt
DMA1_Stream4_IRQn	DMA1 Stream 4 global Interrupt
DMA1_Stream5_IRQn	DMA1 Stream 5 global Interrupt
DMA1_Stream6_IRQn	DMA1 Stream 6 global Interrupt
ADC_IRQn	ADC1, ADC2 and ADC3 global Interrupts
EXTI9_5_IRQn	External Line[9:5] Interrupts
TIM1_BRK_TIM9_IRQn	TIM1 Break interrupt and TIM9 global interrupt
TIM1_UP_TIM10_IRQn	TIM1 Update Interrupt and TIM10 global interrupt
TIM1_TRG_COM_TIM11_IRQn	TIM1 Trigger and Commutation Interrupt and TIM11 global interrupt
TIM1_CC_IRQn	TIM1 Capture Compare Interrupt
TIM2_IRQn	TIM2 global Interrupt
TIM3_IRQn	TIM3 global Interrupt
TIM4_IRQn	TIM4 global Interrupt
I2C1_EV_IRQn	I2C1 Event Interrupt
I2C1_ER_IRQn	I2C1 Error Interrupt

Enumerator

I2C2_EV_IRQn	I2C2 Event Interrupt
I2C2_ER_IRQn	I2C2 Error Interrupt
SPI1_IRQn	SPI1 global Interrupt
SPI2_IRQn	SPI2 global Interrupt
USART1_IRQn	USART1 global Interrupt
USART2_IRQn	USART2 global Interrupt
EXTI15_10_IRQn	External Line[15:10] Interrupts
RTC_Alarm_IRQn	RTC Alarm (A and B) through EXTI Line Interrupt
OTG_FS_WKUP_IRQn	USB OTG FS Wakeup through EXTI line interrupt
DMA1_Stream7_IRQn	DMA1 Stream7 Interrupt
SDIO_IRQn	SDIO global Interrupt
TIM5_IRQn	TIM5 global Interrupt
SPI3_IRQn	SPI3 global Interrupt
DMA2_Stream0_IRQn	DMA2 Stream 0 global Interrupt
DMA2_Stream1_IRQn	DMA2 Stream 1 global Interrupt
DMA2_Stream2_IRQn	DMA2 Stream 2 global Interrupt
DMA2_Stream3_IRQn	DMA2 Stream 3 global Interrupt
DMA2_Stream4_IRQn	DMA2 Stream 4 global Interrupt
OTG_FS_IRQn	USB OTG FS global Interrupt
DMA2_Stream5_IRQn	DMA2 Stream 5 global interrupt
DMA2_Stream6_IRQn	DMA2 Stream 6 global interrupt
DMA2_Stream7_IRQn	DMA2 Stream 7 global interrupt
USART6_IRQn	USART6 global interrupt
I2C3_EV_IRQn	I2C3 event interrupt
I2C3_ER_IRQn	I2C3 error interrupt
FPU_IRQn	FPU global interrupt
SPI4_IRQn	SPI4 global Interrupt

5.168 Exported_types

Collaboration diagram for Exported_types:



Macros

- #define **IS_FUNCTIONAL_STATE**(STATE) (((STATE) == **DISABLE**) || ((STATE) == **ENABLE**))

Typedefs

- typedef int32_t **s32**
- typedef int16_t **s16**
- typedef int8_t **s8**
- typedef const int32_t **sc32**
- typedef const int16_t **sc16**
- typedef const int8_t **sc8**
- typedef **_IO** int32_t **vs32**
- typedef **_IO** int16_t **vs16**
- typedef **_IO** int8_t **vs8**
- typedef **_I** int32_t **vsc32**
- typedef **_I** int16_t **vsc16**
- typedef **_I** int8_t **vsc8**
- typedef uint32_t **u32**
- typedef uint16_t **u16**
- typedef uint8_t **u8**
- typedef const uint32_t **uc32**
- typedef const uint16_t **uc16**
- typedef const uint8_t **uc8**
- typedef **_IO** uint32_t **vu32**
- typedef **_IO** uint16_t **vu16**
- typedef **_IO** uint8_t **vu8**
- typedef **_I** uint32_t **vuc32**
- typedef **_I** uint16_t **vuc16**
- typedef **_I** uint8_t **vuc8**
- typedef enum **FlagStatus** **ITStatus**

Enumerations

- enum **FlagStatus** { **RESET** = 0 , **SET** = !**RESET** , **Reset** , **Set** }
- enum **FunctionalState** { **DISABLE** = 0 , **ENABLE** = !**DISABLE** }
- enum **ErrorStatus** { **ERROR** = 0 , **SUCCESS** = !**ERROR** }

5.168.1 Detailed Description

5.168.2 Typedef Documentation

5.168.2.1 `s32`

```
typedef int32_t s32
< STM32F10x Standard Peripheral Library old types (maintained for legacy purpose)
```

5.168.2.2 `sc16`

```
typedef const int16_t sc16
```

Read Only

5.168.2.3 `sc32`

```
typedef const int32_t sc32
Read Only
```

5.168.2.4 `sc8`

```
typedef const int8_t sc8
Read Only
```

5.168.2.5 `uc16`

```
typedef const uint16_t uc16
Read Only
```

5.168.2.6 `uc32`

```
typedef const uint32_t uc32
Read Only
```

5.168.2.7 uc8

```
typedef const uint8_t uc8
```

Read Only

5.168.2.8 vsc16

```
typedef __I int16_t vsc16
```

Read Only

5.168.2.9 vsc32

```
typedef __I int32_t vsc32
```

Read Only

5.168.2.10 vsc8

```
typedef __I int8_t vsc8
```

Read Only

5.168.2.11 vuc16

```
typedef __I uint16_t vuc16
```

Read Only

5.168.2.12 vuc32

```
typedef __I uint32_t vuc32
```

Read Only

5.168.2.13 vuc8

```
typedef __I uint8_t vuc8
```

Read Only

5.168.3 Enumeration Type Documentation**5.168.3.1 FlagStatus**

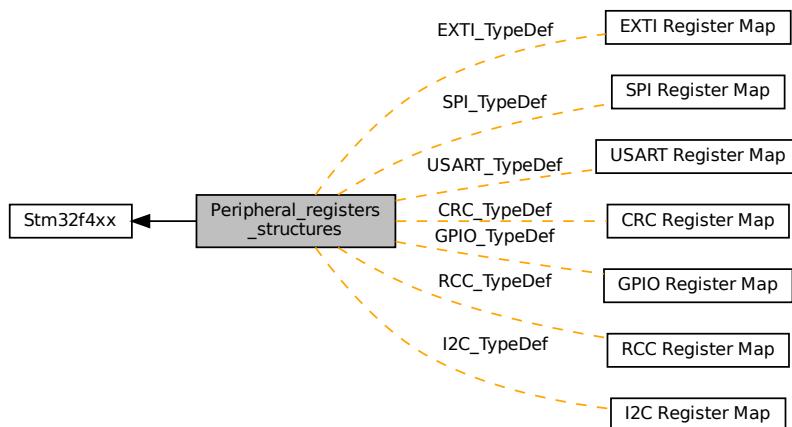
```
enum FlagStatus
```

Enumerator

Reset	Flag is reset.
Set	Flag is set.

5.169 Peripheral_registers_structures

Collaboration diagram for Peripheral_registers_structures:



Classes

- struct [ADC_TypeDef](#)
Analog to Digital Converter
- struct [ADC_Common_TypeDef](#)
- struct [CAN_TxMailBox_TypeDef](#)
Controller Area Network TxMailBox.
- struct [CAN_FIFOMailBox_TypeDef](#)
Controller Area Network FIFOMailBox.
- struct [CAN_FilterRegister_TypeDef](#)
Controller Area Network FilterRegister.
- struct [CAN_TypeDef](#)
Controller Area Network.
- struct [CRC_TypeDef](#)
CRC calculation unit.
- struct [DAC_TypeDef](#)
Digital to Analog Converter.
- struct [DBGMCU_TypeDef](#)
Debug MCU.
- struct [DCMI_TypeDef](#)

- struct [DMA_Stream_TypeDef](#)
DMA Controller.
- struct [DMA_TypeDef](#)
- struct [DMA2D_TypeDef](#)
DMA2D Controller.
- struct [ETH_TypeDef](#)
Ethernet MAC.
- struct [EXTI_TypeDef](#)
External Interrupt/Event Controller.
- struct [FLASH_TypeDef](#)
FLASH Registers.
- struct [GPIO_TypeDef](#)
General Purpose I/O.
- struct [SYSCFG_TypeDef](#)
System configuration controller.
- struct [I2C_TypeDef](#)
Inter-integrated Circuit Interface.
- struct [IWDG_TypeDef](#)
Independent WATCHDOG.
- struct [LTDC_TypeDef](#)
LCD-TFT Display Controller.
- struct [LTDC_Layer_TypeDef](#)
LCD-TFT Display layer x Controller.
- struct [PWR_TypeDef](#)
Power Control.
- struct [RCC_TypeDef](#)
Reset and Clock Control.
- struct [RTC_TypeDef](#)
Real-Time Clock.
- struct [SAI_TypeDef](#)
Serial Audio Interface.
- struct [SAI_Block_TypeDef](#)
- struct [SDIO_TypeDef](#)
SD host Interface.
- struct [SPI_TypeDef](#)
Serial Peripheral Interface.
- struct [TIM_TypeDef](#)
TIM.
- struct [USART_TypeDef](#)
Universal Synchronous Asynchronous Receiver Transmitter.
- struct [WWDG_TypeDef](#)
Window WATCHDOG.
- struct [CRYP_TypeDef](#)
Crypto Processor.
- struct [HASH_TypeDef](#)
HASH.
- struct [HASH_DIGEST_TypeDef](#)
HASH_DIGEST.
- struct [RNG_TypeDef](#)
RNG.

5.169.1 Detailed Description

5.170 Peripheral_memory_map

Collaboration diagram for Peripheral_memory_map:



Macros

- #define FLASH_BASE ((uint32_t)0x08000000)
- #define CCMDATARAM_BASE ((uint32_t)0x10000000)
- #define SRAM1_BASE ((uint32_t)0x20000000)
- #define PERIPH_BASE ((uint32_t)0x40000000)
- #define BKPSRAM_BASE ((uint32_t)0x40024000)
- #define CCMDATARAM_BB_BASE ((uint32_t)0x12000000)
- #define SRAM1_BB_BASE ((uint32_t)0x22000000)
- #define PERIPH_BB_BASE ((uint32_t)0x42000000)
- #define BKPSRAM_BB_BASE ((uint32_t)0x42480000)
- #define SRAM_BASE SRAM1_BASE
- #define SRAM_BB_BASE SRAM1_BB_BASE
- #define APB1PERIPH_BASE PERIPH_BASE
- #define APB2PERIPH_BASE (PERIPH_BASE + 0x00010000)
- #define AHB1PERIPH_BASE (PERIPH_BASE + 0x00020000)
- #define AHB2PERIPH_BASE (PERIPH_BASE + 0x10000000)
- #define TIM2_BASE (APB1PERIPH_BASE + 0x0000)
- #define TIM3_BASE (APB1PERIPH_BASE + 0x0400)
- #define TIM4_BASE (APB1PERIPH_BASE + 0x0800)
- #define TIM5_BASE (APB1PERIPH_BASE + 0x0C00)
- #define TIM6_BASE (APB1PERIPH_BASE + 0x1000)
- #define TIM7_BASE (APB1PERIPH_BASE + 0x1400)
- #define TIM12_BASE (APB1PERIPH_BASE + 0x1800)
- #define TIM13_BASE (APB1PERIPH_BASE + 0x1C00)
- #define TIM14_BASE (APB1PERIPH_BASE + 0x2000)
- #define RTC_BASE (APB1PERIPH_BASE + 0x2800)
- #define WWDG_BASE (APB1PERIPH_BASE + 0x2C00)
- #define IWDG_BASE (APB1PERIPH_BASE + 0x3000)
- #define I2S2ext_BASE (APB1PERIPH_BASE + 0x3400)
- #define SPI2_BASE (APB1PERIPH_BASE + 0x3800)
- #define SPI3_BASE (APB1PERIPH_BASE + 0x3C00)
- #define I2S3ext_BASE (APB1PERIPH_BASE + 0x4000)
- #define USART2_BASE (APB1PERIPH_BASE + 0x4400)
- #define USART3_BASE (APB1PERIPH_BASE + 0x4800)
- #define UART4_BASE (APB1PERIPH_BASE + 0x4C00)

- #define **UART5_BASE** (APB1PERIPH_BASE + 0x5000)
- #define **I2C1_BASE** (APB1PERIPH_BASE + 0x5400)
- #define **I2C2_BASE** (APB1PERIPH_BASE + 0x5800)
- #define **I2C3_BASE** (APB1PERIPH_BASE + 0x5C00)
- #define **CAN1_BASE** (APB1PERIPH_BASE + 0x6400)
- #define **CAN2_BASE** (APB1PERIPH_BASE + 0x6800)
- #define **PWR_BASE** (APB1PERIPH_BASE + 0x7000)
- #define **DAC_BASE** (APB1PERIPH_BASE + 0x7400)
- #define **UART7_BASE** (APB1PERIPH_BASE + 0x7800)
- #define **UART8_BASE** (APB1PERIPH_BASE + 0x7C00)
- #define **TIM1_BASE** (APB2PERIPH_BASE + 0x0000)
- #define **TIM8_BASE** (APB2PERIPH_BASE + 0x0400)
- #define **USART1_BASE** (APB2PERIPH_BASE + 0x1000)
- #define **USART6_BASE** (APB2PERIPH_BASE + 0x1400)
- #define **UART9_BASE** (APB2PERIPH_BASE + 0x1800U)
- #define **UART10_BASE** (APB2PERIPH_BASE + 0x1C00U)
- #define **ADC1_BASE** (APB2PERIPH_BASE + 0x2000)
- #define **ADC2_BASE** (APB2PERIPH_BASE + 0x2100)
- #define **ADC3_BASE** (APB2PERIPH_BASE + 0x2200)
- #define **ADC_BASE** (APB2PERIPH_BASE + 0x2300)
- #define **SDIO_BASE** (APB2PERIPH_BASE + 0x2C00)
- #define **SPI1_BASE** (APB2PERIPH_BASE + 0x3000)
- #define **SPI4_BASE** (APB2PERIPH_BASE + 0x3400)
- #define **SYSCFG_BASE** (APB2PERIPH_BASE + 0x3800)
- #define **EXTI_BASE** (APB2PERIPH_BASE + 0x3C00)
- #define **TIM9_BASE** (APB2PERIPH_BASE + 0x4000)
- #define **TIM10_BASE** (APB2PERIPH_BASE + 0x4400)
- #define **TIM11_BASE** (APB2PERIPH_BASE + 0x4800)
- #define **SPI5_BASE** (APB2PERIPH_BASE + 0x5000)
- #define **SPI6_BASE** (APB2PERIPH_BASE + 0x5400)
- #define **SAI1_BASE** (APB2PERIPH_BASE + 0x5800)
- #define **SAI1_Block_A_BASE** (SAI1_BASE + 0x004)
- #define **SAI1_Block_B_BASE** (SAI1_BASE + 0x024)
- #define **LTDC_BASE** (APB2PERIPH_BASE + 0x6800)
- #define **LTDC_Layer1_BASE** (LTDC_BASE + 0x84)
- #define **LTDC_Layer2_BASE** (LTDC_BASE + 0x104)
- #define **GPIOA_BASE** (AHB1PERIPH_BASE + 0x0000)
- #define **GPIOB_BASE** (AHB1PERIPH_BASE + 0x0400)
- #define **GPIOC_BASE** (AHB1PERIPH_BASE + 0x0800)
- #define **GPIOD_BASE** (AHB1PERIPH_BASE + 0x0C00)
- #define **GPIOE_BASE** (AHB1PERIPH_BASE + 0x1000)
- #define **GPIOF_BASE** (AHB1PERIPH_BASE + 0x1400)
- #define **GPIOG_BASE** (AHB1PERIPH_BASE + 0x1800)
- #define **GPIOH_BASE** (AHB1PERIPH_BASE + 0x1C00)
- #define **GPIOI_BASE** (AHB1PERIPH_BASE + 0x2000)
- #define **GPIOJ_BASE** (AHB1PERIPH_BASE + 0x2400)
- #define **GPIOK_BASE** (AHB1PERIPH_BASE + 0x2800)
- #define **CRC_BASE** (AHB1PERIPH_BASE + 0x3000)
- #define **RCC_BASE** (AHB1PERIPH_BASE + 0x3800)
- #define **FLASH_R_BASE** (AHB1PERIPH_BASE + 0x3C00)
- #define **DMA1_BASE** (AHB1PERIPH_BASE + 0x6000)
- #define **DMA1_Stream0_BASE** (DMA1_BASE + 0x010)
- #define **DMA1_Stream1_BASE** (DMA1_BASE + 0x028)
- #define **DMA1_Stream2_BASE** (DMA1_BASE + 0x040)
- #define **DMA1_Stream3_BASE** (DMA1_BASE + 0x058)

- #define **DMA1_Stream4_BASE** (DMA1_BASE + 0x070)
- #define **DMA1_Stream5_BASE** (DMA1_BASE + 0x088)
- #define **DMA1_Stream6_BASE** (DMA1_BASE + 0x0A0)
- #define **DMA1_Stream7_BASE** (DMA1_BASE + 0x0B8)
- #define **DMA2_BASE** (AHB1PERIPH_BASE + 0x6400)
- #define **DMA2_Stream0_BASE** (DMA2_BASE + 0x010)
- #define **DMA2_Stream1_BASE** (DMA2_BASE + 0x028)
- #define **DMA2_Stream2_BASE** (DMA2_BASE + 0x040)
- #define **DMA2_Stream3_BASE** (DMA2_BASE + 0x058)
- #define **DMA2_Stream4_BASE** (DMA2_BASE + 0x070)
- #define **DMA2_Stream5_BASE** (DMA2_BASE + 0x088)
- #define **DMA2_Stream6_BASE** (DMA2_BASE + 0x0A0)
- #define **DMA2_Stream7_BASE** (DMA2_BASE + 0x0B8)
- #define **ETH_BASE** (AHB1PERIPH_BASE + 0x8000)
- #define **ETH_MAC_BASE** (ETH_BASE)
- #define **ETH MMC_BASE** (ETH_BASE + 0x0100)
- #define **ETH_PTP_BASE** (ETH_BASE + 0x0700)
- #define **ETH_DMA_BASE** (ETH_BASE + 0x1000)
- #define **DMA2D_BASE** (AHB1PERIPH_BASE + 0xB000)
- #define **DCMI_BASE** (AHB2PERIPH_BASE + 0x50000)
- #define **CRYP_BASE** (AHB2PERIPH_BASE + 0x60000)
- #define **HASH_BASE** (AHB2PERIPH_BASE + 0x60400)
- #define **HASH_DIGEST_BASE** (AHB2PERIPH_BASE + 0x60710)
- #define **RNG_BASE** (AHB2PERIPH_BASE + 0x60800)
- #define **DBGMCU_BASE** ((uint32_t)0xE0042000)

5.170.1 Detailed Description

5.170.2 Macro Definition Documentation

5.170.2.1 AHB2PERIPH_BASE

```
#define AHB2PERIPH_BASE (PERIPH_BASE + 0x10000000)
```

APB1 peripherals

5.170.2.2 BKPSRAM_BASE

```
#define BKPSRAM_BASE ((uint32_t)0x40024000)
```

Backup SRAM(4 KB) base address in the alias region

5.170.2.3 BKPSRAM_BB_BASE

```
#define BKPSRAM_BB_BASE ((uint32_t)0x42480000)
```

Backup SRAM(4 KB) base address in the bit-band region

5.170.2.4 CCMDATARAM_BASE

```
#define CCMDATARAM_BASE ((uint32_t)0x10000000)
```

CCM(core coupled memory) data RAM(64 KB) base address in the alias region

5.170.2.5 CCMDATARAM_BB_BASE

```
#define CCMDATARAM_BB_BASE ((uint32_t)0x12000000)
```

CCM(core coupled memory) data RAM(64 KB) base address in the bit-band region

5.170.2.6 DMA2D_BASE

```
#define DMA2D_BASE (AHB1PERIPH_BASE + 0xB000)
```

AHB2 peripherals

5.170.2.7 FLASH_BASE

```
#define FLASH_BASE ((uint32_t)0x08000000)
```

FLASH(up to 1 MB) base address in the alias region

5.170.2.8 GPIOA_BASE

```
#define GPIOA_BASE (AHB1PERIPH_BASE + 0x0000)
```

< AHB1 peripherals

5.170.2.9 PERIPH_BASE

```
#define PERIPH_BASE ((uint32_t)0x40000000)
```

Peripheral base address in the alias region

5.170.2.10 PERIPH_BB_BASE

```
#define PERIPH_BB_BASE ((uint32_t)0x42000000)
```

Peripheral base address in the bit-band region

5.170.2.11 SRAM1_BASE

```
#define SRAM1_BASE ((uint32_t)0x20000000)
```

SRAM1(112 KB) base address in the alias region

5.170.2.12 SRAM1_BB_BASE

```
#define SRAM1_BB_BASE ((uint32_t)0x22000000)
```

SRAM1(112 KB) base address in the bit-band region

5.170.2.13 SRAM_BB_BASE

```
#define SRAM_BB_BASE SRAM1_BB_BASE
```

Peripheral memory map

5.170.2.14 UART8_BASE

```
#define UART8_BASE (APB1PERIPH_BASE + 0x7C00)
```

APB2 peripherals

5.171 Peripheral_declaration

Collaboration diagram for Peripheral_declaration:



Macros

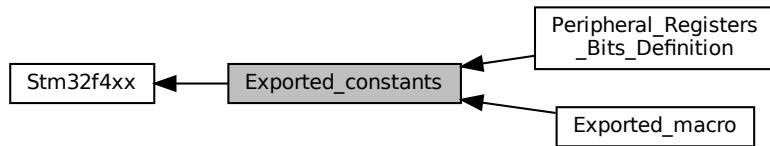
- #define **TIM2** ((**TIM_TypeDef** *) **TIM2_BASE**)
- #define **TIM3** ((**TIM_TypeDef** *) **TIM3_BASE**)
- #define **TIM4** ((**TIM_TypeDef** *) **TIM4_BASE**)
- #define **TIM5** ((**TIM_TypeDef** *) **TIM5_BASE**)
- #define **TIM6** ((**TIM_TypeDef** *) **TIM6_BASE**)
- #define **TIM7** ((**TIM_TypeDef** *) **TIM7_BASE**)
- #define **TIM12** ((**TIM_TypeDef** *) **TIM12_BASE**)
- #define **TIM13** ((**TIM_TypeDef** *) **TIM13_BASE**)
- #define **TIM14** ((**TIM_TypeDef** *) **TIM14_BASE**)
- #define **RTC** ((**RTC_TypeDef** *) **RTC_BASE**)
- #define **WWDG** ((**WWDG_TypeDef** *) **WWDG_BASE**)
- #define **IWDG** ((**IWDG_TypeDef** *) **IWDG_BASE**)
- #define **I2S2ext** ((**SPI_TypeDef** *) **I2S2ext_BASE**)
- #define **SPI2** ((**SPI_TypeDef** *) **SPI2_BASE**)
- #define **SPI3** ((**SPI_TypeDef** *) **SPI3_BASE**)
- #define **I2S3ext** ((**SPI_TypeDef** *) **I2S3ext_BASE**)
- #define **USART2** ((**USART_TypeDef** *) **USART2_BASE**)
- #define **USART3** ((**USART_TypeDef** *) **USART3_BASE**)
- #define **UART4** ((**USART_TypeDef** *) **UART4_BASE**)
- #define **UART5** ((**USART_TypeDef** *) **UART5_BASE**)
- #define **I2C1** ((**I2C_TypeDef** *) **I2C1_BASE**)
- #define **I2C2** ((**I2C_TypeDef** *) **I2C2_BASE**)
- #define **I2C3** ((**I2C_TypeDef** *) **I2C3_BASE**)
- #define **CAN1** ((**CAN_TypeDef** *) **CAN1_BASE**)
- #define **CAN2** ((**CAN_TypeDef** *) **CAN2_BASE**)
- #define **PWR** ((**PWR_TypeDef** *) **PWR_BASE**)
- #define **DAC** ((**DAC_TypeDef** *) **DAC_BASE**)
- #define **UART7** ((**USART_TypeDef** *) **UART7_BASE**)
- #define **UART8** ((**USART_TypeDef** *) **UART8_BASE**)
- #define **UART9** ((**USART_TypeDef** *) **UART9_BASE**)
- #define **UART10** ((**USART_TypeDef** *) **UART10_BASE**)
- #define **TIM1** ((**TIM_TypeDef** *) **TIM1_BASE**)
- #define **TIM8** ((**TIM_TypeDef** *) **TIM8_BASE**)
- #define **USART1** ((**USART_TypeDef** *) **USART1_BASE**)
- #define **USART6** ((**USART_TypeDef** *) **USART6_BASE**)
- #define **ADC** ((**ADC_Common_TypeDef** *) **ADC_BASE**)
- #define **ADC1** ((**ADC_TypeDef** *) **ADC1_BASE**)
- #define **ADC2** ((**ADC_TypeDef** *) **ADC2_BASE**)
- #define **ADC3** ((**ADC_TypeDef** *) **ADC3_BASE**)
- #define **SDIO** ((**SDIO_TypeDef** *) **SDIO_BASE**)
- #define **SPI1** ((**SPI_TypeDef** *) **SPI1_BASE**)
- #define **SPI4** ((**SPI_TypeDef** *) **SPI4_BASE**)
- #define **SYSCFG** ((**SYSCFG_TypeDef** *) **SYSCFG_BASE**)
- #define **EXTI** ((**EXTI_TypeDef** *) **EXTI_BASE**)
- #define **TIM9** ((**TIM_TypeDef** *) **TIM9_BASE**)
- #define **TIM10** ((**TIM_TypeDef** *) **TIM10_BASE**)
- #define **TIM11** ((**TIM_TypeDef** *) **TIM11_BASE**)
- #define **SPI5** ((**SPI_TypeDef** *) **SPI5_BASE**)
- #define **SPI6** ((**SPI_TypeDef** *) **SPI6_BASE**)
- #define **SAI1** ((**SAI_TypeDef** *) **SAI1_BASE**)
- #define **SAI1_Block_A** ((**SAI_Block_TypeDef** *) **SAI1_Block_A_BASE**)
- #define **SAI1_Block_B** ((**SAI_Block_TypeDef** *) **SAI1_Block_B_BASE**)
- #define **LTDC** ((**LTDC_TypeDef** *) **LTDC_BASE**)

- #define **LTDC_Layer1** ((**LTDC_Layer_TypeDef** *)LTDC_Layer1_BASE)
- #define **LTDC_Layer2** ((**LTDC_Layer_TypeDef** *)LTDC_Layer2_BASE)
- #define **GPIOA** ((**GPIO_TypeDef** *) GPIOA_BASE)
- #define **GPIOB** ((**GPIO_TypeDef** *) GPIOB_BASE)
- #define **GPIOC** ((**GPIO_TypeDef** *) GPIOC_BASE)
- #define **GPIOD** ((**GPIO_TypeDef** *) GPIOD_BASE)
- #define **GPIOE** ((**GPIO_TypeDef** *) GPIOE_BASE)
- #define **GPIOF** ((**GPIO_TypeDef** *) GPIOF_BASE)
- #define **GPIOG** ((**GPIO_TypeDef** *) GPIOG_BASE)
- #define **GPIOH** ((**GPIO_TypeDef** *) GPIOH_BASE)
- #define **GPIOI** ((**GPIO_TypeDef** *) GPIOI_BASE)
- #define **GPIOJ** ((**GPIO_TypeDef** *) GPIOJ_BASE)
- #define **GPIOK** ((**GPIO_TypeDef** *) GPIOK_BASE)
- #define **CRC** ((**CRC_TypeDef** *) CRC_BASE)
- #define **RCC** ((**RCC_TypeDef** *) RCC_BASE)
- #define **FLASH** ((**FLASH_TypeDef** *) FLASH_R_BASE)
- #define **DMA1** ((**DMA_TypeDef** *) DMA1_BASE)
- #define **DMA1_Stream0** ((**DMA_Stream_TypeDef** *) DMA1_Stream0_BASE)
- #define **DMA1_Stream1** ((**DMA_Stream_TypeDef** *) DMA1_Stream1_BASE)
- #define **DMA1_Stream2** ((**DMA_Stream_TypeDef** *) DMA1_Stream2_BASE)
- #define **DMA1_Stream3** ((**DMA_Stream_TypeDef** *) DMA1_Stream3_BASE)
- #define **DMA1_Stream4** ((**DMA_Stream_TypeDef** *) DMA1_Stream4_BASE)
- #define **DMA1_Stream5** ((**DMA_Stream_TypeDef** *) DMA1_Stream5_BASE)
- #define **DMA1_Stream6** ((**DMA_Stream_TypeDef** *) DMA1_Stream6_BASE)
- #define **DMA1_Stream7** ((**DMA_Stream_TypeDef** *) DMA1_Stream7_BASE)
- #define **DMA2** ((**DMA_TypeDef** *) DMA2_BASE)
- #define **DMA2_Stream0** ((**DMA_Stream_TypeDef** *) DMA2_Stream0_BASE)
- #define **DMA2_Stream1** ((**DMA_Stream_TypeDef** *) DMA2_Stream1_BASE)
- #define **DMA2_Stream2** ((**DMA_Stream_TypeDef** *) DMA2_Stream2_BASE)
- #define **DMA2_Stream3** ((**DMA_Stream_TypeDef** *) DMA2_Stream3_BASE)
- #define **DMA2_Stream4** ((**DMA_Stream_TypeDef** *) DMA2_Stream4_BASE)
- #define **DMA2_Stream5** ((**DMA_Stream_TypeDef** *) DMA2_Stream5_BASE)
- #define **DMA2_Stream6** ((**DMA_Stream_TypeDef** *) DMA2_Stream6_BASE)
- #define **DMA2_Stream7** ((**DMA_Stream_TypeDef** *) DMA2_Stream7_BASE)
- #define **ETH** ((**ETH_TypeDef** *) ETH_BASE)
- #define **DMA2D** ((**DMA2D_TypeDef** *) DMA2D_BASE)
- #define **DCMI** ((**DCMI_TypeDef** *) DCMI_BASE)
- #define **CRYP** ((**CRYP_TypeDef** *) CRYP_BASE)
- #define **HASH** ((**HASH_TypeDef** *) HASH_BASE)
- #define **HASH_DIGEST** ((**HASH_DIGEST_TypeDef** *) HASH_DIGEST_BASE)
- #define **RNG** ((**RNG_TypeDef** *) RNG_BASE)
- #define **DBGMCU** ((**DBGMCU_TypeDef** *) DBGMCU_BASE)

5.171.1 Detailed Description

5.172 Exported_constants

Collaboration diagram for Exported_constants:



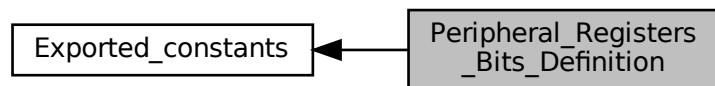
Modules

- [Peripheral_Registers_Bits_Definition](#)
- [Exported_macro](#)

5.172.1 Detailed Description

5.173 Peripheral_Registers_Bits_Definition

Collaboration diagram for Peripheral_Registers_Bits_Definition:



Macros

- [#define ADC_SR_AWD \(\(uint8_t\)0x01\)](#)
- [#define ADC_SR_EOC \(\(uint8_t\)0x02\)](#)
- [#define ADC_SR_JEOC \(\(uint8_t\)0x04\)](#)
- [#define ADC_SR_JSTRT \(\(uint8_t\)0x08\)](#)
- [#define ADC_SR_STRT \(\(uint8_t\)0x10\)](#)
- [#define ADC_SR_OVR \(\(uint8_t\)0x20\)](#)
- [#define ADC_CR1_AWDCH \(\(uint32_t\)0x0000001F\)](#)
- [#define ADC_CR1_AWDCH_0 \(\(uint32_t\)0x00000001\)](#)

- #define ADC_CR1_AWDCH_1 ((uint32_t)0x00000002)
- #define ADC_CR1_AWDCH_2 ((uint32_t)0x00000004)
- #define ADC_CR1_AWDCH_3 ((uint32_t)0x00000008)
- #define ADC_CR1_AWDCH_4 ((uint32_t)0x00000010)
- #define ADC_CR1_EOCIE ((uint32_t)0x00000020)
- #define ADC_CR1_AWDIE ((uint32_t)0x00000040)
- #define ADC_CR1_JEOCIE ((uint32_t)0x00000080)
- #define ADC_CR1_SCAN ((uint32_t)0x00000100)
- #define ADC_CR1_AWDSGL ((uint32_t)0x00000200)
- #define ADC_CR1_JAUTO ((uint32_t)0x00000400)
- #define ADC_CR1_DISCEN ((uint32_t)0x00000800)
- #define ADC_CR1_JDISCEN ((uint32_t)0x00001000)
- #define ADC_CR1_DISCNUM ((uint32_t)0x0000E000)
- #define ADC_CR1_DISCNUM_0 ((uint32_t)0x00002000)
- #define ADC_CR1_DISCNUM_1 ((uint32_t)0x00004000)
- #define ADC_CR1_DISCNUM_2 ((uint32_t)0x00008000)
- #define ADC_CR1_JAWDEN ((uint32_t)0x00400000)
- #define ADC_CR1_AWDEN ((uint32_t)0x00800000)
- #define ADC_CR1_RES ((uint32_t)0x03000000)
- #define ADC_CR1_RES_0 ((uint32_t)0x01000000)
- #define ADC_CR1_RES_1 ((uint32_t)0x02000000)
- #define ADC_CR1_OVRIE ((uint32_t)0x04000000)
- #define ADC_CR2_ADON ((uint32_t)0x00000001)
- #define ADC_CR2_CONT ((uint32_t)0x00000002)
- #define ADC_CR2_DMA ((uint32_t)0x00000100)
- #define ADC_CR2_DDS ((uint32_t)0x00000200)
- #define ADC_CR2_EOCS ((uint32_t)0x00000400)
- #define ADC_CR2_ALIGN ((uint32_t)0x00000800)
- #define ADC_CR2_JEXTSEL ((uint32_t)0x000F0000)
- #define ADC_CR2_JEXTSEL_0 ((uint32_t)0x00010000)
- #define ADC_CR2_JEXTSEL_1 ((uint32_t)0x00020000)
- #define ADC_CR2_JEXTSEL_2 ((uint32_t)0x00040000)
- #define ADC_CR2_JEXTSEL_3 ((uint32_t)0x00080000)
- #define ADC_CR2_JEXTEN ((uint32_t)0x00300000)
- #define ADC_CR2_JEXTEN_0 ((uint32_t)0x00100000)
- #define ADC_CR2_JEXTEN_1 ((uint32_t)0x00200000)
- #define ADC_CR2_JSWSTART ((uint32_t)0x00400000)
- #define ADC_CR2_EXTSEL ((uint32_t)0x0F000000)
- #define ADC_CR2_EXTSEL_0 ((uint32_t)0x01000000)
- #define ADC_CR2_EXTSEL_1 ((uint32_t)0x02000000)
- #define ADC_CR2_EXTSEL_2 ((uint32_t)0x04000000)
- #define ADC_CR2_EXTSEL_3 ((uint32_t)0x08000000)
- #define ADC_CR2_EXTN ((uint32_t)0x30000000)
- #define ADC_CR2_EXTN_0 ((uint32_t)0x10000000)
- #define ADC_CR2_EXTN_1 ((uint32_t)0x20000000)
- #define ADC_CR2_SWSTART ((uint32_t)0x40000000)
- #define ADC_SMPR1_SMP10 ((uint32_t)0x00000007)
- #define ADC_SMPR1_SMP10_0 ((uint32_t)0x00000001)
- #define ADC_SMPR1_SMP10_1 ((uint32_t)0x00000002)
- #define ADC_SMPR1_SMP10_2 ((uint32_t)0x00000004)
- #define ADC_SMPR1_SMP11 ((uint32_t)0x00000038)
- #define ADC_SMPR1_SMP11_0 ((uint32_t)0x00000008)
- #define ADC_SMPR1_SMP11_1 ((uint32_t)0x00000010)
- #define ADC_SMPR1_SMP11_2 ((uint32_t)0x00000020)
- #define ADC_SMPR1_SMP12 ((uint32_t)0x000001C0)

- #define ADC_SMPR1_SMP12_0 ((uint32_t)0x00000040)
- #define ADC_SMPR1_SMP12_1 ((uint32_t)0x00000080)
- #define ADC_SMPR1_SMP12_2 ((uint32_t)0x00000100)
- #define ADC_SMPR1_SMP13 ((uint32_t)0x00000E00)
- #define ADC_SMPR1_SMP13_0 ((uint32_t)0x00000200)
- #define ADC_SMPR1_SMP13_1 ((uint32_t)0x00000400)
- #define ADC_SMPR1_SMP13_2 ((uint32_t)0x00000800)
- #define ADC_SMPR1_SMP14 ((uint32_t)0x00007000)
- #define ADC_SMPR1_SMP14_0 ((uint32_t)0x00001000)
- #define ADC_SMPR1_SMP14_1 ((uint32_t)0x00002000)
- #define ADC_SMPR1_SMP14_2 ((uint32_t)0x00004000)
- #define ADC_SMPR1_SMP15 ((uint32_t)0x00038000)
- #define ADC_SMPR1_SMP15_0 ((uint32_t)0x00008000)
- #define ADC_SMPR1_SMP15_1 ((uint32_t)0x00010000)
- #define ADC_SMPR1_SMP15_2 ((uint32_t)0x00020000)
- #define ADC_SMPR1_SMP16 ((uint32_t)0x001C0000)
- #define ADC_SMPR1_SMP16_0 ((uint32_t)0x00040000)
- #define ADC_SMPR1_SMP16_1 ((uint32_t)0x00080000)
- #define ADC_SMPR1_SMP16_2 ((uint32_t)0x00100000)
- #define ADC_SMPR1_SMP17 ((uint32_t)0x00E00000)
- #define ADC_SMPR1_SMP17_0 ((uint32_t)0x00200000)
- #define ADC_SMPR1_SMP17_1 ((uint32_t)0x00400000)
- #define ADC_SMPR1_SMP17_2 ((uint32_t)0x00800000)
- #define ADC_SMPR1_SMP18 ((uint32_t)0x07000000)
- #define ADC_SMPR1_SMP18_0 ((uint32_t)0x01000000)
- #define ADC_SMPR1_SMP18_1 ((uint32_t)0x02000000)
- #define ADC_SMPR1_SMP18_2 ((uint32_t)0x04000000)
- #define ADC_SMPR2_SMP0 ((uint32_t)0x00000007)
- #define ADC_SMPR2_SMP0_0 ((uint32_t)0x00000001)
- #define ADC_SMPR2_SMP0_1 ((uint32_t)0x00000002)
- #define ADC_SMPR2_SMP0_2 ((uint32_t)0x00000004)
- #define ADC_SMPR2_SMP1 ((uint32_t)0x00000038)
- #define ADC_SMPR2_SMP1_0 ((uint32_t)0x00000008)
- #define ADC_SMPR2_SMP1_1 ((uint32_t)0x00000010)
- #define ADC_SMPR2_SMP1_2 ((uint32_t)0x00000020)
- #define ADC_SMPR2_SMP2 ((uint32_t)0x000001C0)
- #define ADC_SMPR2_SMP2_0 ((uint32_t)0x00000040)
- #define ADC_SMPR2_SMP2_1 ((uint32_t)0x00000080)
- #define ADC_SMPR2_SMP2_2 ((uint32_t)0x00000100)
- #define ADC_SMPR2_SMP3 ((uint32_t)0x000000E00)
- #define ADC_SMPR2_SMP3_0 ((uint32_t)0x00000200)
- #define ADC_SMPR2_SMP3_1 ((uint32_t)0x00000400)
- #define ADC_SMPR2_SMP3_2 ((uint32_t)0x00000800)
- #define ADC_SMPR2_SMP4 ((uint32_t)0x00007000)
- #define ADC_SMPR2_SMP4_0 ((uint32_t)0x00001000)
- #define ADC_SMPR2_SMP4_1 ((uint32_t)0x00002000)
- #define ADC_SMPR2_SMP4_2 ((uint32_t)0x00004000)
- #define ADC_SMPR2_SMP5 ((uint32_t)0x00038000)
- #define ADC_SMPR2_SMP5_0 ((uint32_t)0x00008000)
- #define ADC_SMPR2_SMP5_1 ((uint32_t)0x00010000)
- #define ADC_SMPR2_SMP5_2 ((uint32_t)0x00020000)
- #define ADC_SMPR2_SMP6 ((uint32_t)0x001C0000)
- #define ADC_SMPR2_SMP6_0 ((uint32_t)0x00040000)
- #define ADC_SMPR2_SMP6_1 ((uint32_t)0x00080000)
- #define ADC_SMPR2_SMP6_2 ((uint32_t)0x00100000)

```
• #define ADC_SMPR2_SMP7 ((uint32_t)0x00E00000)
• #define ADC_SMPR2_SMP7_0 ((uint32_t)0x00200000)
• #define ADC_SMPR2_SMP7_1 ((uint32_t)0x00400000)
• #define ADC_SMPR2_SMP7_2 ((uint32_t)0x00800000)
• #define ADC_SMPR2_SMP8 ((uint32_t)0x07000000)
• #define ADC_SMPR2_SMP8_0 ((uint32_t)0x01000000)
• #define ADC_SMPR2_SMP8_1 ((uint32_t)0x02000000)
• #define ADC_SMPR2_SMP8_2 ((uint32_t)0x04000000)
• #define ADC_SMPR2_SMP9 ((uint32_t)0x38000000)
• #define ADC_SMPR2_SMP9_0 ((uint32_t)0x08000000)
• #define ADC_SMPR2_SMP9_1 ((uint32_t)0x10000000)
• #define ADC_SMPR2_SMP9_2 ((uint32_t)0x20000000)
• #define ADC_JOFR1_JOFFSET1 ((uint16_t)0xFFFF)
• #define ADC_JOFR2_JOFFSET2 ((uint16_t)0xFFFF)
• #define ADC_JOFR3_JOFFSET3 ((uint16_t)0xFFFF)
• #define ADC_JOFR4_JOFFSET4 ((uint16_t)0xFFFF)
• #define ADC_HTR_HT ((uint16_t)0xFFFF)
• #define ADC_LTR_LT ((uint16_t)0xFFFF)
• #define ADC_SQR1_SQ13 ((uint32_t)0x00000001F)
• #define ADC_SQR1_SQ13_0 ((uint32_t)0x000000001)
• #define ADC_SQR1_SQ13_1 ((uint32_t)0x000000002)
• #define ADC_SQR1_SQ13_2 ((uint32_t)0x000000004)
• #define ADC_SQR1_SQ13_3 ((uint32_t)0x000000008)
• #define ADC_SQR1_SQ13_4 ((uint32_t)0x000000010)
• #define ADC_SQR1_SQ14 ((uint32_t)0x0000003E0)
• #define ADC_SQR1_SQ14_0 ((uint32_t)0x000000020)
• #define ADC_SQR1_SQ14_1 ((uint32_t)0x000000040)
• #define ADC_SQR1_SQ14_2 ((uint32_t)0x000000080)
• #define ADC_SQR1_SQ14_3 ((uint32_t)0x000000100)
• #define ADC_SQR1_SQ14_4 ((uint32_t)0x000000200)
• #define ADC_SQR1_SQ15 ((uint32_t)0x00007C00)
• #define ADC_SQR1_SQ15_0 ((uint32_t)0x00000400)
• #define ADC_SQR1_SQ15_1 ((uint32_t)0x00000800)
• #define ADC_SQR1_SQ15_2 ((uint32_t)0x00001000)
• #define ADC_SQR1_SQ15_3 ((uint32_t)0x00002000)
• #define ADC_SQR1_SQ15_4 ((uint32_t)0x00004000)
• #define ADC_SQR1_SQ16 ((uint32_t)0x000F8000)
• #define ADC_SQR1_SQ16_0 ((uint32_t)0x00008000)
• #define ADC_SQR1_SQ16_1 ((uint32_t)0x00010000)
• #define ADC_SQR1_SQ16_2 ((uint32_t)0x00020000)
• #define ADC_SQR1_SQ16_3 ((uint32_t)0x00040000)
• #define ADC_SQR1_SQ16_4 ((uint32_t)0x00080000)
• #define ADC_SQR1_L ((uint32_t)0x00F00000)
• #define ADC_SQR1_L_0 ((uint32_t)0x00100000)
• #define ADC_SQR1_L_1 ((uint32_t)0x00200000)
• #define ADC_SQR1_L_2 ((uint32_t)0x00400000)
• #define ADC_SQR1_L_3 ((uint32_t)0x00800000)
• #define ADC_SQR2_SQ7 ((uint32_t)0x0000001F)
• #define ADC_SQR2_SQ7_0 ((uint32_t)0x00000001)
• #define ADC_SQR2_SQ7_1 ((uint32_t)0x00000002)
• #define ADC_SQR2_SQ7_2 ((uint32_t)0x00000004)
• #define ADC_SQR2_SQ7_3 ((uint32_t)0x00000008)
• #define ADC_SQR2_SQ7_4 ((uint32_t)0x00000010)
• #define ADC_SQR2_SQ8 ((uint32_t)0x000003E0)
• #define ADC_SQR2_SQ8_0 ((uint32_t)0x00000020)
```

```
• #define ADC_SQR2_SQ8_1 ((uint32_t)0x00000040)
• #define ADC_SQR2_SQ8_2 ((uint32_t)0x00000080)
• #define ADC_SQR2_SQ8_3 ((uint32_t)0x00000100)
• #define ADC_SQR2_SQ8_4 ((uint32_t)0x00000200)
• #define ADC_SQR2_SQ9 ((uint32_t)0x00007C00)
• #define ADC_SQR2_SQ9_0 ((uint32_t)0x00000400)
• #define ADC_SQR2_SQ9_1 ((uint32_t)0x00000800)
• #define ADC_SQR2_SQ9_2 ((uint32_t)0x00001000)
• #define ADC_SQR2_SQ9_3 ((uint32_t)0x00002000)
• #define ADC_SQR2_SQ9_4 ((uint32_t)0x00004000)
• #define ADC_SQR2_SQ10 ((uint32_t)0x000F8000)
• #define ADC_SQR2_SQ10_0 ((uint32_t)0x00008000)
• #define ADC_SQR2_SQ10_1 ((uint32_t)0x00010000)
• #define ADC_SQR2_SQ10_2 ((uint32_t)0x00020000)
• #define ADC_SQR2_SQ10_3 ((uint32_t)0x00040000)
• #define ADC_SQR2_SQ10_4 ((uint32_t)0x00080000)
• #define ADC_SQR2_SQ11 ((uint32_t)0x01F00000)
• #define ADC_SQR2_SQ11_0 ((uint32_t)0x00100000)
• #define ADC_SQR2_SQ11_1 ((uint32_t)0x00200000)
• #define ADC_SQR2_SQ11_2 ((uint32_t)0x00400000)
• #define ADC_SQR2_SQ11_3 ((uint32_t)0x00800000)
• #define ADC_SQR2_SQ11_4 ((uint32_t)0x01000000)
• #define ADC_SQR2_SQ12 ((uint32_t)0x3E000000)
• #define ADC_SQR2_SQ12_0 ((uint32_t)0x02000000)
• #define ADC_SQR2_SQ12_1 ((uint32_t)0x04000000)
• #define ADC_SQR2_SQ12_2 ((uint32_t)0x08000000)
• #define ADC_SQR2_SQ12_3 ((uint32_t)0x10000000)
• #define ADC_SQR2_SQ12_4 ((uint32_t)0x20000000)
• #define ADC_SQR3_SQ1 ((uint32_t)0x0000001F)
• #define ADC_SQR3_SQ1_0 ((uint32_t)0x00000001)
• #define ADC_SQR3_SQ1_1 ((uint32_t)0x00000002)
• #define ADC_SQR3_SQ1_2 ((uint32_t)0x00000004)
• #define ADC_SQR3_SQ1_3 ((uint32_t)0x00000008)
• #define ADC_SQR3_SQ1_4 ((uint32_t)0x00000010)
• #define ADC_SQR3_SQ2 ((uint32_t)0x000003E0)
• #define ADC_SQR3_SQ2_0 ((uint32_t)0x00000020)
• #define ADC_SQR3_SQ2_1 ((uint32_t)0x00000040)
• #define ADC_SQR3_SQ2_2 ((uint32_t)0x00000080)
• #define ADC_SQR3_SQ2_3 ((uint32_t)0x00000100)
• #define ADC_SQR3_SQ2_4 ((uint32_t)0x00000200)
• #define ADC_SQR3_SQ3 ((uint32_t)0x00007C00)
• #define ADC_SQR3_SQ3_0 ((uint32_t)0x00000400)
• #define ADC_SQR3_SQ3_1 ((uint32_t)0x00000800)
• #define ADC_SQR3_SQ3_2 ((uint32_t)0x00001000)
• #define ADC_SQR3_SQ3_3 ((uint32_t)0x00002000)
• #define ADC_SQR3_SQ3_4 ((uint32_t)0x00004000)
• #define ADC_SQR3_SQ4 ((uint32_t)0x000F8000)
• #define ADC_SQR3_SQ4_0 ((uint32_t)0x00008000)
• #define ADC_SQR3_SQ4_1 ((uint32_t)0x00010000)
• #define ADC_SQR3_SQ4_2 ((uint32_t)0x00020000)
• #define ADC_SQR3_SQ4_3 ((uint32_t)0x00040000)
• #define ADC_SQR3_SQ4_4 ((uint32_t)0x00080000)
• #define ADC_SQR3_SQ5 ((uint32_t)0x01F00000)
• #define ADC_SQR3_SQ5_0 ((uint32_t)0x00100000)
• #define ADC_SQR3_SQ5_1 ((uint32_t)0x00200000)
```

- #define ADC_SQR3_SQ5_2 ((uint32_t)0x00400000)
- #define ADC_SQR3_SQ5_3 ((uint32_t)0x00800000)
- #define ADC_SQR3_SQ5_4 ((uint32_t)0x01000000)
- #define ADC_SQR3_SQ6 ((uint32_t)0x3E000000)
- #define ADC_SQR3_SQ6_0 ((uint32_t)0x02000000)
- #define ADC_SQR3_SQ6_1 ((uint32_t)0x04000000)
- #define ADC_SQR3_SQ6_2 ((uint32_t)0x08000000)
- #define ADC_SQR3_SQ6_3 ((uint32_t)0x10000000)
- #define ADC_SQR3_SQ6_4 ((uint32_t)0x20000000)
- #define ADC_JSQR_JSQ1 ((uint32_t)0x00000001F)
- #define ADC_JSQR_JSQ1_0 ((uint32_t)0x00000001)
- #define ADC_JSQR_JSQ1_1 ((uint32_t)0x00000002)
- #define ADC_JSQR_JSQ1_2 ((uint32_t)0x00000004)
- #define ADC_JSQR_JSQ1_3 ((uint32_t)0x00000008)
- #define ADC_JSQR_JSQ1_4 ((uint32_t)0x00000010)
- #define ADC_JSQR_JSQ2 ((uint32_t)0x0000003E0)
- #define ADC_JSQR_JSQ2_0 ((uint32_t)0x00000020)
- #define ADC_JSQR_JSQ2_1 ((uint32_t)0x00000040)
- #define ADC_JSQR_JSQ2_2 ((uint32_t)0x00000080)
- #define ADC_JSQR_JSQ2_3 ((uint32_t)0x00000100)
- #define ADC_JSQR_JSQ2_4 ((uint32_t)0x00000200)
- #define ADC_JSQR_JSQ3 ((uint32_t)0x00007C00)
- #define ADC_JSQR_JSQ3_0 ((uint32_t)0x00000400)
- #define ADC_JSQR_JSQ3_1 ((uint32_t)0x00000800)
- #define ADC_JSQR_JSQ3_2 ((uint32_t)0x00001000)
- #define ADC_JSQR_JSQ3_3 ((uint32_t)0x00002000)
- #define ADC_JSQR_JSQ3_4 ((uint32_t)0x00004000)
- #define ADC_JSQR_JSQ4 ((uint32_t)0x000F8000)
- #define ADC_JSQR_JSQ4_0 ((uint32_t)0x00008000)
- #define ADC_JSQR_JSQ4_1 ((uint32_t)0x00010000)
- #define ADC_JSQR_JSQ4_2 ((uint32_t)0x00020000)
- #define ADC_JSQR_JSQ4_3 ((uint32_t)0x00040000)
- #define ADC_JSQR_JSQ4_4 ((uint32_t)0x00080000)
- #define ADC_JSQR_JL ((uint32_t)0x00300000)
- #define ADC_JSQR_JL_0 ((uint32_t)0x00100000)
- #define ADC_JSQR_JL_1 ((uint32_t)0x00200000)
- #define ADC_JDR1_JDATA ((uint16_t)0xFFFF)
- #define ADC_JDR2_JDATA ((uint16_t)0xFFFF)
- #define ADC_JDR3_JDATA ((uint16_t)0xFFFF)
- #define ADC_JDR4_JDATA ((uint16_t)0xFFFF)
- #define ADC_DR_DATA ((uint32_t)0x0000FFFF)
- #define ADC_DR_ADC2DATA ((uint32_t)0xFFFF0000)
- #define ADC_CSR_AWD1 ((uint32_t)0x00000001)
- #define ADC_CSR_EOC1 ((uint32_t)0x00000002)
- #define ADC_CSR_JEOC1 ((uint32_t)0x00000004)
- #define ADC_CSR_JSTRT1 ((uint32_t)0x00000008)
- #define ADC_CSR_STRT1 ((uint32_t)0x00000010)
- #define ADC_CSR_OVR1 ((uint32_t)0x00000020)
- #define ADC_CSR_AWD2 ((uint32_t)0x00000100)
- #define ADC_CSR_EOC2 ((uint32_t)0x00000200)
- #define ADC_CSR_JEOC2 ((uint32_t)0x00000400)
- #define ADC_CSR_JSTRT2 ((uint32_t)0x00000800)
- #define ADC_CSR_STRT2 ((uint32_t)0x00001000)
- #define ADC_CSR_OVR2 ((uint32_t)0x00002000)
- #define ADC_CSR_AWD3 ((uint32_t)0x00010000)

- #define ADC_CSR_EOC3 ((uint32_t)0x00020000)
- #define ADC_CSR_JEOC3 ((uint32_t)0x00040000)
- #define ADC_CSR_JSTRT3 ((uint32_t)0x00080000)
- #define ADC_CSR_STRT3 ((uint32_t)0x00100000)
- #define ADC_CSR_OVR3 ((uint32_t)0x02000000)
- #define ADC_CSR_DOVRI ADC_CSR_OVR1
- #define ADC_CSR_DOVRS ADC_CSR_OVR2
- #define ADC_CSR_DOVRC ADC_CSR_OVR3
- #define ADC_CCR_MULTI ((uint32_t)0x0000001F)
- #define ADC_CCR_MULTI_0 ((uint32_t)0x00000001)
- #define ADC_CCR_MULTI_1 ((uint32_t)0x00000002)
- #define ADC_CCR_MULTI_2 ((uint32_t)0x00000004)
- #define ADC_CCR_MULTI_3 ((uint32_t)0x00000008)
- #define ADC_CCR_MULTI_4 ((uint32_t)0x00000010)
- #define ADC_CCR_DELAY ((uint32_t)0x00000F00)
- #define ADC_CCR_DELAY_0 ((uint32_t)0x00000100)
- #define ADC_CCR_DELAY_1 ((uint32_t)0x00000200)
- #define ADC_CCR_DELAY_2 ((uint32_t)0x00000400)
- #define ADC_CCR_DELAY_3 ((uint32_t)0x00000800)
- #define ADC_CCR_DDS ((uint32_t)0x00002000)
- #define ADC_CCR_DMA ((uint32_t)0x0000C000)
- #define ADC_CCR_DMA_0 ((uint32_t)0x00004000)
- #define ADC_CCR_DMA_1 ((uint32_t)0x00008000)
- #define ADC_CCR_ADCPRE ((uint32_t)0x00030000)
- #define ADC_CCR_ADCPRE_0 ((uint32_t)0x00010000)
- #define ADC_CCR_ADCPRE_1 ((uint32_t)0x00020000)
- #define ADC_CCR_VBATE ((uint32_t)0x00400000)
- #define ADC_CCR_TSVREFE ((uint32_t)0x00800000)
- #define ADC_CDR_DATA1 ((uint32_t)0x0000FFFF)
- #define ADC_CDR_DATA2 ((uint32_t)0xFFFF0000)
- #define CAN_MCR_INRQ ((uint16_t)0x0001)
- #define CAN_MCR_SLEEP ((uint16_t)0x0002)
- #define CAN_MCR_TXFP ((uint16_t)0x0004)
- #define CAN_MCR_RFLM ((uint16_t)0x0008)
- #define CAN_MCR_NART ((uint16_t)0x0010)
- #define CAN_MCR_AWUM ((uint16_t)0x0020)
- #define CAN_MCR_ABOM ((uint16_t)0x0040)
- #define CAN_MCR_TTCM ((uint16_t)0x0080)
- #define CAN_MCR_RESET ((uint16_t)0x8000)
- #define CAN_MSR_INAK ((uint16_t)0x0001)
- #define CAN_MSR_SLAK ((uint16_t)0x0002)
- #define CAN_MSR_ERRI ((uint16_t)0x0004)
- #define CAN_MSR_WKUI ((uint16_t)0x0008)
- #define CAN_MSR_SLAKI ((uint16_t)0x0010)
- #define CAN_MSR_TXM ((uint16_t)0x0100)
- #define CAN_MSR_RXM ((uint16_t)0x0200)
- #define CAN_MSR_SAMP ((uint16_t)0x0400)
- #define CAN_MSR_RX ((uint16_t)0x0800)
- #define CAN_TSR_RQCP0 ((uint32_t)0x00000001)
- #define CAN_TSR_TXOK0 ((uint32_t)0x00000002)
- #define CAN_TSR_ALST0 ((uint32_t)0x00000004)
- #define CAN_TSR_TERR0 ((uint32_t)0x00000008)
- #define CAN_TSR_ABRQ0 ((uint32_t)0x00000080)
- #define CAN_TSR_RQCP1 ((uint32_t)0x00000100)
- #define CAN_TSR_TXOK1 ((uint32_t)0x00000200)

- #define CAN_TSR_ALST1 ((uint32_t)0x00000400)
- #define CAN_TSR_TERR1 ((uint32_t)0x00000800)
- #define CAN_TSR_ABRQ1 ((uint32_t)0x00008000)
- #define CAN_TSR_RQCP2 ((uint32_t)0x00010000)
- #define CAN_TSR_TXOK2 ((uint32_t)0x00020000)
- #define CAN_TSR_ALST2 ((uint32_t)0x00040000)
- #define CAN_TSR_TERR2 ((uint32_t)0x00080000)
- #define CAN_TSR_ABRQ2 ((uint32_t)0x00800000)
- #define CAN_TSR_CODE ((uint32_t)0x03000000)
- #define CAN_TSR_TME ((uint32_t)0x1C000000)
- #define CAN_TSR_TME0 ((uint32_t)0x04000000)
- #define CAN_TSR_TME1 ((uint32_t)0x08000000)
- #define CAN_TSR_TME2 ((uint32_t)0x10000000)
- #define CAN_TSR_LOW ((uint32_t)0xE0000000)
- #define CAN_TSR_LOW0 ((uint32_t)0x20000000)
- #define CAN_TSR_LOW1 ((uint32_t)0x40000000)
- #define CAN_TSR_LOW2 ((uint32_t)0x80000000)
- #define CAN_RF0R_FMP0 ((uint8_t)0x03)
- #define CAN_RF0R_FULL0 ((uint8_t)0x08)
- #define CAN_RF0R_FOVR0 ((uint8_t)0x10)
- #define CAN_RF0R_RFOM0 ((uint8_t)0x20)
- #define CAN_RF1R_FMP1 ((uint8_t)0x03)
- #define CAN_RF1R_FULL1 ((uint8_t)0x08)
- #define CAN_RF1R_FOVR1 ((uint8_t)0x10)
- #define CAN_RF1R_RFOM1 ((uint8_t)0x20)
- #define CAN_IER_TMEIE ((uint32_t)0x00000001)
- #define CAN_IER_FMPIE0 ((uint32_t)0x00000002)
- #define CAN_IER_FFIE0 ((uint32_t)0x00000004)
- #define CAN_IER_FOVIE0 ((uint32_t)0x00000008)
- #define CAN_IER_FMPIE1 ((uint32_t)0x00000010)
- #define CAN_IER_FFIE1 ((uint32_t)0x00000020)
- #define CAN_IER_FOVIE1 ((uint32_t)0x00000040)
- #define CAN_IER_EWGIE ((uint32_t)0x00000100)
- #define CAN_IER_EPVIE ((uint32_t)0x00000200)
- #define CAN_IER_BOFIE ((uint32_t)0x00000400)
- #define CAN_IER_LECIE ((uint32_t)0x00000800)
- #define CAN_IER_ERRIE ((uint32_t)0x00008000)
- #define CAN_IER_WKUIE ((uint32_t)0x00010000)
- #define CAN_IER_SLKIE ((uint32_t)0x00020000)
- #define CAN_ESR_EWGF ((uint32_t)0x00000001)
- #define CAN_ESR_EPVF ((uint32_t)0x00000002)
- #define CAN_ESR_BOFF ((uint32_t)0x00000004)
- #define CAN_ESR_LEC ((uint32_t)0x00000070)
- #define CAN_ESR_LEC_0 ((uint32_t)0x00000010)
- #define CAN_ESR_LEC_1 ((uint32_t)0x00000020)
- #define CAN_ESR_LEC_2 ((uint32_t)0x00000040)
- #define CAN_ESR_TEC ((uint32_t)0x00FF0000)
- #define CAN_ESR_REC ((uint32_t)0xFF000000)
- #define CAN_BTR_BRP ((uint32_t)0x000003FF)
- #define CAN_BTR_TS1 ((uint32_t)0x000F0000)
- #define CAN_BTR_TS2 ((uint32_t)0x00700000)
- #define CAN_BTR_SJW ((uint32_t)0x03000000)
- #define CAN_BTR_LBKM ((uint32_t)0x40000000)
- #define CAN_BTR_SILM ((uint32_t)0x80000000)
- #define CAN_TI0R_TXRQ ((uint32_t)0x00000001)

```
• #define CAN_TI0R_RTR ((uint32_t)0x00000002)
• #define CAN_TI0R_IDE ((uint32_t)0x00000004)
• #define CAN_TI0R_EXID ((uint32_t)0x001FFFF8)
• #define CAN_TI0R_STID ((uint32_t)0xFFE00000)
• #define CAN_TDT0R_DLC ((uint32_t)0x0000000F)
• #define CAN_TDT0R_TGT ((uint32_t)0x00000100)
• #define CAN_TDT0R_TIME ((uint32_t)0xFFFF0000)
• #define CAN_TDL0R_DATA0 ((uint32_t)0x000000FF)
• #define CAN_TDL0R_DATA1 ((uint32_t)0x0000FF00)
• #define CAN_TDL0R_DATA2 ((uint32_t)0x00FF0000)
• #define CAN_TDL0R_DATA3 ((uint32_t)0xFF000000)
• #define CAN_TDH0R_DATA4 ((uint32_t)0x000000FF)
• #define CAN_TDH0R_DATA5 ((uint32_t)0x0000FF00)
• #define CAN_TDH0R_DATA6 ((uint32_t)0x00FF0000)
• #define CAN_TDH0R_DATA7 ((uint32_t)0xFF000000)
• #define CAN_TI1R_TXRQ ((uint32_t)0x00000001)
• #define CAN_TI1R_RTR ((uint32_t)0x00000002)
• #define CAN_TI1R_IDE ((uint32_t)0x00000004)
• #define CAN_TI1R_EXID ((uint32_t)0x001FFFF8)
• #define CAN_TI1R_STID ((uint32_t)0xFFE00000)
• #define CAN_TDT1R_DLC ((uint32_t)0x0000000F)
• #define CAN_TDT1R_TGT ((uint32_t)0x00000100)
• #define CAN_TDT1R_TIME ((uint32_t)0xFFFF0000)
• #define CAN_TDL1R_DATA0 ((uint32_t)0x000000FF)
• #define CAN_TDL1R_DATA1 ((uint32_t)0x0000FF00)
• #define CAN_TDL1R_DATA2 ((uint32_t)0x00FF0000)
• #define CAN_TDL1R_DATA3 ((uint32_t)0xFF000000)
• #define CAN_TDH1R_DATA4 ((uint32_t)0x000000FF)
• #define CAN_TDH1R_DATA5 ((uint32_t)0x0000FF00)
• #define CAN_TDH1R_DATA6 ((uint32_t)0x00FF0000)
• #define CAN_TDH1R_DATA7 ((uint32_t)0xFF000000)
• #define CAN_TI2R_TXRQ ((uint32_t)0x00000001)
• #define CAN_TI2R_RTR ((uint32_t)0x00000002)
• #define CAN_TI2R_IDE ((uint32_t)0x00000004)
• #define CAN_TI2R_EXID ((uint32_t)0x001FFFF8)
• #define CAN_TI2R_STID ((uint32_t)0xFFE00000)
• #define CAN_TDT2R_DLC ((uint32_t)0x0000000F)
• #define CAN_TDT2R_TGT ((uint32_t)0x00000100)
• #define CAN_TDT2R_TIME ((uint32_t)0xFFFF0000)
• #define CAN_TDL2R_DATA0 ((uint32_t)0x000000FF)
• #define CAN_TDL2R_DATA1 ((uint32_t)0x0000FF00)
• #define CAN_TDL2R_DATA2 ((uint32_t)0x00FF0000)
• #define CAN_TDL2R_DATA3 ((uint32_t)0xFF000000)
• #define CAN_TDH2R_DATA4 ((uint32_t)0x000000FF)
• #define CAN_TDH2R_DATA5 ((uint32_t)0x0000FF00)
• #define CAN_TDH2R_DATA6 ((uint32_t)0x00FF0000)
• #define CAN_TDH2R_DATA7 ((uint32_t)0xFF000000)
• #define CAN_RI0R_RTR ((uint32_t)0x00000002)
• #define CAN_RI0R_IDE ((uint32_t)0x00000004)
• #define CAN_RI0R_EXID ((uint32_t)0x001FFFF8)
• #define CAN_RI0R_STID ((uint32_t)0xFFE00000)
• #define CAN_RDT0R_DLC ((uint32_t)0x0000000F)
• #define CAN_RDT0R_FMI ((uint32_t)0x0000FF00)
• #define CAN_RDT0R_TIME ((uint32_t)0xFFFF0000)
• #define CAN_RDL0R_DATA0 ((uint32_t)0x000000FF)
```

- #define CAN_RDL0R_DATA1 ((uint32_t)0x0000FF00)
- #define CAN_RDL0R_DATA2 ((uint32_t)0x00FF0000)
- #define CAN_RDL0R_DATA3 ((uint32_t)0xFF000000)
- #define CAN_RDH0R_DATA4 ((uint32_t)0x000000FF)
- #define CAN_RDH0R_DATA5 ((uint32_t)0x0000FF00)
- #define CAN_RDH0R_DATA6 ((uint32_t)0x00FF0000)
- #define CAN_RDH0R_DATA7 ((uint32_t)0xFF000000)
- #define CAN_RI1R_RTR ((uint32_t)0x00000002)
- #define CAN_RI1R_IDE ((uint32_t)0x00000004)
- #define CAN_RI1R_EXID ((uint32_t)0x001FFFF8)
- #define CAN_RI1R_STID ((uint32_t)0xFFE00000)
- #define CAN_RDT1R_DLC ((uint32_t)0x0000000F)
- #define CAN_RDT1R_FMI ((uint32_t)0x0000FF00)
- #define CAN_RDT1R_TIME ((uint32_t)0xFFFF0000)
- #define CAN_RDL1R_DATA0 ((uint32_t)0x000000FF)
- #define CAN_RDL1R_DATA1 ((uint32_t)0x0000FF00)
- #define CAN_RDL1R_DATA2 ((uint32_t)0x00FF0000)
- #define CAN_RDL1R_DATA3 ((uint32_t)0xFF000000)
- #define CAN_RDH1R_DATA4 ((uint32_t)0x000000FF)
- #define CAN_RDH1R_DATA5 ((uint32_t)0x0000FF00)
- #define CAN_RDH1R_DATA6 ((uint32_t)0x00FF0000)
- #define CAN_RDH1R_DATA7 ((uint32_t)0xFF000000)
- #define CAN_FMR_INIT ((uint8_t)0x01)
- #define CAN_FM1R_FBM ((uint16_t)0x3FFF)
- #define CAN_FM1R_FBM0 ((uint16_t)0x0001)
- #define CAN_FM1R_FBM1 ((uint16_t)0x0002)
- #define CAN_FM1R_FBM2 ((uint16_t)0x0004)
- #define CAN_FM1R_FBM3 ((uint16_t)0x0008)
- #define CAN_FM1R_FBM4 ((uint16_t)0x0010)
- #define CAN_FM1R_FBM5 ((uint16_t)0x0020)
- #define CAN_FM1R_FBM6 ((uint16_t)0x0040)
- #define CAN_FM1R_FBM7 ((uint16_t)0x0080)
- #define CAN_FM1R_FBM8 ((uint16_t)0x0100)
- #define CAN_FM1R_FBM9 ((uint16_t)0x0200)
- #define CAN_FM1R_FBM10 ((uint16_t)0x0400)
- #define CAN_FM1R_FBM11 ((uint16_t)0x0800)
- #define CAN_FM1R_FBM12 ((uint16_t)0x1000)
- #define CAN_FM1R_FBM13 ((uint16_t)0x2000)
- #define CAN_FS1R_FSC ((uint16_t)0x3FFF)
- #define CAN_FS1R_FSC0 ((uint16_t)0x0001)
- #define CAN_FS1R_FSC1 ((uint16_t)0x0002)
- #define CAN_FS1R_FSC2 ((uint16_t)0x0004)
- #define CAN_FS1R_FSC3 ((uint16_t)0x0008)
- #define CAN_FS1R_FSC4 ((uint16_t)0x0010)
- #define CAN_FS1R_FSC5 ((uint16_t)0x0020)
- #define CAN_FS1R_FSC6 ((uint16_t)0x0040)
- #define CAN_FS1R_FSC7 ((uint16_t)0x0080)
- #define CAN_FS1R_FSC8 ((uint16_t)0x0100)
- #define CAN_FS1R_FSC9 ((uint16_t)0x0200)
- #define CAN_FS1R_FSC10 ((uint16_t)0x0400)
- #define CAN_FS1R_FSC11 ((uint16_t)0x0800)
- #define CAN_FS1R_FSC12 ((uint16_t)0x1000)
- #define CAN_FS1R_FSC13 ((uint16_t)0x2000)
- #define CAN_FFA1R_FFA ((uint16_t)0x3FFF)
- #define CAN_FFA1R_FFA0 ((uint16_t)0x0001)

- #define CAN_FFA1R_FFA1 ((uint16_t)0x0002)
- #define CAN_FFA1R_FFA2 ((uint16_t)0x0004)
- #define CAN_FFA1R_FFA3 ((uint16_t)0x0008)
- #define CAN_FFA1R_FFA4 ((uint16_t)0x0010)
- #define CAN_FFA1R_FFA5 ((uint16_t)0x0020)
- #define CAN_FFA1R_FFA6 ((uint16_t)0x0040)
- #define CAN_FFA1R_FFA7 ((uint16_t)0x0080)
- #define CAN_FFA1R_FFA8 ((uint16_t)0x0100)
- #define CAN_FFA1R_FFA9 ((uint16_t)0x0200)
- #define CAN_FFA1R_FFA10 ((uint16_t)0x0400)
- #define CAN_FFA1R_FFA11 ((uint16_t)0x0800)
- #define CAN_FFA1R_FFA12 ((uint16_t)0x1000)
- #define CAN_FFA1R_FFA13 ((uint16_t)0x2000)
- #define CAN_FA1R_FACT ((uint16_t)0x3FFF)
- #define CAN_FA1R_FACT0 ((uint16_t)0x0001)
- #define CAN_FA1R_FACT1 ((uint16_t)0x0002)
- #define CAN_FA1R_FACT2 ((uint16_t)0x0004)
- #define CAN_FA1R_FACT3 ((uint16_t)0x0008)
- #define CAN_FA1R_FACT4 ((uint16_t)0x0010)
- #define CAN_FA1R_FACT5 ((uint16_t)0x0020)
- #define CAN_FA1R_FACT6 ((uint16_t)0x0040)
- #define CAN_FA1R_FACT7 ((uint16_t)0x0080)
- #define CAN_FA1R_FACT8 ((uint16_t)0x0100)
- #define CAN_FA1R_FACT9 ((uint16_t)0x0200)
- #define CAN_FA1R_FACT10 ((uint16_t)0x0400)
- #define CAN_FA1R_FACT11 ((uint16_t)0x0800)
- #define CAN_FA1R_FACT12 ((uint16_t)0x1000)
- #define CAN_FA1R_FACT13 ((uint16_t)0x2000)
- #define CAN_F0R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F0R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F0R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F0R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F0R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F0R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F0R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F0R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F0R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F0R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F0R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F0R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F0R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F0R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F0R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F0R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F0R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F0R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F0R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F0R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F0R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F0R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F0R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F0R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F0R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F0R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F0R1_FB26 ((uint32_t)0x04000000)

- #define CAN_F0R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F0R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F0R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F0R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F0R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F1R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F1R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F1R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F1R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F1R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F1R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F1R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F1R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F1R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F1R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F1R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F1R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F1R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F1R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F1R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F1R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F1R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F1R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F1R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F1R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F1R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F1R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F1R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F1R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F1R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F1R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F1R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F1R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F1R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F1R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F1R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F1R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F2R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F2R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F2R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F2R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F2R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F2R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F2R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F2R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F2R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F2R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F2R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F2R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F2R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F2R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F2R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F2R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F2R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F2R1_FB17 ((uint32_t)0x00020000)

- #define CAN_F2R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F2R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F2R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F2R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F2R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F2R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F2R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F2R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F2R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F2R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F2R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F2R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F2R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F2R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F3R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F3R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F3R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F3R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F3R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F3R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F3R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F3R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F3R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F3R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F3R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F3R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F3R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F3R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F3R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F3R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F3R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F3R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F3R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F3R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F3R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F3R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F3R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F3R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F3R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F3R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F3R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F3R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F3R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F3R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F3R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F3R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F4R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F4R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F4R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F4R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F4R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F4R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F4R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F4R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F4R1_FB8 ((uint32_t)0x00000100)

- #define CAN_F4R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F4R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F4R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F4R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F4R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F4R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F4R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F4R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F4R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F4R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F4R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F4R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F4R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F4R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F4R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F4R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F4R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F4R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F4R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F4R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F4R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F4R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F4R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F5R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F5R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F5R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F5R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F5R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F5R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F5R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F5R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F5R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F5R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F5R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F5R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F5R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F5R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F5R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F5R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F5R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F5R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F5R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F5R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F5R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F5R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F5R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F5R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F5R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F5R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F5R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F5R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F5R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F5R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F5R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F5R1_FB31 ((uint32_t)0x80000000)

- #define CAN_F6R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F6R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F6R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F6R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F6R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F6R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F6R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F6R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F6R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F6R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F6R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F6R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F6R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F6R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F6R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F6R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F6R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F6R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F6R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F6R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F6R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F6R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F6R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F6R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F6R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F6R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F6R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F6R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F6R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F6R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F6R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F6R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F7R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F7R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F7R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F7R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F7R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F7R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F7R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F7R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F7R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F7R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F7R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F7R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F7R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F7R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F7R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F7R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F7R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F7R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F7R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F7R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F7R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F7R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F7R1_FB22 ((uint32_t)0x00400000)

- #define CAN_F7R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F7R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F7R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F7R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F7R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F7R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F7R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F7R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F7R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F8R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F8R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F8R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F8R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F8R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F8R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F8R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F8R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F8R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F8R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F8R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F8R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F8R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F8R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F8R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F8R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F8R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F8R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F8R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F8R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F8R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F8R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F8R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F8R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F8R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F8R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F8R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F8R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F8R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F8R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F8R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F8R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F9R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F9R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F9R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F9R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F9R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F9R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F9R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F9R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F9R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F9R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F9R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F9R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F9R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F9R1_FB13 ((uint32_t)0x00002000)

- #define CAN_F9R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F9R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F9R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F9R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F9R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F9R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F9R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F9R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F9R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F9R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F9R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F9R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F9R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F9R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F9R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F9R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F9R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F9R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F10R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F10R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F10R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F10R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F10R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F10R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F10R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F10R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F10R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F10R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F10R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F10R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F10R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F10R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F10R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F10R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F10R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F10R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F10R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F10R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F10R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F10R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F10R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F10R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F10R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F10R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F10R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F10R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F10R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F10R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F10R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F10R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F11R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F11R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F11R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F11R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F11R1_FB4 ((uint32_t)0x00000010)

- #define CAN_F11R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F11R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F11R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F11R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F11R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F11R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F11R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F11R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F11R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F11R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F11R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F11R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F11R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F11R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F11R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F11R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F11R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F11R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F11R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F11R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F11R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F11R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F11R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F11R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F11R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F11R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F11R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F12R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F12R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F12R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F12R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F12R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F12R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F12R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F12R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F12R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F12R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F12R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F12R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F12R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F12R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F12R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F12R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F12R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F12R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F12R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F12R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F12R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F12R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F12R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F12R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F12R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F12R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F12R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F12R1_FB27 ((uint32_t)0x08000000)

- #define CAN_F12R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F12R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F12R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F12R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F13R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F13R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F13R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F13R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F13R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F13R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F13R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F13R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F13R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F13R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F13R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F13R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F13R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F13R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F13R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F13R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F13R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F13R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F13R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F13R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F13R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F13R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F13R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F13R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F13R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F13R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F13R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F13R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F13R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F13R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F13R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F13R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F0R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F0R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F0R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F0R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F0R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F0R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F0R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F0R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F0R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F0R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F0R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F0R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F0R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F0R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F0R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F0R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F0R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F0R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F0R2_FB18 ((uint32_t)0x00040000)

- #define CAN_F0R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F0R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F0R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F0R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F0R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F0R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F0R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F0R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F0R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F0R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F0R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F0R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F0R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F1R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F1R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F1R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F1R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F1R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F1R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F1R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F1R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F1R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F1R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F1R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F1R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F1R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F1R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F1R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F1R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F1R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F1R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F1R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F1R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F1R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F1R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F1R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F1R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F1R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F1R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F1R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F1R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F1R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F1R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F1R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F1R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F2R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F2R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F2R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F2R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F2R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F2R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F2R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F2R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F2R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F2R2_FB9 ((uint32_t)0x00000200)

```
• #define CAN_F2R2_FB10 ((uint32_t)0x00000400)
• #define CAN_F2R2_FB11 ((uint32_t)0x00000800)
• #define CAN_F2R2_FB12 ((uint32_t)0x00001000)
• #define CAN_F2R2_FB13 ((uint32_t)0x00002000)
• #define CAN_F2R2_FB14 ((uint32_t)0x00004000)
• #define CAN_F2R2_FB15 ((uint32_t)0x00008000)
• #define CAN_F2R2_FB16 ((uint32_t)0x00010000)
• #define CAN_F2R2_FB17 ((uint32_t)0x00020000)
• #define CAN_F2R2_FB18 ((uint32_t)0x00040000)
• #define CAN_F2R2_FB19 ((uint32_t)0x00080000)
• #define CAN_F2R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F2R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F2R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F2R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F2R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F2R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F2R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F2R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F2R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F2R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F2R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F2R2_FB31 ((uint32_t)0x80000000)
• #define CAN_F3R2_FB0 ((uint32_t)0x00000001)
• #define CAN_F3R2_FB1 ((uint32_t)0x00000002)
• #define CAN_F3R2_FB2 ((uint32_t)0x00000004)
• #define CAN_F3R2_FB3 ((uint32_t)0x00000008)
• #define CAN_F3R2_FB4 ((uint32_t)0x00000010)
• #define CAN_F3R2_FB5 ((uint32_t)0x00000020)
• #define CAN_F3R2_FB6 ((uint32_t)0x00000040)
• #define CAN_F3R2_FB7 ((uint32_t)0x00000080)
• #define CAN_F3R2_FB8 ((uint32_t)0x00000100)
• #define CAN_F3R2_FB9 ((uint32_t)0x00000200)
• #define CAN_F3R2_FB10 ((uint32_t)0x00000400)
• #define CAN_F3R2_FB11 ((uint32_t)0x00000800)
• #define CAN_F3R2_FB12 ((uint32_t)0x00001000)
• #define CAN_F3R2_FB13 ((uint32_t)0x00002000)
• #define CAN_F3R2_FB14 ((uint32_t)0x00004000)
• #define CAN_F3R2_FB15 ((uint32_t)0x00008000)
• #define CAN_F3R2_FB16 ((uint32_t)0x00010000)
• #define CAN_F3R2_FB17 ((uint32_t)0x00020000)
• #define CAN_F3R2_FB18 ((uint32_t)0x00040000)
• #define CAN_F3R2_FB19 ((uint32_t)0x00080000)
• #define CAN_F3R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F3R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F3R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F3R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F3R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F3R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F3R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F3R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F3R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F3R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F3R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F3R2_FB31 ((uint32_t)0x80000000)
• #define CAN_F4R2_FB0 ((uint32_t)0x00000001)
```

- #define CAN_F4R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F4R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F4R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F4R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F4R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F4R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F4R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F4R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F4R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F4R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F4R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F4R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F4R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F4R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F4R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F4R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F4R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F4R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F4R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F4R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F4R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F4R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F4R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F4R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F4R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F4R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F4R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F4R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F4R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F4R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F4R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F5R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F5R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F5R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F5R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F5R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F5R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F5R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F5R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F5R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F5R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F5R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F5R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F5R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F5R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F5R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F5R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F5R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F5R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F5R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F5R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F5R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F5R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F5R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F5R2_FB23 ((uint32_t)0x00800000)

- #define CAN_F5R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F5R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F5R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F5R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F5R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F5R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F5R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F5R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F6R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F6R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F6R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F6R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F6R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F6R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F6R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F6R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F6R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F6R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F6R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F6R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F6R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F6R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F6R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F6R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F6R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F6R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F6R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F6R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F6R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F6R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F6R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F6R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F6R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F6R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F6R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F6R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F6R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F6R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F6R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F6R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F7R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F7R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F7R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F7R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F7R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F7R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F7R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F7R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F7R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F7R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F7R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F7R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F7R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F7R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F7R2_FB14 ((uint32_t)0x00004000)

- #define CAN_F7R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F7R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F7R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F7R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F7R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F7R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F7R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F7R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F7R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F7R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F7R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F7R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F7R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F7R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F7R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F7R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F7R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F8R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F8R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F8R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F8R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F8R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F8R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F8R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F8R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F8R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F8R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F8R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F8R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F8R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F8R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F8R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F8R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F8R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F8R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F8R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F8R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F8R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F8R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F8R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F8R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F8R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F8R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F8R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F8R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F8R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F8R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F8R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F8R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F9R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F9R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F9R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F9R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F9R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F9R2_FB5 ((uint32_t)0x00000020)

- #define CAN_F9R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F9R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F9R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F9R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F9R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F9R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F9R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F9R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F9R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F9R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F9R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F9R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F9R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F9R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F9R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F9R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F9R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F9R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F9R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F9R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F9R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F9R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F9R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F9R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F9R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F9R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F10R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F10R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F10R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F10R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F10R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F10R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F10R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F10R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F10R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F10R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F10R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F10R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F10R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F10R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F10R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F10R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F10R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F10R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F10R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F10R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F10R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F10R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F10R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F10R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F10R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F10R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F10R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F10R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F10R2_FB28 ((uint32_t)0x10000000)

- #define CAN_F10R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F10R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F10R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F11R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F11R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F11R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F11R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F11R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F11R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F11R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F11R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F11R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F11R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F11R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F11R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F11R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F11R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F11R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F11R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F11R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F11R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F11R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F11R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F11R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F11R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F11R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F11R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F11R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F11R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F11R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F11R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F11R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F11R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F11R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F11R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F12R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F12R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F12R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F12R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F12R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F12R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F12R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F12R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F12R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F12R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F12R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F12R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F12R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F12R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F12R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F12R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F12R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F12R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F12R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F12R2_FB19 ((uint32_t)0x00080000)

```
• #define CAN_F12R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F12R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F12R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F12R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F12R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F12R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F12R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F12R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F12R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F12R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F12R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F12R2_FB31 ((uint32_t)0x80000000)
• #define CAN_F13R2_FB0 ((uint32_t)0x00000001)
• #define CAN_F13R2_FB1 ((uint32_t)0x00000002)
• #define CAN_F13R2_FB2 ((uint32_t)0x00000004)
• #define CAN_F13R2_FB3 ((uint32_t)0x00000008)
• #define CAN_F13R2_FB4 ((uint32_t)0x00000010)
• #define CAN_F13R2_FB5 ((uint32_t)0x00000020)
• #define CAN_F13R2_FB6 ((uint32_t)0x00000040)
• #define CAN_F13R2_FB7 ((uint32_t)0x00000080)
• #define CAN_F13R2_FB8 ((uint32_t)0x00000100)
• #define CAN_F13R2_FB9 ((uint32_t)0x00000200)
• #define CAN_F13R2_FB10 ((uint32_t)0x00000400)
• #define CAN_F13R2_FB11 ((uint32_t)0x00000800)
• #define CAN_F13R2_FB12 ((uint32_t)0x00001000)
• #define CAN_F13R2_FB13 ((uint32_t)0x00002000)
• #define CAN_F13R2_FB14 ((uint32_t)0x00004000)
• #define CAN_F13R2_FB15 ((uint32_t)0x00008000)
• #define CAN_F13R2_FB16 ((uint32_t)0x00010000)
• #define CAN_F13R2_FB17 ((uint32_t)0x00020000)
• #define CAN_F13R2_FB18 ((uint32_t)0x00040000)
• #define CAN_F13R2_FB19 ((uint32_t)0x00080000)
• #define CAN_F13R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F13R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F13R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F13R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F13R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F13R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F13R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F13R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F13R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F13R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F13R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F13R2_FB31 ((uint32_t)0x80000000)
• #define CRC_DR_DR ((uint32_t)0xFFFFFFFF)
• #define CRC_IDR_IDR ((uint8_t)0xFF)
• #define CRC_CR_RESET ((uint8_t)0x01)
• #define CRYP_CR_ALGODIR ((uint32_t)0x00000004)
• #define CRYP_CR_ALGOMODE ((uint32_t)0x00080038)
• #define CRYP_CR_ALGOMODE_0 ((uint32_t)0x00000008)
• #define CRYP_CR_ALGOMODE_1 ((uint32_t)0x00000010)
• #define CRYP_CR_ALGOMODE_2 ((uint32_t)0x00000020)
• #define CRYP_CR_ALGOMODE_TDES_ECB ((uint32_t)0x00000000)
• #define CRYP_CR_ALGOMODE_TDES_CBC ((uint32_t)0x00000008)
• #define CRYP_CR_ALGOMODE_DES_ECB ((uint32_t)0x00000010)
```

- #define **CRYP_CR_ALGOMODE_DES_CBC** ((uint32_t)0x00000018)
- #define **CRYP_CR_ALGOMODE_AES_ECB** ((uint32_t)0x00000020)
- #define **CRYP_CR_ALGOMODE_AES_CBC** ((uint32_t)0x00000028)
- #define **CRYP_CR_ALGOMODE_AES_CTR** ((uint32_t)0x00000030)
- #define **CRYP_CR_ALGOMODE_AES_KEY** ((uint32_t)0x00000038)
- #define **CRYP_CR_DATATYPE** ((uint32_t)0x000000C0)
- #define **CRYP_CR_DATATYPE_0** ((uint32_t)0x00000040)
- #define **CRYP_CR_DATATYPE_1** ((uint32_t)0x00000080)
- #define **CRYP_CR_KEYSIZE** ((uint32_t)0x00000300)
- #define **CRYP_CR_KEYSIZE_0** ((uint32_t)0x00000100)
- #define **CRYP_CR_KEYSIZE_1** ((uint32_t)0x00000200)
- #define **CRYP_CR_FFLUSH** ((uint32_t)0x00004000)
- #define **CRYP_CR_CRYPTEN** ((uint32_t)0x00008000)
- #define **CRYP_CR_GCM_CCMPH** ((uint32_t)0x00030000)
- #define **CRYP_CR_GCM_CCMPH_0** ((uint32_t)0x00010000)
- #define **CRYP_CR_GCM_CCMPH_1** ((uint32_t)0x00020000)
- #define **CRYP_CR_ALGOMODE_3** ((uint32_t)0x00080000)
- #define **CRYP_SR_IFEM** ((uint32_t)0x00000001)
- #define **CRYP_SR_IFNF** ((uint32_t)0x00000002)
- #define **CRYP_SR_OFNE** ((uint32_t)0x00000004)
- #define **CRYP_SR_OFFU** ((uint32_t)0x00000008)
- #define **CRYP_SR_BUSY** ((uint32_t)0x00000010)
- #define **CRYP_DMACR_DIEN** ((uint32_t)0x00000001)
- #define **CRYP_DMACR_DOEN** ((uint32_t)0x00000002)
- #define **CRYP_IMSCR_INIM** ((uint32_t)0x00000001)
- #define **CRYP_IMSCR_OUTIM** ((uint32_t)0x00000002)
- #define **CRYP_RISR_OUTRIS** ((uint32_t)0x00000001)
- #define **CRYP_RISR_INRIS** ((uint32_t)0x00000002)
- #define **CRYP_MISR_INMIS** ((uint32_t)0x00000001)
- #define **CRYP_MISR_OUTMIS** ((uint32_t)0x00000002)
- #define **DAC_CR_EN1** ((uint32_t)0x00000001)
- #define **DAC_CR_BOFF1** ((uint32_t)0x00000002)
- #define **DAC_CR_TEN1** ((uint32_t)0x00000004)
- #define **DAC_CR_TSEL1** ((uint32_t)0x00000038)
- #define **DAC_CR_TSEL1_0** ((uint32_t)0x00000008)
- #define **DAC_CR_TSEL1_1** ((uint32_t)0x00000010)
- #define **DAC_CR_TSEL1_2** ((uint32_t)0x00000020)
- #define **DAC_CR_WAVE1** ((uint32_t)0x000000C0)
- #define **DAC_CR_WAVE1_0** ((uint32_t)0x00000040)
- #define **DAC_CR_WAVE1_1** ((uint32_t)0x00000080)
- #define **DAC_CR_MAMP1** ((uint32_t)0x00000F00)
- #define **DAC_CR_MAMP1_0** ((uint32_t)0x00000100)
- #define **DAC_CR_MAMP1_1** ((uint32_t)0x00000200)
- #define **DAC_CR_MAMP1_2** ((uint32_t)0x00000400)
- #define **DAC_CR_MAMP1_3** ((uint32_t)0x00000800)
- #define **DAC_CR_DMAEN1** ((uint32_t)0x00001000)
- #define **DAC_CR_DMAUDRIE1** ((uint32_t)0x00002000)
- #define **DAC_CR_EN2** ((uint32_t)0x00010000)
- #define **DAC_CR_BOFF2** ((uint32_t)0x00020000)
- #define **DAC_CR_TEN2** ((uint32_t)0x00040000)
- #define **DAC_CR_TSEL2** ((uint32_t)0x00380000)
- #define **DAC_CR_TSEL2_0** ((uint32_t)0x00080000)
- #define **DAC_CR_TSEL2_1** ((uint32_t)0x00100000)
- #define **DAC_CR_TSEL2_2** ((uint32_t)0x00200000)
- #define **DAC_CR_WAVE2** ((uint32_t)0x00C00000)

- #define **DAC_CR_WAVE2_0** ((uint32_t)0x00400000)
- #define **DAC_CR_WAVE2_1** ((uint32_t)0x00800000)
- #define **DAC_CR_MAMP2** ((uint32_t)0xF0000000)
- #define **DAC_CR_MAMP2_0** ((uint32_t)0x01000000)
- #define **DAC_CR_MAMP2_1** ((uint32_t)0x02000000)
- #define **DAC_CR_MAMP2_2** ((uint32_t)0x04000000)
- #define **DAC_CR_MAMP2_3** ((uint32_t)0x08000000)
- #define **DAC_CR_DMAEN2** ((uint32_t)0x10000000)
- #define **DAC_CR_DMAUDRIE2** ((uint32_t)0x20000000U)
- #define **DAC_SWTRIGR_SWTRIG1** ((uint8_t)0x01)
- #define **DAC_SWTRIGR_SWTRIG2** ((uint8_t)0x02)
- #define **DAC_DHR12R1_DACC1DHR** ((uint16_t)0xFFFF)
- #define **DAC_DHR12L1_DACC1DHR** ((uint16_t)0xFFFF0)
- #define **DAC_DHR8R1_DACC1DHR** ((uint8_t)0xFF)
- #define **DAC_DHR12R2_DACC2DHR** ((uint16_t)0xFFFF)
- #define **DAC_DHR12L2_DACC2DHR** ((uint16_t)0xFFFF0)
- #define **DAC_DHR8R2_DACC2DHR** ((uint8_t)0xFF)
- #define **DAC_DHR12RD_DACC1DHR** ((uint32_t)0x00000FFF)
- #define **DAC_DHR12RD_DACC2DHR** ((uint32_t)0x0FFF0000)
- #define **DAC_DHR12LD_DACC1DHR** ((uint32_t)0x0000FFF0)
- #define **DAC_DHR12LD_DACC2DHR** ((uint32_t)0xFFFF0000)
- #define **DAC_DHR8RD_DACC1DHR** ((uint16_t)0x00FF)
- #define **DAC_DHR8RD_DACC2DHR** ((uint16_t)0xFF00)
- #define **DAC_DOR1_DACC1DOR** ((uint16_t)0x0FFF)
- #define **DAC_DOR2_DACC2DOR** ((uint16_t)0xFFFF)
- #define **DAC_SR_DMAUDR1** ((uint32_t)0x00002000)
- #define **DAC_SR_DMAUDR2** ((uint32_t)0x20000000)
- #define **DCMI_CR_CAPTURE** ((uint32_t)0x00000001)
- #define **DCMI_CR_CM** ((uint32_t)0x00000002)
- #define **DCMI_CR_CROP** ((uint32_t)0x00000004)
- #define **DCMI_CR_JPEG** ((uint32_t)0x00000008)
- #define **DCMI_CR_ESS** ((uint32_t)0x00000010)
- #define **DCMI_CR_PCKPOL** ((uint32_t)0x00000020)
- #define **DCMI_CR_HSPOL** ((uint32_t)0x00000040)
- #define **DCMI_CR_VSPOL** ((uint32_t)0x00000080)
- #define **DCMI_CR_FCRC_0** ((uint32_t)0x00000100)
- #define **DCMI_CR_FCRC_1** ((uint32_t)0x00000200)
- #define **DCMI_CR_EDM_0** ((uint32_t)0x00000400)
- #define **DCMI_CR_EDM_1** ((uint32_t)0x00000800)
- #define **DCMI_CR_CRE** ((uint32_t)0x00001000)
- #define **DCMI_CR_ENABLE** ((uint32_t)0x00004000)
- #define **DCMI_SR_HSYNC** ((uint32_t)0x00000001)
- #define **DCMI_SR_VSYNC** ((uint32_t)0x00000002)
- #define **DCMI_SR_FNE** ((uint32_t)0x00000004)
- #define **DCMI_RIS_FRAME_RIS** ((uint32_t)0x00000001)
- #define **DCMI_RIS_OVR_RIS** ((uint32_t)0x00000002)
- #define **DCMI_RIS_ERR_RIS** ((uint32_t)0x00000004)
- #define **DCMI_RIS_VSYNC_RIS** ((uint32_t)0x00000008)
- #define **DCMI_RIS_LINE_RIS** ((uint32_t)0x00000010)
- #define **DCMI_RISR_FRAME_RIS** DCMI_RIS_FRAME_RIS
- #define **DCMI_RISR_OVR_RIS** DCMI_RIS_OVR_RIS
- #define **DCMI_RISR_ERR_RIS** DCMI_RIS_ERR_RIS
- #define **DCMI_RISR_VSYNC_RIS** DCMI_RIS_VSYNC_RIS
- #define **DCMI_RISR_LINE_RIS** DCMI_RIS_LINE_RIS
- #define **DCMI_RISR_OVF_RIS** DCMI_RIS_OVR_RIS

- #define **DCMI_IER_FRAME_IE** ((uint32_t)0x00000001)
- #define **DCMI_IER_OVR_IE** ((uint32_t)0x00000002)
- #define **DCMI_IER_ERR_IE** ((uint32_t)0x00000004)
- #define **DCMI_IER_VSYNC_IE** ((uint32_t)0x00000008)
- #define **DCMI_IER_LINE_IE** ((uint32_t)0x00000010)
- #define **DCMI_IER_OVF_IE** DCMI_IER_OVR_IE
- #define **DCMI_MIS_FRAME_MIS** ((uint32_t)0x00000001)
- #define **DCMI_MIS_OVR_MIS** ((uint32_t)0x00000002)
- #define **DCMI_MIS_ERR_MIS** ((uint32_t)0x00000004)
- #define **DCMI_MIS_VSYNC_MIS** ((uint32_t)0x00000008)
- #define **DCMI_MIS_LINE_MIS** ((uint32_t)0x00000010)
- #define **DCMI_MISR_FRAME_MIS** DCMI_MIS_FRAME_MIS
- #define **DCMI_MISR_OVF_MIS** DCMI_MIS_OVR_MIS
- #define **DCMI_MISR_ERR_MIS** DCMI_MIS_ERR_MIS
- #define **DCMI_MISR_VSYNC_MIS** DCMI_MIS_VSYNC_MIS
- #define **DCMI_MISR_LINE_MIS** DCMI_MIS_LINE_MIS
- #define **DCMI_ICR_FRAME_ISC** ((uint32_t)0x00000001)
- #define **DCMI_ICR_OVR_ISC** ((uint32_t)0x00000002)
- #define **DCMI_ICR_ERR_ISC** ((uint32_t)0x00000004)
- #define **DCMI_ICR_VSYNC_ISC** ((uint32_t)0x00000008)
- #define **DCMI_ICR_LINE_ISC** ((uint32_t)0x00000010)
- #define **DCMI_ICR_OVF_ISC** DCMI_ICR_OVR_ISC
- #define **DCMI_ESCR_FSC** ((uint32_t)0x000000FF)
- #define **DCMI_ESCR_LSC** ((uint32_t)0x0000FF00)
- #define **DCMI_ESCR_LEC** ((uint32_t)0x00FF0000)
- #define **DCMI_ESCR_FEC** ((uint32_t)0xFF000000)
- #define **DCMI_ESUR_FSU** ((uint32_t)0x000000FF)
- #define **DCMI_ESUR_LSU** ((uint32_t)0x0000FF00)
- #define **DCMI_ESUR_LEU** ((uint32_t)0x00FF0000)
- #define **DCMI_ESUR_FEU** ((uint32_t)0xFF000000)
- #define **DCMI_CWSTRT_HOFFCNT** ((uint32_t)0x00003FFF)
- #define **DCMI_CWSTRT_VST** ((uint32_t)0x1FFF0000)
- #define **DCMI_CWSIZE_CAPCNT** ((uint32_t)0x00003FFF)
- #define **DCMI_CWSIZE_VLINE** ((uint32_t)0x3FFF0000)
- #define **DCMI_DR_BYTE0** ((uint32_t)0x000000FF)
- #define **DCMI_DR_BYTE1** ((uint32_t)0x0000FF00)
- #define **DCMI_DR_BYTE2** ((uint32_t)0x00FF0000)
- #define **DCMI_DR_BYTE3** ((uint32_t)0xFF000000)
- #define **DFSDM_CHCFGR1_DFSDMEN** ((uint32_t)0x80000000)
- #define **DFSDM_CHCFGR1_CKOUTSRC** ((uint32_t)0x40000000)
- #define **DFSDM_CHCFGR1_CKOUTDIV** ((uint32_t)0x00FF0000)
- #define **DFSDM_CHCFGR1_DATPACK** ((uint32_t)0x0000C000)
- #define **DFSDM_CHCFGR1_DATPACK_1** ((uint32_t)0x00008000)
- #define **DFSDM_CHCFGR1_DATPACK_0** ((uint32_t)0x00004000)
- #define **DFSDM_CHCFGR1_DATMPX** ((uint32_t)0x00003000)
- #define **DFSDM_CHCFGR1_DATMPX_1** ((uint32_t)0x00002000)
- #define **DFSDM_CHCFGR1_DATMPX_0** ((uint32_t)0x00001000)
- #define **DFSDM_CHCFGR1_CHINSEL** ((uint32_t)0x00000100)
- #define **DFSDM_CHCFGR1_CHEN** ((uint32_t)0x00000080)
- #define **DFSDM_CHCFGR1_CKABEN** ((uint32_t)0x00000040)
- #define **DFSDM_CHCFGR1_SCDEN** ((uint32_t)0x00000020)
- #define **DFSDM_CHCFGR1_SPICKSEL** ((uint32_t)0x0000000C)
- #define **DFSDM_CHCFGR1_SPICKSEL_1** ((uint32_t)0x00000008)
- #define **DFSDM_CHCFGR1_SPICKSEL_0** ((uint32_t)0x00000004)
- #define **DFSDM_CHCFGR1_SITP** ((uint32_t)0x00000003)

- #define DFSDM_CHCFGR1_SITP_1 ((uint32_t)0x00000002)
- #define DFSDM_CHCFGR1_SITP_0 ((uint32_t)0x00000001)
- #define DFSDM_CHCFGR2_OFFSET ((uint32_t)0xFFFFFFF00)
- #define DFSDM_CHCFGR2_DTRBS ((uint32_t)0x000000F8)
- #define DFSDM_CHAWSCDR_AWFORD ((uint32_t)0x00C00000)
- #define DFSDM_CHAWSCDR_AWFORD_1 ((uint32_t)0x00800000)
- #define DFSDM_CHAWSCDR_AWFORD_0 ((uint32_t)0x00400000)
- #define DFSDM_CHAWSCDR_AWFOSR ((uint32_t)0x001F0000)
- #define DFSDM_CHAWSCDR_BKSCD ((uint32_t)0x0000F000)
- #define DFSDM_CHAWSCDR_SCDT ((uint32_t)0x000000FF)
- #define DFSDM_CHWDATR_WDATA ((uint32_t)0x0000FFFF)
- #define DFSDM_CHDATINR_INDAT0 ((uint32_t)0x0000FFFF)
- #define DFSDM_CHDATINR_INDAT1 ((uint32_t)0xFFFF0000)
- #define DFSDM_FLTCR1_AWFSEL ((uint32_t)0x40000000)
- #define DFSDM_FLTCR1_FAST ((uint32_t)0x20000000)
- #define DFSDM_FLTCR1_RCH ((uint32_t)0x07000000)
- #define DFSDM_FLTCR1_RDMAEN ((uint32_t)0x00200000)
- #define DFSDM_FLTCR1_RSYNC ((uint32_t)0x00080000)
- #define DFSDM_FLTCR1_RCONT ((uint32_t)0x00040000)
- #define DFSDM_FLTCR1_RSWSTART ((uint32_t)0x00020000)
- #define DFSDM_FLTCR1_JEXTEN ((uint32_t)0x00006000)
- #define DFSDM_FLTCR1_JEXTEN_1 ((uint32_t)0x00004000)
- #define DFSDM_FLTCR1_JEXTEN_0 ((uint32_t)0x00002000)
- #define DFSDM_FLTCR1_JEXTSEL ((uint32_t)0x00000700)
- #define DFSDM_FLTCR1_JEXTSEL_2 ((uint32_t)0x00000400)
- #define DFSDM_FLTCR1_JEXTSEL_1 ((uint32_t)0x00000200)
- #define DFSDM_FLTCR1_JEXTSEL_0 ((uint32_t)0x00000100)
- #define DFSDM_FLTCR1_JDMAEN ((uint32_t)0x00000020)
- #define DFSDM_FLTCR1_JSCAN ((uint32_t)0x00000010)
- #define DFSDM_FLTCR1_JSYNC ((uint32_t)0x00000008)
- #define DFSDM_FLTCR1_JSWSTART ((uint32_t)0x00000002)
- #define DFSDM_FLTCR1_DFEN ((uint32_t)0x00000001)
- #define DFSDM_FLTCR2_AWDCH ((uint32_t)0x000F0000)
- #define DFSDM_FLTCR2_EXCH ((uint32_t)0x00000F00)
- #define DFSDM_FLTCR2_CKABIE ((uint32_t)0x00000040)
- #define DFSDM_FLTCR2_SCDIE ((uint32_t)0x00000020)
- #define DFSDM_FLTCR2_AWDIE ((uint32_t)0x00000010)
- #define DFSDM_FLTCR2_ROVRIE ((uint32_t)0x00000008)
- #define DFSDM_FLTCR2_JOVRIE ((uint32_t)0x00000004)
- #define DFSDM_FLTCR2_REOCIE ((uint32_t)0x00000002)
- #define DFSDM_FLTCR2_JEOCIE ((uint32_t)0x00000001)
- #define DFSDM_FLTISR_SCDF ((uint32_t)0x0F000000)
- #define DFSDM_FLTISR_CKABF ((uint32_t)0x000F0000)
- #define DFSDM_FLTISR_RCIP ((uint32_t)0x00004000)
- #define DFSDM_FLTISR_JCIP ((uint32_t)0x00002000)
- #define DFSDM_FLTISR_AWDF ((uint32_t)0x00000010)
- #define DFSDM_FLTISR_ROVRF ((uint32_t)0x00000008)
- #define DFSDM_FLTISR_JOVRF ((uint32_t)0x00000004)
- #define DFSDM_FLTISR_REOFC ((uint32_t)0x00000002)
- #define DFSDM_FLTISR_JEOCF ((uint32_t)0x00000001)
- #define DFSDM_FLTICR_CLRSCSDF ((uint32_t)0x0F000000)
- #define DFSDM_FLTICR_CLRKABF ((uint32_t)0x000F0000)
- #define DFSDM_FLTICR_CLRRORVRF ((uint32_t)0x00000008)
- #define DFSDM_FLTICR_CLRJOVRF ((uint32_t)0x00000004)
- #define DFSDM_FLTJCHGR_JCHG ((uint32_t)0x000000FF)

- #define DFSDM_FLTFCR_FORD ((uint32_t)0xE0000000)
- #define DFSDM_FLTFCR_FORD_2 ((uint32_t)0x80000000)
- #define DFSDM_FLTFCR_FORD_1 ((uint32_t)0x40000000)
- #define DFSDM_FLTFCR_FORD_0 ((uint32_t)0x20000000)
- #define DFSDM_FLTFCR_FOSR ((uint32_t)0x03FF0000)
- #define DFSDM_FLTFCR_IOSR ((uint32_t)0x000000FF)
- #define DFSDM_FLTJDATAR_JDATA ((uint32_t)0xFFFFFFF0)
- #define DFSDM_FLTJDATAR_JDATACH ((uint32_t)0x00000007)
- #define DFSDM_FLTRDATAR_RDATA ((uint32_t)0xFFFFFFF0)
- #define DFSDM_FLTRDATAR_RPEND ((uint32_t)0x00000010)
- #define DFSDM_FLTRDATAR_RDATAACH ((uint32_t)0x00000007)
- #define DFSDM_FLTAWHTR_AWHT ((uint32_t)0xFFFFFFF0)
- #define DFSDM_FLTAWHTR_BKAWH ((uint32_t)0x0000000F)
- #define DFSDM_FLTAWLTR_AWLT ((uint32_t)0xFFFFFFF0)
- #define DFSDM_FLTAWLTR_BKAWL ((uint32_t)0x0000000F)
- #define DFSDM_FLTAWSR_AWHTF ((uint32_t)0x00000F00)
- #define DFSDM_FLTAWSR_AWLTF ((uint32_t)0x0000000F)
- #define DFSDM_FLTAWCRR_CLRAWHTF ((uint32_t)0x00000F00)
- #define DFSDM_FLTAWCRR_CLRAWLTF ((uint32_t)0x0000000F)
- #define DFSDM_FLTEXMAX_EXMAX ((uint32_t)0xFFFFFFF0)
- #define DFSDM_FLTEXMAX_EXMAXCH ((uint32_t)0x00000007)
- #define DFSDM_FLTEXMIN_EXMIN ((uint32_t)0xFFFFFFF0)
- #define DFSDM_FLTEXMIN_EXMINCH ((uint32_t)0x00000007)
- #define DFSDM_FLTCNVTIMR_CNVCT ((uint32_t)0xFFFFFFF0)
- #define DMA_SxCR_CHSEL ((uint32_t)0x0E000000)
- #define DMA_SxCR_CHSEL_0 ((uint32_t)0x02000000)
- #define DMA_SxCR_CHSEL_1 ((uint32_t)0x04000000)
- #define DMA_SxCR_CHSEL_2 ((uint32_t)0x08000000)
- #define DMA_SxCR_MBURST ((uint32_t)0x01800000)
- #define DMA_SxCR_MBURST_0 ((uint32_t)0x00800000)
- #define DMA_SxCR_MBURST_1 ((uint32_t)0x01000000)
- #define DMA_SxCR_PBURST ((uint32_t)0x00600000)
- #define DMA_SxCR_PBURST_0 ((uint32_t)0x00200000)
- #define DMA_SxCR_PBURST_1 ((uint32_t)0x00400000)
- #define DMA_SxCR_ACK ((uint32_t)0x00100000)
- #define DMA_SxCR_CT ((uint32_t)0x00080000)
- #define DMA_SxCR_DBM ((uint32_t)0x00040000)
- #define DMA_SxCR_PL ((uint32_t)0x00030000)
- #define DMA_SxCR_PL_0 ((uint32_t)0x00010000)
- #define DMA_SxCR_PL_1 ((uint32_t)0x00020000)
- #define DMA_SxCR_PINCOS ((uint32_t)0x000008000)
- #define DMA_SxCR_MSIZE ((uint32_t)0x00006000)
- #define DMA_SxCR_MSIZE_0 ((uint32_t)0x00002000)
- #define DMA_SxCR_MSIZE_1 ((uint32_t)0x00004000)
- #define DMA_SxCR_PSIZE ((uint32_t)0x00001800)
- #define DMA_SxCR_PSIZE_0 ((uint32_t)0x00000800)
- #define DMA_SxCR_PSIZE_1 ((uint32_t)0x00001000)
- #define DMA_SxCR_MINC ((uint32_t)0x00000400)
- #define DMA_SxCR_PINC ((uint32_t)0x00000200)
- #define DMA_SxCR_CIRC ((uint32_t)0x00000100)
- #define DMA_SxCR_DIR ((uint32_t)0x000000C0)
- #define DMA_SxCR_DIR_0 ((uint32_t)0x00000040)
- #define DMA_SxCR_DIR_1 ((uint32_t)0x00000080)
- #define DMA_SxCR_PFCTRL ((uint32_t)0x00000020)
- #define DMA_SxCR_TCIE ((uint32_t)0x00000010)

- #define **DMA_SxCR_HTIE** ((uint32_t)0x00000008)
- #define **DMA_SxCR_TEIE** ((uint32_t)0x00000004)
- #define **DMA_SxCR_DMEIE** ((uint32_t)0x00000002)
- #define **DMA_SxCR_EN** ((uint32_t)0x00000001)
- #define **DMA_SxNDT** ((uint32_t)0x0000FFFF)
- #define **DMA_SxNDT_0** ((uint32_t)0x00000001)
- #define **DMA_SxNDT_1** ((uint32_t)0x00000002)
- #define **DMA_SxNDT_2** ((uint32_t)0x00000004)
- #define **DMA_SxNDT_3** ((uint32_t)0x00000008)
- #define **DMA_SxNDT_4** ((uint32_t)0x00000010)
- #define **DMA_SxNDT_5** ((uint32_t)0x00000020)
- #define **DMA_SxNDT_6** ((uint32_t)0x00000040)
- #define **DMA_SxNDT_7** ((uint32_t)0x00000080)
- #define **DMA_SxNDT_8** ((uint32_t)0x00000100)
- #define **DMA_SxNDT_9** ((uint32_t)0x00000200)
- #define **DMA_SxNDT_10** ((uint32_t)0x00000400)
- #define **DMA_SxNDT_11** ((uint32_t)0x00000800)
- #define **DMA_SxNDT_12** ((uint32_t)0x00001000)
- #define **DMA_SxNDT_13** ((uint32_t)0x00002000)
- #define **DMA_SxNDT_14** ((uint32_t)0x00004000)
- #define **DMA_SxNDT_15** ((uint32_t)0x00008000)
- #define **DMA_SxFCR_FEIE** ((uint32_t)0x00000080)
- #define **DMA_SxFCR_FS** ((uint32_t)0x00000038)
- #define **DMA_SxFCR_FS_0** ((uint32_t)0x00000008)
- #define **DMA_SxFCR_FS_1** ((uint32_t)0x00000010)
- #define **DMA_SxFCR_FS_2** ((uint32_t)0x00000020)
- #define **DMA_SxFCR_DMDIS** ((uint32_t)0x00000004)
- #define **DMA_SxFCR_FTH** ((uint32_t)0x00000003)
- #define **DMA_SxFCR_FTH_0** ((uint32_t)0x00000001)
- #define **DMA_SxFCR_FTH_1** ((uint32_t)0x00000002)
- #define **DMA_LISR_TCIF3** ((uint32_t)0x08000000)
- #define **DMA_LISR_HTIF3** ((uint32_t)0x04000000)
- #define **DMA_LISR_TEIF3** ((uint32_t)0x02000000)
- #define **DMA_LISR_DMEIF3** ((uint32_t)0x01000000)
- #define **DMA_LISR_FEIF3** ((uint32_t)0x00400000)
- #define **DMA_LISR_TCIF2** ((uint32_t)0x00200000)
- #define **DMA_LISR_HTIF2** ((uint32_t)0x00100000)
- #define **DMA_LISR_TEIF2** ((uint32_t)0x00080000)
- #define **DMA_LISR_DMEIF2** ((uint32_t)0x00040000)
- #define **DMA_LISR_FEIF2** ((uint32_t)0x00010000)
- #define **DMA_LISR_TCIF1** ((uint32_t)0x00000800)
- #define **DMA_LISR_HTIF1** ((uint32_t)0x00000400)
- #define **DMA_LISR_TEIF1** ((uint32_t)0x00000200)
- #define **DMA_LISR_DMEIF1** ((uint32_t)0x00000100)
- #define **DMA_LISR_FEIF1** ((uint32_t)0x00000040)
- #define **DMA_LISR_TCIF0** ((uint32_t)0x00000020)
- #define **DMA_LISR_HTIF0** ((uint32_t)0x00000010)
- #define **DMA_LISR_TEIF0** ((uint32_t)0x00000008)
- #define **DMA_LISR_DMEIFO** ((uint32_t)0x00000004)
- #define **DMA_LISR_FEIFO** ((uint32_t)0x00000001)
- #define **DMA_HISR_TCIF7** ((uint32_t)0x08000000)
- #define **DMA_HISR_HTIF7** ((uint32_t)0x04000000)
- #define **DMA_HISR_TEIF7** ((uint32_t)0x02000000)
- #define **DMA_HISR_DMEIF7** ((uint32_t)0x01000000)
- #define **DMA_HISR_FEIF7** ((uint32_t)0x00400000)

- #define **DMA_HISR_TCIF6** ((uint32_t)0x00200000)
- #define **DMA_HISR_HTIF6** ((uint32_t)0x00100000)
- #define **DMA_HISR_TEIF6** ((uint32_t)0x00080000)
- #define **DMA_HISR_DMEIF6** ((uint32_t)0x00040000)
- #define **DMA_HISR_FEIF6** ((uint32_t)0x00010000)
- #define **DMA_HISR_TCIF5** ((uint32_t)0x00000800)
- #define **DMA_HISR_HTIF5** ((uint32_t)0x00000400)
- #define **DMA_HISR_TEIF5** ((uint32_t)0x00000200)
- #define **DMA_HISR_DMEIF5** ((uint32_t)0x00000100)
- #define **DMA_HISR_FEIF5** ((uint32_t)0x00000040)
- #define **DMA_HISR_TCIF4** ((uint32_t)0x00000020)
- #define **DMA_HISR_HTIF4** ((uint32_t)0x00000010)
- #define **DMA_HISR_TEIF4** ((uint32_t)0x00000008)
- #define **DMA_HISR_DMEIF4** ((uint32_t)0x00000004)
- #define **DMA_HISR_FEIF4** ((uint32_t)0x00000001)
- #define **DMA_LIFCR_CTCIF3** ((uint32_t)0x08000000)
- #define **DMA_LIFCR_CHTIF3** ((uint32_t)0x04000000)
- #define **DMA_LIFCR_CTEIF3** ((uint32_t)0x02000000)
- #define **DMA_LIFCR_CDMEIF3** ((uint32_t)0x01000000)
- #define **DMA_LIFCR_CFEIF3** ((uint32_t)0x00400000)
- #define **DMA_LIFCR_CTCIF2** ((uint32_t)0x00200000)
- #define **DMA_LIFCR_CHTIF2** ((uint32_t)0x00100000)
- #define **DMA_LIFCR_CTEIF2** ((uint32_t)0x00080000)
- #define **DMA_LIFCR_CDMEIF2** ((uint32_t)0x00040000)
- #define **DMA_LIFCR_CFEIF2** ((uint32_t)0x00010000)
- #define **DMA_LIFCR_CTCIF1** ((uint32_t)0x00000800)
- #define **DMA_LIFCR_CHTIF1** ((uint32_t)0x00000400)
- #define **DMA_LIFCR_CTEIF1** ((uint32_t)0x00000200)
- #define **DMA_LIFCR_CDMEIF1** ((uint32_t)0x00000100)
- #define **DMA_LIFCR_CFEIF1** ((uint32_t)0x00000040)
- #define **DMA_LIFCR_CTCIF0** ((uint32_t)0x00000020)
- #define **DMA_LIFCR_CHTIFO** ((uint32_t)0x00000010)
- #define **DMA_LIFCR_CTEIFO** ((uint32_t)0x00000008)
- #define **DMA_LIFCR_CDMEIFO** ((uint32_t)0x00000004)
- #define **DMA_LIFCR_CFEIFO** ((uint32_t)0x00000001)
- #define **DMA_HIFCR_CTCIF7** ((uint32_t)0x08000000)
- #define **DMA_HIFCR_CHTIF7** ((uint32_t)0x04000000)
- #define **DMA_HIFCR_CTEIF7** ((uint32_t)0x02000000)
- #define **DMA_HIFCR_CDMEIF7** ((uint32_t)0x01000000)
- #define **DMA_HIFCR_CFEIF7** ((uint32_t)0x00400000)
- #define **DMA_HIFCR_CTCIF6** ((uint32_t)0x00200000)
- #define **DMA_HIFCR_CHTIF6** ((uint32_t)0x00100000)
- #define **DMA_HIFCR_CTEIF6** ((uint32_t)0x00080000)
- #define **DMA_HIFCR_CDMEIF6** ((uint32_t)0x00040000)
- #define **DMA_HIFCR_CFEIF6** ((uint32_t)0x00010000)
- #define **DMA_HIFCR_CTCIF5** ((uint32_t)0x00000800)
- #define **DMA_HIFCR_CHTIF5** ((uint32_t)0x00000400)
- #define **DMA_HIFCR_CTEIF5** ((uint32_t)0x00000200)
- #define **DMA_HIFCR_CDMEIF5** ((uint32_t)0x00000100)
- #define **DMA_HIFCR_CFEIF5** ((uint32_t)0x00000040)
- #define **DMA_HIFCR_CTCIF4** ((uint32_t)0x00000020)
- #define **DMA_HIFCR_CHTIF4** ((uint32_t)0x00000010)
- #define **DMA_HIFCR_CTEIF4** ((uint32_t)0x00000008)
- #define **DMA_HIFCR_CDMEIF4** ((uint32_t)0x00000004)
- #define **DMA_HIFCR_CFEIF4** ((uint32_t)0x00000001)

- #define DMA2D_CR_START ((uint32_t)0x00000001)
- #define DMA2D_CR_SUSP ((uint32_t)0x00000002)
- #define DMA2D_CR_ABORT ((uint32_t)0x00000004)
- #define DMA2D_CR_TEIE ((uint32_t)0x00000100)
- #define DMA2D_CR_TCIE ((uint32_t)0x00000200)
- #define DMA2D_CR_TWIE ((uint32_t)0x00000400)
- #define DMA2D_CR_CAEIE ((uint32_t)0x00000800)
- #define DMA2D_CR_CTCIE ((uint32_t)0x00001000)
- #define DMA2D_CR_CEIE ((uint32_t)0x00002000)
- #define DMA2D_CR_MODE ((uint32_t)0x00030000)
- #define DMA2D_ISR_TEIF ((uint32_t)0x00000001)
- #define DMA2D_ISR_TCIF ((uint32_t)0x00000002)
- #define DMA2D_ISR_TWIF ((uint32_t)0x00000004)
- #define DMA2D_ISR_CAEIF ((uint32_t)0x00000008)
- #define DMA2D_ISR_CTCIF ((uint32_t)0x00000010)
- #define DMA2D_ISR_CEIF ((uint32_t)0x00000020)
- #define DMA2D_IFCR_CTEIF ((uint32_t)0x00000001)
- #define DMA2D_IFCR_CTCIF ((uint32_t)0x00000002)
- #define DMA2D_IFCR_CTWIF ((uint32_t)0x00000004)
- #define DMA2D_IFCR_CAECIF ((uint32_t)0x00000008)
- #define DMA2D_IFCR_CCTCIF ((uint32_t)0x00000010)
- #define DMA2D_IFCR_CCEIF ((uint32_t)0x00000020)
- #define DMA2D_IFSR_CTEIF DMA2D_IFCR_CTEIF
- #define DMA2D_IFSR_CTCIF DMA2D_IFCR_CTCIF
- #define DMA2D_IFSR_CTWIF DMA2D_IFCR_CTWIF
- #define DMA2D_IFSR_CCAEIF DMA2D_IFCR_CAECIF
- #define DMA2D_IFSR_CCTCIF DMA2D_IFCR_CCTCIF
- #define DMA2D_IFSR_CCEIF DMA2D_IFCR_CCEIF
- #define DMA2D_FGMAR_MA ((uint32_t)0xFFFFFFFF)
- #define DMA2D_FGOR_LO ((uint32_t)0x00003FFF)
- #define DMA2D_BGMAR_MA ((uint32_t)0xFFFFFFFF)
- #define DMA2D_BGOR_LO ((uint32_t)0x00003FFF)
- #define DMA2D_FGPFCCR_CM ((uint32_t)0x0000000F)
- #define DMA2D_FGPFCCR_CM_0 ((uint32_t)0x00000001)
- #define DMA2D_FGPFCCR_CM_1 ((uint32_t)0x00000002)
- #define DMA2D_FGPFCCR_CM_2 ((uint32_t)0x00000004)
- #define DMA2D_FGPFCCR_CM_3 ((uint32_t)0x00000008)
- #define DMA2D_FGPFCCR_CM_3 ((uint32_t)0x00000008)
- #define DMA2D_FGPFCCR_CCM ((uint32_t)0x00000010)
- #define DMA2D_FGPFCCR_START ((uint32_t)0x00000020)
- #define DMA2D_FGPFCCR_CS ((uint32_t)0x0000FF00)
- #define DMA2D_FGPFCCR_AM ((uint32_t)0x00030000)
- #define DMA2D_FGPFCCR_AM_0 ((uint32_t)0x00100000)
- #define DMA2D_FGPFCCR_AM_1 ((uint32_t)0x00200000)
- #define DMA2D_FGPFCCR_ALPHA ((uint32_t)0xFF000000)
- #define DMA2D_FGCOLR_BLUE ((uint32_t)0x000000FF)
- #define DMA2D_FGCOLR_GREEN ((uint32_t)0x0000FF00)
- #define DMA2D_FGCOLR_RED ((uint32_t)0x00FF0000)
- #define DMA2D_BGPFCCR_CM ((uint32_t)0x0000000F)
- #define DMA2D_BGPFCCR_CM_0 ((uint32_t)0x00000001)
- #define DMA2D_BGPFCCR_CM_1 ((uint32_t)0x00000002)
- #define DMA2D_BGPFCCR_CM_2 ((uint32_t)0x00000004)
- #define DMA2D_BGPFCCR_CCM ((uint32_t)0x00000010)
- #define DMA2D_BGPFCCR_START ((uint32_t)0x00000020)
- #define DMA2D_BGPFCCR_CS ((uint32_t)0x0000FF00)

- #define DMA2D_BGPFCCR_AM ((uint32_t)0x00030000)
- #define DMA2D_BGPFCCR_AM_0 ((uint32_t)0x00010000)
- #define DMA2D_BGPFCCR_AM_1 ((uint32_t)0x00020000)
- #define DMA2D_BGPFCCR_ALPHA ((uint32_t)0xFF000000)
- #define DMA2D_BGCOLR_BLUE ((uint32_t)0x000000FF)
- #define DMA2D_BGCOLR_GREEN ((uint32_t)0x0000FF00)
- #define DMA2D_BGCOLR_RED ((uint32_t)0x00FF0000)
- #define DMA2D_FGCMAR_MA ((uint32_t)0xFFFFFFFF)
- #define DMA2D_BGCMAR_MA ((uint32_t)0xFFFFFFFF)
- #define DMA2D_OPFCCR_CM ((uint32_t)0x00000007)
- #define DMA2D_OPFCCR_CM_0 ((uint32_t)0x00000001)
- #define DMA2D_OPFCCR_CM_1 ((uint32_t)0x00000002)
- #define DMA2D_OPFCCR_CM_2 ((uint32_t)0x00000004)
- #define DMA2D_OCOLR_BLUE_1 ((uint32_t)0x000000FF)
- #define DMA2D_OCOLR_GREEN_1 ((uint32_t)0x0000FF00)
- #define DMA2D_OCOLR_RED_1 ((uint32_t)0x00FF0000)
- #define DMA2D_OCOLR_ALPHA_1 ((uint32_t)0xFF000000)
- #define DMA2D_OCOLR_BLUE_2 ((uint32_t)0x0000001F)
- #define DMA2D_OCOLR_GREEN_2 ((uint32_t)0x000007E0)
- #define DMA2D_OCOLR_RED_2 ((uint32_t)0x0000F800)
- #define DMA2D_OCOLR_BLUE_3 ((uint32_t)0x0000001F)
- #define DMA2D_OCOLR_GREEN_3 ((uint32_t)0x000003E0)
- #define DMA2D_OCOLR_RED_3 ((uint32_t)0x00007C00)
- #define DMA2D_OCOLR_ALPHA_3 ((uint32_t)0x00008000)
- #define DMA2D_OCOLR_BLUE_4 ((uint32_t)0x0000000F)
- #define DMA2D_OCOLR_GREEN_4 ((uint32_t)0x000000F0)
- #define DMA2D_OCOLR_RED_4 ((uint32_t)0x00000F00)
- #define DMA2D_OCOLR_ALPHA_4 ((uint32_t)0x0000F000)
- #define DMA2D_OMAR_MA ((uint32_t)0xFFFFFFFF)
- #define DMA2D_OOR_LO ((uint32_t)0x00003FFF)
- #define DMA2D_NLR_NL ((uint32_t)0x0000FFFF)
- #define DMA2D_NLR_PL ((uint32_t)0x3FFF0000)
- #define DMA2D_LWR_LW ((uint32_t)0x0000FFFF)
- #define DMA2D_AMTCR_EN ((uint32_t)0x00000001)
- #define DMA2D_AMTCR_DT ((uint32_t)0x0000FF00)
- #define EXTI_IMR_MR0 ((uint32_t)0x00000001)
- #define EXTI_IMR_MR1 ((uint32_t)0x00000002)
- #define EXTI_IMR_MR2 ((uint32_t)0x00000004)
- #define EXTI_IMR_MR3 ((uint32_t)0x00000008)
- #define EXTI_IMR_MR4 ((uint32_t)0x00000010)
- #define EXTI_IMR_MR5 ((uint32_t)0x00000020)
- #define EXTI_IMR_MR6 ((uint32_t)0x00000040)
- #define EXTI_IMR_MR7 ((uint32_t)0x00000080)
- #define EXTI_IMR_MR8 ((uint32_t)0x00000100)
- #define EXTI_IMR_MR9 ((uint32_t)0x00000200)
- #define EXTI_IMR_MR10 ((uint32_t)0x00000400)
- #define EXTI_IMR_MR11 ((uint32_t)0x00000800)
- #define EXTI_IMR_MR12 ((uint32_t)0x00001000)
- #define EXTI_IMR_MR13 ((uint32_t)0x00002000)
- #define EXTI_IMR_MR14 ((uint32_t)0x00004000)
- #define EXTI_IMR_MR15 ((uint32_t)0x00008000)
- #define EXTI_IMR_MR16 ((uint32_t)0x00010000)
- #define EXTI_IMR_MR17 ((uint32_t)0x00020000)
- #define EXTI_IMR_MR18 ((uint32_t)0x00040000)
- #define EXTI_IMR_MR19 ((uint32_t)0x00080000)

- #define `EXTI_IMR_MR23` ((`uint32_t`)0x00800000)
- #define `EXTI_EMR_MR0` ((`uint32_t`)0x00000001)
- #define `EXTI_EMR_MR1` ((`uint32_t`)0x00000002)
- #define `EXTI_EMR_MR2` ((`uint32_t`)0x00000004)
- #define `EXTI_EMR_MR3` ((`uint32_t`)0x00000008)
- #define `EXTI_EMR_MR4` ((`uint32_t`)0x00000010)
- #define `EXTI_EMR_MR5` ((`uint32_t`)0x00000020)
- #define `EXTI_EMR_MR6` ((`uint32_t`)0x00000040)
- #define `EXTI_EMR_MR7` ((`uint32_t`)0x00000080)
- #define `EXTI_EMR_MR8` ((`uint32_t`)0x00000100)
- #define `EXTI_EMR_MR9` ((`uint32_t`)0x00000200)
- #define `EXTI_EMR_MR10` ((`uint32_t`)0x00000400)
- #define `EXTI_EMR_MR11` ((`uint32_t`)0x00000800)
- #define `EXTI_EMR_MR12` ((`uint32_t`)0x00001000)
- #define `EXTI_EMR_MR13` ((`uint32_t`)0x00002000)
- #define `EXTI_EMR_MR14` ((`uint32_t`)0x00004000)
- #define `EXTI_EMR_MR15` ((`uint32_t`)0x00008000)
- #define `EXTI_EMR_MR16` ((`uint32_t`)0x00010000)
- #define `EXTI_EMR_MR17` ((`uint32_t`)0x00020000)
- #define `EXTI_EMR_MR18` ((`uint32_t`)0x00040000)
- #define `EXTI_EMR_MR19` ((`uint32_t`)0x00080000)
- #define `EXTI_EMR_MR23` ((`uint32_t`)0x00800000)
- #define `EXTI_RTSR_TR0` ((`uint32_t`)0x00000001)
- #define `EXTI_RTSR_TR1` ((`uint32_t`)0x00000002)
- #define `EXTI_RTSR_TR2` ((`uint32_t`)0x00000004)
- #define `EXTI_RTSR_TR3` ((`uint32_t`)0x00000008)
- #define `EXTI_RTSR_TR4` ((`uint32_t`)0x00000010)
- #define `EXTI_RTSR_TR5` ((`uint32_t`)0x00000020)
- #define `EXTI_RTSR_TR6` ((`uint32_t`)0x00000040)
- #define `EXTI_RTSR_TR7` ((`uint32_t`)0x00000080)
- #define `EXTI_RTSR_TR8` ((`uint32_t`)0x00000100)
- #define `EXTI_RTSR_TR9` ((`uint32_t`)0x00000200)
- #define `EXTI_RTSR_TR10` ((`uint32_t`)0x00000400)
- #define `EXTI_RTSR_TR11` ((`uint32_t`)0x00000800)
- #define `EXTI_RTSR_TR12` ((`uint32_t`)0x00001000)
- #define `EXTI_RTSR_TR13` ((`uint32_t`)0x00002000)
- #define `EXTI_RTSR_TR14` ((`uint32_t`)0x00004000)
- #define `EXTI_RTSR_TR15` ((`uint32_t`)0x00008000)
- #define `EXTI_RTSR_TR16` ((`uint32_t`)0x00010000)
- #define `EXTI_RTSR_TR17` ((`uint32_t`)0x00020000)
- #define `EXTI_RTSR_TR18` ((`uint32_t`)0x00040000)
- #define `EXTI_RTSR_TR19` ((`uint32_t`)0x00080000)
- #define `EXTI_RTSR_TR23` ((`uint32_t`)0x00800000)
- #define `EXTI_FTSR_TR0` ((`uint32_t`)0x00000001)
- #define `EXTI_FTSR_TR1` ((`uint32_t`)0x00000002)
- #define `EXTI_FTSR_TR2` ((`uint32_t`)0x00000004)
- #define `EXTI_FTSR_TR3` ((`uint32_t`)0x00000008)
- #define `EXTI_FTSR_TR4` ((`uint32_t`)0x00000010)
- #define `EXTI_FTSR_TR5` ((`uint32_t`)0x00000020)
- #define `EXTI_FTSR_TR6` ((`uint32_t`)0x00000040)
- #define `EXTI_FTSR_TR7` ((`uint32_t`)0x00000080)
- #define `EXTI_FTSR_TR8` ((`uint32_t`)0x00000100)
- #define `EXTI_FTSR_TR9` ((`uint32_t`)0x00000200)
- #define `EXTI_FTSR_TR10` ((`uint32_t`)0x00000400)
- #define `EXTI_FTSR_TR11` ((`uint32_t`)0x00000800)

- #define **EXTI_FTSR_TR12** ((uint32_t)0x00001000)
- #define **EXTI_FTSR_TR13** ((uint32_t)0x00002000)
- #define **EXTI_FTSR_TR14** ((uint32_t)0x00004000)
- #define **EXTI_FTSR_TR15** ((uint32_t)0x00008000)
- #define **EXTI_FTSR_TR16** ((uint32_t)0x00010000)
- #define **EXTI_FTSR_TR17** ((uint32_t)0x00020000)
- #define **EXTI_FTSR_TR18** ((uint32_t)0x00040000)
- #define **EXTI_FTSR_TR19** ((uint32_t)0x00080000)
- #define **EXTI_FTSR_TR23** ((uint32_t)0x00800000)
- #define **EXTI_SWIER_SWIER0** ((uint32_t)0x00000001)
- #define **EXTI_SWIER_SWIER1** ((uint32_t)0x00000002)
- #define **EXTI_SWIER_SWIER2** ((uint32_t)0x00000004)
- #define **EXTI_SWIER_SWIER3** ((uint32_t)0x00000008)
- #define **EXTI_SWIER_SWIER4** ((uint32_t)0x00000010)
- #define **EXTI_SWIER_SWIER5** ((uint32_t)0x00000020)
- #define **EXTI_SWIER_SWIER6** ((uint32_t)0x00000040)
- #define **EXTI_SWIER_SWIER7** ((uint32_t)0x00000080)
- #define **EXTI_SWIER_SWIER8** ((uint32_t)0x00000100)
- #define **EXTI_SWIER_SWIER9** ((uint32_t)0x00000200)
- #define **EXTI_SWIER_SWIER10** ((uint32_t)0x00000400)
- #define **EXTI_SWIER_SWIER11** ((uint32_t)0x00000800)
- #define **EXTI_SWIER_SWIER12** ((uint32_t)0x00001000)
- #define **EXTI_SWIER_SWIER13** ((uint32_t)0x00002000)
- #define **EXTI_SWIER_SWIER14** ((uint32_t)0x00004000)
- #define **EXTI_SWIER_SWIER15** ((uint32_t)0x00008000)
- #define **EXTI_SWIER_SWIER16** ((uint32_t)0x00010000)
- #define **EXTI_SWIER_SWIER17** ((uint32_t)0x00020000)
- #define **EXTI_SWIER_SWIER18** ((uint32_t)0x00040000)
- #define **EXTI_SWIER_SWIER19** ((uint32_t)0x00080000)
- #define **EXTI_SWIER_SWIER23** ((uint32_t)0x00800000)
- #define **EXTI_PR_PR0** ((uint32_t)0x00000001)
- #define **EXTI_PR_PR1** ((uint32_t)0x00000002)
- #define **EXTI_PR_PR2** ((uint32_t)0x00000004)
- #define **EXTI_PR_PR3** ((uint32_t)0x00000008)
- #define **EXTI_PR_PR4** ((uint32_t)0x00000010)
- #define **EXTI_PR_PR5** ((uint32_t)0x00000020)
- #define **EXTI_PR_PR6** ((uint32_t)0x00000040)
- #define **EXTI_PR_PR7** ((uint32_t)0x00000080)
- #define **EXTI_PR_PR8** ((uint32_t)0x00000100)
- #define **EXTI_PR_PR9** ((uint32_t)0x00000200)
- #define **EXTI_PR_PR10** ((uint32_t)0x00000400)
- #define **EXTI_PR_PR11** ((uint32_t)0x00000800)
- #define **EXTI_PR_PR12** ((uint32_t)0x00001000)
- #define **EXTI_PR_PR13** ((uint32_t)0x00002000)
- #define **EXTI_PR_PR14** ((uint32_t)0x00004000)
- #define **EXTI_PR_PR15** ((uint32_t)0x00008000)
- #define **EXTI_PR_PR16** ((uint32_t)0x00010000)
- #define **EXTI_PR_PR17** ((uint32_t)0x00020000)
- #define **EXTI_PR_PR18** ((uint32_t)0x00040000)
- #define **EXTI_PR_PR19** ((uint32_t)0x00080000)
- #define **EXTI_PR_PR23** ((uint32_t)0x00800000)
- #define **FLASH_ACR_LATENCY** ((uint32_t)0x0000000F)
- #define **FLASH_ACR_LATENCY_0WS** ((uint32_t)0x00000000)
- #define **FLASH_ACR_LATENCY_1WS** ((uint32_t)0x00000001)
- #define **FLASH_ACR_LATENCY_2WS** ((uint32_t)0x00000002)

- #define **FLASH_ACR_LATENCY_3WS** ((uint32_t)0x00000003)
- #define **FLASH_ACR_LATENCY_4WS** ((uint32_t)0x00000004)
- #define **FLASH_ACR_LATENCY_5WS** ((uint32_t)0x00000005)
- #define **FLASH_ACR_LATENCY_6WS** ((uint32_t)0x00000006)
- #define **FLASH_ACR_LATENCY_7WS** ((uint32_t)0x00000007)
- #define **FLASH_ACR_LATENCY_8WS** ((uint32_t)0x00000008)
- #define **FLASH_ACR_LATENCY_9WS** ((uint32_t)0x00000009)
- #define **FLASH_ACR_LATENCY_10WS** ((uint32_t)0x0000000A)
- #define **FLASH_ACR_LATENCY_11WS** ((uint32_t)0x0000000B)
- #define **FLASH_ACR_LATENCY_12WS** ((uint32_t)0x0000000C)
- #define **FLASH_ACR_LATENCY_13WS** ((uint32_t)0x0000000D)
- #define **FLASH_ACR_LATENCY_14WS** ((uint32_t)0x0000000E)
- #define **FLASH_ACR_LATENCY_15WS** ((uint32_t)0x0000000F)
- #define **FLASH_ACR_PRFTEN** ((uint32_t)0x00000100)
- #define **FLASH_ACR_ICEN** ((uint32_t)0x00000200)
- #define **FLASH_ACR_DCEN** ((uint32_t)0x00000400)
- #define **FLASH_ACR_ICRST** ((uint32_t)0x00000800)
- #define **FLASH_ACR_DCRST** ((uint32_t)0x00001000)
- #define **FLASH_ACR_BYTE0_ADDRESS** ((uint32_t)0x40023C00)
- #define **FLASH_ACR_BYTE2_ADDRESS** ((uint32_t)0x40023C03)
- #define **FLASH_SR_EOP** ((uint32_t)0x00000001)
- #define **FLASH_SR_SOP** ((uint32_t)0x00000002)
- #define **FLASH_SR_WRPERR** ((uint32_t)0x00000010)
- #define **FLASH_SR_PGAERR** ((uint32_t)0x00000020)
- #define **FLASH_SR_PGPERR** ((uint32_t)0x00000040)
- #define **FLASH_SR_PGSERR** ((uint32_t)0x00000080)
- #define **FLASH_SR_BSY** ((uint32_t)0x00010000)
- #define **FLASH_CR_PG** ((uint32_t)0x00000001)
- #define **FLASH_CR_SER** ((uint32_t)0x00000002)
- #define **FLASH_CR_MER** ((uint32_t)0x00000004)
- #define **FLASH_CR_MER1** **FLASH_CR_MER**
- #define **FLASH_CR_SNBB** ((uint32_t)0x000000F8)
- #define **FLASH_CR_SNBB_0** ((uint32_t)0x00000008)
- #define **FLASH_CR_SNBB_1** ((uint32_t)0x00000010)
- #define **FLASH_CR_SNBB_2** ((uint32_t)0x00000020)
- #define **FLASH_CR_SNBB_3** ((uint32_t)0x00000040)
- #define **FLASH_CR_SNBB_4** ((uint32_t)0x00000040)
- #define **FLASH_CR_PSIZE** ((uint32_t)0x00000300)
- #define **FLASH_CR_PSIZE_0** ((uint32_t)0x00000100)
- #define **FLASH_CR_PSIZE_1** ((uint32_t)0x00000200)
- #define **FLASH_CR_MER2** ((uint32_t)0x00008000)
- #define **FLASH_CR_STRT** ((uint32_t)0x00010000)
- #define **FLASH_CR_EOPIE** ((uint32_t)0x01000000)
- #define **FLASH_CR_LOCK** ((uint32_t)0x80000000)
- #define **FLASH_OPTCR_OPTLOCK** ((uint32_t)0x00000001)
- #define **FLASH_OPTCR_OPTSTRT** ((uint32_t)0x00000002)
- #define **FLASH_OPTCR_BORLEV_0** ((uint32_t)0x00000004)
- #define **FLASH_OPTCR_BORLEV_1** ((uint32_t)0x00000008)
- #define **FLASH_OPTCR_BORLEV** ((uint32_t)0x0000000C)
- #define **FLASH_OPTCR_BFB2** ((uint32_t)0x00000010)
- #define **FLASH_OPTCR_WDG_SW** ((uint32_t)0x00000020)
- #define **FLASH_OPTCR_nRST_STOP** ((uint32_t)0x00000040)
- #define **FLASH_OPTCR_nRST_STDBY** ((uint32_t)0x00000080)
- #define **FLASH_OPTCR_RDP** ((uint32_t)0x0000FF00)
- #define **FLASH_OPTCR_RDP_0** ((uint32_t)0x00000100)

- #define **FLASH_OPTCR_RDP_1** ((uint32_t)0x00000200)
- #define **FLASH_OPTCR_RDP_2** ((uint32_t)0x00000400)
- #define **FLASH_OPTCR_RDP_3** ((uint32_t)0x00000800)
- #define **FLASH_OPTCR_RDP_4** ((uint32_t)0x00001000)
- #define **FLASH_OPTCR_RDP_5** ((uint32_t)0x00002000)
- #define **FLASH_OPTCR_RDP_6** ((uint32_t)0x00004000)
- #define **FLASH_OPTCR_RDP_7** ((uint32_t)0x00008000)
- #define **FLASH_OPTCR_nWRP** ((uint32_t)0xFFFF0000)
- #define **FLASH_OPTCR_nWRP_0** ((uint32_t)0x00010000)
- #define **FLASH_OPTCR_nWRP_1** ((uint32_t)0x00020000)
- #define **FLASH_OPTCR_nWRP_2** ((uint32_t)0x00040000)
- #define **FLASH_OPTCR_nWRP_3** ((uint32_t)0x00080000)
- #define **FLASH_OPTCR_nWRP_4** ((uint32_t)0x00100000)
- #define **FLASH_OPTCR_nWRP_5** ((uint32_t)0x00200000)
- #define **FLASH_OPTCR_nWRP_6** ((uint32_t)0x00400000)
- #define **FLASH_OPTCR_nWRP_7** ((uint32_t)0x00800000)
- #define **FLASH_OPTCR_nWRP_8** ((uint32_t)0x01000000)
- #define **FLASH_OPTCR_nWRP_9** ((uint32_t)0x02000000)
- #define **FLASH_OPTCR_nWRP_10** ((uint32_t)0x04000000)
- #define **FLASH_OPTCR_nWRP_11** ((uint32_t)0x08000000)
- #define **FLASH_OPTCR_DB1M** ((uint32_t)0x40000000)
- #define **FLASH_OPTCR_SPRMOD** ((uint32_t)0x80000000)
- #define **FLASH_OPTCR1_nWRP** ((uint32_t)0xFFFF0000)
- #define **FLASH_OPTCR1_nWRP_0** ((uint32_t)0x00010000)
- #define **FLASH_OPTCR1_nWRP_1** ((uint32_t)0x00020000)
- #define **FLASH_OPTCR1_nWRP_2** ((uint32_t)0x00040000)
- #define **FLASH_OPTCR1_nWRP_3** ((uint32_t)0x00080000)
- #define **FLASH_OPTCR1_nWRP_4** ((uint32_t)0x00100000)
- #define **FLASH_OPTCR1_nWRP_5** ((uint32_t)0x00200000)
- #define **FLASH_OPTCR1_nWRP_6** ((uint32_t)0x00400000)
- #define **FLASH_OPTCR1_nWRP_7** ((uint32_t)0x00800000)
- #define **FLASH_OPTCR1_nWRP_8** ((uint32_t)0x01000000)
- #define **FLASH_OPTCR1_nWRP_9** ((uint32_t)0x02000000)
- #define **FLASH_OPTCR1_nWRP_10** ((uint32_t)0x04000000)
- #define **FLASH_OPTCR1_nWRP_11** ((uint32_t)0x08000000)
- #define **GPIO_MODER_MODERO** ((uint32_t)0x00000003)
- #define **GPIO_MODER_MODERO_0** ((uint32_t)0x00000001)
- #define **GPIO_MODER_MODERO_1** ((uint32_t)0x00000002)
- #define **GPIO_MODER_MODERO1** ((uint32_t)0x0000000C)
- #define **GPIO_MODER_MODER1_0** ((uint32_t)0x00000004)
- #define **GPIO_MODER_MODER1_1** ((uint32_t)0x00000008)
- #define **GPIO_MODER_MODER2** ((uint32_t)0x00000030)
- #define **GPIO_MODER_MODER2_0** ((uint32_t)0x00000010)
- #define **GPIO_MODER_MODER2_1** ((uint32_t)0x00000020)
- #define **GPIO_MODER_MODER3** ((uint32_t)0x000000C0)
- #define **GPIO_MODER_MODER3_0** ((uint32_t)0x00000040)
- #define **GPIO_MODER_MODER3_1** ((uint32_t)0x00000080)
- #define **GPIO_MODER_MODER4** ((uint32_t)0x00000300)
- #define **GPIO_MODER_MODER4_0** ((uint32_t)0x00000100)
- #define **GPIO_MODER_MODER4_1** ((uint32_t)0x00000200)
- #define **GPIO_MODER_MODER5** ((uint32_t)0x00000C00)
- #define **GPIO_MODER_MODER5_0** ((uint32_t)0x00000400)
- #define **GPIO_MODER_MODER5_1** ((uint32_t)0x00000800)
- #define **GPIO_MODER_MODER6** ((uint32_t)0x00003000)
- #define **GPIO_MODER_MODER6_0** ((uint32_t)0x00001000)

- #define **GPIO_MODER_MODER6_1** ((uint32_t)0x00002000)
- #define **GPIO_MODER_MODER7** ((uint32_t)0x0000C000)
- #define **GPIO_MODER_MODER7_0** ((uint32_t)0x00004000)
- #define **GPIO_MODER_MODER7_1** ((uint32_t)0x00008000)
- #define **GPIO_MODER_MODER8** ((uint32_t)0x00030000)
- #define **GPIO_MODER_MODER8_0** ((uint32_t)0x00010000)
- #define **GPIO_MODER_MODER8_1** ((uint32_t)0x00020000)
- #define **GPIO_MODER_MODER9** ((uint32_t)0x000C0000)
- #define **GPIO_MODER_MODER9_0** ((uint32_t)0x00040000)
- #define **GPIO_MODER_MODER9_1** ((uint32_t)0x00080000)
- #define **GPIO_MODER_MODER10** ((uint32_t)0x00300000)
- #define **GPIO_MODER_MODER10_0** ((uint32_t)0x00100000)
- #define **GPIO_MODER_MODER10_1** ((uint32_t)0x00200000)
- #define **GPIO_MODER_MODER11** ((uint32_t)0x00C00000)
- #define **GPIO_MODER_MODER11_0** ((uint32_t)0x00400000)
- #define **GPIO_MODER_MODER11_1** ((uint32_t)0x00800000)
- #define **GPIO_MODER_MODER12** ((uint32_t)0x03000000)
- #define **GPIO_MODER_MODER12_0** ((uint32_t)0x01000000)
- #define **GPIO_MODER_MODER12_1** ((uint32_t)0x02000000)
- #define **GPIO_MODER_MODER13** ((uint32_t)0x0C000000)
- #define **GPIO_MODER_MODER13_0** ((uint32_t)0x04000000)
- #define **GPIO_MODER_MODER13_1** ((uint32_t)0x08000000)
- #define **GPIO_MODER_MODER14** ((uint32_t)0x30000000)
- #define **GPIO_MODER_MODER14_0** ((uint32_t)0x10000000)
- #define **GPIO_MODER_MODER14_1** ((uint32_t)0x20000000)
- #define **GPIO_MODER_MODER15** ((uint32_t)0xC0000000)
- #define **GPIO_MODER_MODER15_0** ((uint32_t)0x40000000)
- #define **GPIO_MODER_MODER15_1** ((uint32_t)0x80000000)
- #define **GPIO_OTYPER_OT_0** ((uint32_t)0x00000001)
- #define **GPIO_OTYPER_OT_1** ((uint32_t)0x00000002)
- #define **GPIO_OTYPER_OT_2** ((uint32_t)0x00000004)
- #define **GPIO_OTYPER_OT_3** ((uint32_t)0x00000008)
- #define **GPIO_OTYPER_OT_4** ((uint32_t)0x00000010)
- #define **GPIO_OTYPER_OT_5** ((uint32_t)0x00000020)
- #define **GPIO_OTYPER_OT_6** ((uint32_t)0x00000040)
- #define **GPIO_OTYPER_OT_7** ((uint32_t)0x00000080)
- #define **GPIO_OTYPER_OT_8** ((uint32_t)0x00000100)
- #define **GPIO_OTYPER_OT_9** ((uint32_t)0x00000200)
- #define **GPIO_OTYPER_OT_10** ((uint32_t)0x00000400)
- #define **GPIO_OTYPER_OT_11** ((uint32_t)0x00000800)
- #define **GPIO_OTYPER_OT_12** ((uint32_t)0x00001000)
- #define **GPIO_OTYPER_OT_13** ((uint32_t)0x00002000)
- #define **GPIO_OTYPER_OT_14** ((uint32_t)0x00004000)
- #define **GPIO_OTYPER_OT_15** ((uint32_t)0x00008000)
- #define **GPIO_OSPEEDER_OSPEEDR0** ((uint32_t)0x00000003)
- #define **GPIO_OSPEEDER_OSPEEDR0_0** ((uint32_t)0x00000001)
- #define **GPIO_OSPEEDER_OSPEEDR0_1** ((uint32_t)0x00000002)
- #define **GPIO_OSPEEDER_OSPEEDR1** ((uint32_t)0x0000000C)
- #define **GPIO_OSPEEDER_OSPEEDR1_0** ((uint32_t)0x00000004)
- #define **GPIO_OSPEEDER_OSPEEDR1_1** ((uint32_t)0x00000008)
- #define **GPIO_OSPEEDER_OSPEEDR2** ((uint32_t)0x00000030)
- #define **GPIO_OSPEEDER_OSPEEDR2_0** ((uint32_t)0x00000010)
- #define **GPIO_OSPEEDER_OSPEEDR2_1** ((uint32_t)0x00000020)
- #define **GPIO_OSPEEDER_OSPEEDR3** ((uint32_t)0x000000C0)
- #define **GPIO_OSPEEDER_OSPEEDR3_0** ((uint32_t)0x00000040)

- #define **GPIO_OSPEEDER_OSPEEDR3_1** ((uint32_t)0x00000080)
- #define **GPIO_OSPEEDER_OSPEEDR4** ((uint32_t)0x000000300)
- #define **GPIO_OSPEEDER_OSPEEDR4_0** ((uint32_t)0x000000100)
- #define **GPIO_OSPEEDER_OSPEEDR4_1** ((uint32_t)0x000000200)
- #define **GPIO_OSPEEDER_OSPEEDR5** ((uint32_t)0x00000C00)
- #define **GPIO_OSPEEDER_OSPEEDR5_0** ((uint32_t)0x000000400)
- #define **GPIO_OSPEEDER_OSPEEDR5_1** ((uint32_t)0x000000800)
- #define **GPIO_OSPEEDER_OSPEEDR6** ((uint32_t)0x000003000)
- #define **GPIO_OSPEEDER_OSPEEDR6_0** ((uint32_t)0x000001000)
- #define **GPIO_OSPEEDER_OSPEEDR6_1** ((uint32_t)0x000002000)
- #define **GPIO_OSPEEDER_OSPEEDR7** ((uint32_t)0x00000C000)
- #define **GPIO_OSPEEDER_OSPEEDR7_0** ((uint32_t)0x000004000)
- #define **GPIO_OSPEEDER_OSPEEDR7_1** ((uint32_t)0x000008000)
- #define **GPIO_OSPEEDER_OSPEEDR8** ((uint32_t)0x000300000)
- #define **GPIO_OSPEEDER_OSPEEDR8_0** ((uint32_t)0x000100000)
- #define **GPIO_OSPEEDER_OSPEEDR8_1** ((uint32_t)0x000200000)
- #define **GPIO_OSPEEDER_OSPEEDR9** ((uint32_t)0x000C00000)
- #define **GPIO_OSPEEDER_OSPEEDR9_0** ((uint32_t)0x000400000)
- #define **GPIO_OSPEEDER_OSPEEDR9_1** ((uint32_t)0x000800000)
- #define **GPIO_OSPEEDER_OSPEEDR10** ((uint32_t)0x003000000)
- #define **GPIO_OSPEEDER_OSPEEDR10_0** ((uint32_t)0x001000000)
- #define **GPIO_OSPEEDER_OSPEEDR10_1** ((uint32_t)0x002000000)
- #define **GPIO_OSPEEDER_OSPEEDR11** ((uint32_t)0x00C000000)
- #define **GPIO_OSPEEDER_OSPEEDR11_0** ((uint32_t)0x004000000)
- #define **GPIO_OSPEEDER_OSPEEDR11_1** ((uint32_t)0x008000000)
- #define **GPIO_OSPEEDER_OSPEEDR12** ((uint32_t)0x030000000)
- #define **GPIO_OSPEEDER_OSPEEDR12_0** ((uint32_t)0x010000000)
- #define **GPIO_OSPEEDER_OSPEEDR12_1** ((uint32_t)0x020000000)
- #define **GPIO_OSPEEDER_OSPEEDR13** ((uint32_t)0x0C0000000)
- #define **GPIO_OSPEEDER_OSPEEDR13_0** ((uint32_t)0x040000000)
- #define **GPIO_OSPEEDER_OSPEEDR13_1** ((uint32_t)0x080000000)
- #define **GPIO_OSPEEDER_OSPEEDR14** ((uint32_t)0x300000000)
- #define **GPIO_OSPEEDER_OSPEEDR14_0** ((uint32_t)0x100000000)
- #define **GPIO_OSPEEDER_OSPEEDR14_1** ((uint32_t)0x200000000)
- #define **GPIO_OSPEEDER_OSPEEDR15** ((uint32_t)0xC00000000)
- #define **GPIO_OSPEEDER_OSPEEDR15_0** ((uint32_t)0x400000000)
- #define **GPIO_OSPEEDER_OSPEEDR15_1** ((uint32_t)0x800000000)
- #define **GPIO_PUPDR_PUPDR0** ((uint32_t)0x00000003)
- #define **GPIO_PUPDR_PUPDR0_0** ((uint32_t)0x000000001)
- #define **GPIO_PUPDR_PUPDR0_1** ((uint32_t)0x000000002)
- #define **GPIO_PUPDR_PUPDR1** ((uint32_t)0x00000000C)
- #define **GPIO_PUPDR_PUPDR1_0** ((uint32_t)0x000000004)
- #define **GPIO_PUPDR_PUPDR1_1** ((uint32_t)0x000000008)
- #define **GPIO_PUPDR_PUPDR2** ((uint32_t)0x000000030)
- #define **GPIO_PUPDR_PUPDR2_0** ((uint32_t)0x000000010)
- #define **GPIO_PUPDR_PUPDR2_1** ((uint32_t)0x000000020)
- #define **GPIO_PUPDR_PUPDR3** ((uint32_t)0x0000000C0)
- #define **GPIO_PUPDR_PUPDR3_0** ((uint32_t)0x000000040)
- #define **GPIO_PUPDR_PUPDR3_1** ((uint32_t)0x000000080)
- #define **GPIO_PUPDR_PUPDR4** ((uint32_t)0x000000300)
- #define **GPIO_PUPDR_PUPDR4_0** ((uint32_t)0x000000100)
- #define **GPIO_PUPDR_PUPDR4_1** ((uint32_t)0x000000200)
- #define **GPIO_PUPDR_PUPDR5** ((uint32_t)0x00000C00)
- #define **GPIO_PUPDR_PUPDR5_0** ((uint32_t)0x00000400)
- #define **GPIO_PUPDR_PUPDR5_1** ((uint32_t)0x00000800)

- #define **GPIO_PUPDR_PUPDR6** ((uint32_t)0x00003000)
- #define **GPIO_PUPDR_PUPDR6_0** ((uint32_t)0x00001000)
- #define **GPIO_PUPDR_PUPDR6_1** ((uint32_t)0x00002000)
- #define **GPIO_PUPDR_PUPDR7** ((uint32_t)0x0000C000)
- #define **GPIO_PUPDR_PUPDR7_0** ((uint32_t)0x00004000)
- #define **GPIO_PUPDR_PUPDR7_1** ((uint32_t)0x00008000)
- #define **GPIO_PUPDR_PUPDR8** ((uint32_t)0x00030000)
- #define **GPIO_PUPDR_PUPDR8_0** ((uint32_t)0x00010000)
- #define **GPIO_PUPDR_PUPDR8_1** ((uint32_t)0x00020000)
- #define **GPIO_PUPDR_PUPDR9** ((uint32_t)0x000C0000)
- #define **GPIO_PUPDR_PUPDR9_0** ((uint32_t)0x00040000)
- #define **GPIO_PUPDR_PUPDR9_1** ((uint32_t)0x00080000)
- #define **GPIO_PUPDR_PUPDR10** ((uint32_t)0x00300000)
- #define **GPIO_PUPDR_PUPDR10_0** ((uint32_t)0x00100000)
- #define **GPIO_PUPDR_PUPDR10_1** ((uint32_t)0x00200000)
- #define **GPIO_PUPDR_PUPDR11** ((uint32_t)0x00C00000)
- #define **GPIO_PUPDR_PUPDR11_0** ((uint32_t)0x00400000)
- #define **GPIO_PUPDR_PUPDR11_1** ((uint32_t)0x00800000)
- #define **GPIO_PUPDR_PUPDR12** ((uint32_t)0x03000000)
- #define **GPIO_PUPDR_PUPDR12_0** ((uint32_t)0x01000000)
- #define **GPIO_PUPDR_PUPDR12_1** ((uint32_t)0x02000000)
- #define **GPIO_PUPDR_PUPDR13** ((uint32_t)0x0C000000)
- #define **GPIO_PUPDR_PUPDR13_0** ((uint32_t)0x04000000)
- #define **GPIO_PUPDR_PUPDR13_1** ((uint32_t)0x08000000)
- #define **GPIO_PUPDR_PUPDR14** ((uint32_t)0x30000000)
- #define **GPIO_PUPDR_PUPDR14_0** ((uint32_t)0x10000000)
- #define **GPIO_PUPDR_PUPDR14_1** ((uint32_t)0x20000000)
- #define **GPIO_PUPDR_PUPDR15** ((uint32_t)0xC0000000)
- #define **GPIO_PUPDR_PUPDR15_0** ((uint32_t)0x40000000)
- #define **GPIO_PUPDR_PUPDR15_1** ((uint32_t)0x80000000)
- #define **GPIO_IDR_IDR_0** ((uint32_t)0x00000001)
- #define **GPIO_IDR_IDR_1** ((uint32_t)0x00000002)
- #define **GPIO_IDR_IDR_2** ((uint32_t)0x00000004)
- #define **GPIO_IDR_IDR_3** ((uint32_t)0x00000008)
- #define **GPIO_IDR_IDR_4** ((uint32_t)0x00000010)
- #define **GPIO_IDR_IDR_5** ((uint32_t)0x00000020)
- #define **GPIO_IDR_IDR_6** ((uint32_t)0x00000040)
- #define **GPIO_IDR_IDR_7** ((uint32_t)0x00000080)
- #define **GPIO_IDR_IDR_8** ((uint32_t)0x00000100)
- #define **GPIO_IDR_IDR_9** ((uint32_t)0x00000200)
- #define **GPIO_IDR_IDR_10** ((uint32_t)0x00000400)
- #define **GPIO_IDR_IDR_11** ((uint32_t)0x00000800)
- #define **GPIO_IDR_IDR_12** ((uint32_t)0x00001000)
- #define **GPIO_IDR_IDR_13** ((uint32_t)0x00002000)
- #define **GPIO_IDR_IDR_14** ((uint32_t)0x00004000)
- #define **GPIO_IDR_IDR_15** ((uint32_t)0x00008000)
- #define **GPIO_OTYPER_IDR_0** GPIO_IDR_IDR_0
- #define **GPIO_OTYPER_IDR_1** GPIO_IDR_IDR_1
- #define **GPIO_OTYPER_IDR_2** GPIO_IDR_IDR_2
- #define **GPIO_OTYPER_IDR_3** GPIO_IDR_IDR_3
- #define **GPIO_OTYPER_IDR_4** GPIO_IDR_IDR_4
- #define **GPIO_OTYPER_IDR_5** GPIO_IDR_IDR_5
- #define **GPIO_OTYPER_IDR_6** GPIO_IDR_IDR_6
- #define **GPIO_OTYPER_IDR_7** GPIO_IDR_IDR_7
- #define **GPIO_OTYPER_IDR_8** GPIO_IDR_IDR_8

- #define **GPIO_OTYPER_IDR_9** GPIO_IDR_IDR_9
- #define **GPIO_OTYPER_IDR_10** GPIO_IDR_IDR_10
- #define **GPIO_OTYPER_IDR_11** GPIO_IDR_IDR_11
- #define **GPIO_OTYPER_IDR_12** GPIO_IDR_IDR_12
- #define **GPIO_OTYPER_IDR_13** GPIO_IDR_IDR_13
- #define **GPIO_OTYPER_IDR_14** GPIO_IDR_IDR_14
- #define **GPIO_OTYPER_IDR_15** GPIO_IDR_IDR_15
- #define **GPIO_ODR_ODR_0** ((uint32_t)0x00000001)
- #define **GPIO_ODR_ODR_1** ((uint32_t)0x00000002)
- #define **GPIO_ODR_ODR_2** ((uint32_t)0x00000004)
- #define **GPIO_ODR_ODR_3** ((uint32_t)0x00000008)
- #define **GPIO_ODR_ODR_4** ((uint32_t)0x00000010)
- #define **GPIO_ODR_ODR_5** ((uint32_t)0x00000020)
- #define **GPIO_ODR_ODR_6** ((uint32_t)0x00000040)
- #define **GPIO_ODR_ODR_7** ((uint32_t)0x00000080)
- #define **GPIO_ODR_ODR_8** ((uint32_t)0x00000100)
- #define **GPIO_ODR_ODR_9** ((uint32_t)0x00000200)
- #define **GPIO_ODR_ODR_10** ((uint32_t)0x00000400)
- #define **GPIO_ODR_ODR_11** ((uint32_t)0x00000800)
- #define **GPIO_ODR_ODR_12** ((uint32_t)0x00001000)
- #define **GPIO_ODR_ODR_13** ((uint32_t)0x00002000)
- #define **GPIO_ODR_ODR_14** ((uint32_t)0x00004000)
- #define **GPIO_ODR_ODR_15** ((uint32_t)0x00008000)
- #define **GPIO_OTYPER_ODR_0** GPIO_ODR_ODR_0
- #define **GPIO_OTYPER_ODR_1** GPIO_ODR_ODR_1
- #define **GPIO_OTYPER_ODR_2** GPIO_ODR_ODR_2
- #define **GPIO_OTYPER_ODR_3** GPIO_ODR_ODR_3
- #define **GPIO_OTYPER_ODR_4** GPIO_ODR_ODR_4
- #define **GPIO_OTYPER_ODR_5** GPIO_ODR_ODR_5
- #define **GPIO_OTYPER_ODR_6** GPIO_ODR_ODR_6
- #define **GPIO_OTYPER_ODR_7** GPIO_ODR_ODR_7
- #define **GPIO_OTYPER_ODR_8** GPIO_ODR_ODR_8
- #define **GPIO_OTYPER_ODR_9** GPIO_ODR_ODR_9
- #define **GPIO_OTYPER_ODR_10** GPIO_ODR_ODR_10
- #define **GPIO_OTYPER_ODR_11** GPIO_ODR_ODR_11
- #define **GPIO_OTYPER_ODR_12** GPIO_ODR_ODR_12
- #define **GPIO_OTYPER_ODR_13** GPIO_ODR_ODR_13
- #define **GPIO_OTYPER_ODR_14** GPIO_ODR_ODR_14
- #define **GPIO_OTYPER_ODR_15** GPIO_ODR_ODR_15
- #define **GPIO_BSRR_BS_0** ((uint32_t)0x00000001)
- #define **GPIO_BSRR_BS_1** ((uint32_t)0x00000002)
- #define **GPIO_BSRR_BS_2** ((uint32_t)0x00000004)
- #define **GPIO_BSRR_BS_3** ((uint32_t)0x00000008)
- #define **GPIO_BSRR_BS_4** ((uint32_t)0x00000010)
- #define **GPIO_BSRR_BS_5** ((uint32_t)0x00000020)
- #define **GPIO_BSRR_BS_6** ((uint32_t)0x00000040)
- #define **GPIO_BSRR_BS_7** ((uint32_t)0x00000080)
- #define **GPIO_BSRR_BS_8** ((uint32_t)0x00000100)
- #define **GPIO_BSRR_BS_9** ((uint32_t)0x00000200)
- #define **GPIO_BSRR_BS_10** ((uint32_t)0x00000400)
- #define **GPIO_BSRR_BS_11** ((uint32_t)0x00000800)
- #define **GPIO_BSRR_BS_12** ((uint32_t)0x00001000)
- #define **GPIO_BSRR_BS_13** ((uint32_t)0x00002000)
- #define **GPIO_BSRR_BS_14** ((uint32_t)0x00004000)
- #define **GPIO_BSRR_BS_15** ((uint32_t)0x00008000)

- #define **GPIO_BSRR_BR_0** ((uint32_t)0x00010000)
- #define **GPIO_BSRR_BR_1** ((uint32_t)0x00020000)
- #define **GPIO_BSRR_BR_2** ((uint32_t)0x00040000)
- #define **GPIO_BSRR_BR_3** ((uint32_t)0x00080000)
- #define **GPIO_BSRR_BR_4** ((uint32_t)0x00100000)
- #define **GPIO_BSRR_BR_5** ((uint32_t)0x00200000)
- #define **GPIO_BSRR_BR_6** ((uint32_t)0x00400000)
- #define **GPIO_BSRR_BR_7** ((uint32_t)0x00800000)
- #define **GPIO_BSRR_BR_8** ((uint32_t)0x01000000)
- #define **GPIO_BSRR_BR_9** ((uint32_t)0x02000000)
- #define **GPIO_BSRR_BR_10** ((uint32_t)0x04000000)
- #define **GPIO_BSRR_BR_11** ((uint32_t)0x08000000)
- #define **GPIO_BSRR_BR_12** ((uint32_t)0x10000000)
- #define **GPIO_BSRR_BR_13** ((uint32_t)0x20000000)
- #define **GPIO_BSRR_BR_14** ((uint32_t)0x40000000)
- #define **GPIO_BSRR_BR_15** ((uint32_t)0x80000000)
- #define **HASH_CR_INIT** ((uint32_t)0x00000004)
- #define **HASH_CR_DMAE** ((uint32_t)0x00000008)
- #define **HASH_CR_DATATYPE** ((uint32_t)0x00000030)
- #define **HASH_CR_DATATYPE_0** ((uint32_t)0x00000010)
- #define **HASH_CR_DATATYPE_1** ((uint32_t)0x00000020)
- #define **HASH_CR_MODE** ((uint32_t)0x00000040)
- #define **HASH_CR_ALGO** ((uint32_t)0x00040080)
- #define **HASH_CR_ALGO_0** ((uint32_t)0x00000080)
- #define **HASH_CR_ALGO_1** ((uint32_t)0x00040000)
- #define **HASH_CR_NBW** ((uint32_t)0x00000F00)
- #define **HASH_CR_NBW_0** ((uint32_t)0x00000100)
- #define **HASH_CR_NBW_1** ((uint32_t)0x00000200)
- #define **HASH_CR_NBW_2** ((uint32_t)0x00000400)
- #define **HASH_CR_NBW_3** ((uint32_t)0x00000800)
- #define **HASH_CR_DINNE** ((uint32_t)0x00001000)
- #define **HASH_CR_MDMAT** ((uint32_t)0x00002000)
- #define **HASH_CR_LKEY** ((uint32_t)0x00010000)
- #define **HASH_STR_NBW** ((uint32_t)0x0000001F)
- #define **HASH_STR_NBW_0** ((uint32_t)0x00000001)
- #define **HASH_STR_NBW_1** ((uint32_t)0x00000002)
- #define **HASH_STR_NBW_2** ((uint32_t)0x00000004)
- #define **HASH_STR_NBW_3** ((uint32_t)0x00000008)
- #define **HASH_STR_NBW_4** ((uint32_t)0x00000010)
- #define **HASH_STR_DCAL** ((uint32_t)0x00000100)
- #define **HASH_IMR_DINIM** ((uint32_t)0x00000001)
- #define **HASH_IMR_DCIM** ((uint32_t)0x00000002)
- #define **HASH_SR_DINIS** ((uint32_t)0x00000001)
- #define **HASH_SR_DCIS** ((uint32_t)0x00000002)
- #define **HASH_SR_DMAS** ((uint32_t)0x00000004)
- #define **HASH_SR_BUSY** ((uint32_t)0x00000008)
- #define **I2C_CR1_PE** ((uint16_t)0x0001)
- #define **I2C_CR1_SMBUS** ((uint16_t)0x0002)
- #define **I2C_CR1_SMBTYPE** ((uint16_t)0x0008)
- #define **I2C_CR1_ENARP** ((uint16_t)0x0010)
- #define **I2C_CR1_ENPEC** ((uint16_t)0x0020)
- #define **I2C_CR1_ENGC** ((uint16_t)0x0040)
- #define **I2C_CR1_NOSTRETCH** ((uint16_t)0x0080)
- #define **I2C_CR1_START** ((uint16_t)0x0100)
- #define **I2C_CR1_STOP** ((uint16_t)0x0200)

- #define I2C_CR1_ACK ((uint16_t)0x0400)
- #define I2C_CR1_POS ((uint16_t)0x0800)
- #define I2C_CR1_PEC ((uint16_t)0x1000)
- #define I2C_CR1_ALERT ((uint16_t)0x2000)
- #define I2C_CR1_SWRST ((uint16_t)0x8000)
- #define I2C_CR2_FREQ ((uint16_t)0x003F)
- #define I2C_CR2_FREQ_0 ((uint16_t)0x0001)
- #define I2C_CR2_FREQ_1 ((uint16_t)0x0002)
- #define I2C_CR2_FREQ_2 ((uint16_t)0x0004)
- #define I2C_CR2_FREQ_3 ((uint16_t)0x0008)
- #define I2C_CR2_FREQ_4 ((uint16_t)0x0010)
- #define I2C_CR2_FREQ_5 ((uint16_t)0x0020)
- #define I2C_CR2_ITERREN ((uint16_t)0x0100)
- #define I2C_CR2_IYTEVTEN ((uint16_t)0x0200)
- #define I2C_CR2_ITBUFEN ((uint16_t)0x0400)
- #define I2C_CR2_DMAEN ((uint16_t)0x0800)
- #define I2C_CR2_LAST ((uint16_t)0x1000)
- #define I2C_OAR1_ADD1_7 ((uint16_t)0x00FE)
- #define I2C_OAR1_ADD8_9 ((uint16_t)0x0300)
- #define I2C_OAR1_ADD0 ((uint16_t)0x0001)
- #define I2C_OAR1_ADD1 ((uint16_t)0x0002)
- #define I2C_OAR1_ADD2 ((uint16_t)0x0004)
- #define I2C_OAR1_ADD3 ((uint16_t)0x0008)
- #define I2C_OAR1_ADD4 ((uint16_t)0x0010)
- #define I2C_OAR1_ADD5 ((uint16_t)0x0020)
- #define I2C_OAR1_ADD6 ((uint16_t)0x0040)
- #define I2C_OAR1_ADD7 ((uint16_t)0x0080)
- #define I2C_OAR1_ADD8 ((uint16_t)0x0100)
- #define I2C_OAR1_ADD9 ((uint16_t)0x0200)
- #define I2C_OAR1_ADDMODE ((uint16_t)0x8000)
- #define I2C_OAR2_ENDUAL ((uint8_t)0x01)
- #define I2C_OAR2_ADD2 ((uint8_t)0xFE)
- #define I2C_DR_DR ((uint8_t)0xFF)
- #define I2C_SR1_SB ((uint16_t)0x0001)
- #define I2C_SR1_ADDR ((uint16_t)0x0002)
- #define I2C_SR1_BTF ((uint16_t)0x0004)
- #define I2C_SR1_ADD10 ((uint16_t)0x0008)
- #define I2C_SR1_STOPF ((uint16_t)0x0010)
- #define I2C_SR1_RXNE ((uint16_t)0x0040)
- #define I2C_SR1_TXE ((uint16_t)0x0080)
- #define I2C_SR1_BERR ((uint16_t)0x0100)
- #define I2C_SR1_ARLO ((uint16_t)0x0200)
- #define I2C_SR1_AF ((uint16_t)0x0400)
- #define I2C_SR1_OVR ((uint16_t)0x0800)
- #define I2C_SR1_PECERR ((uint16_t)0x1000)
- #define I2C_SR1_TIMEOUT ((uint16_t)0x4000)
- #define I2C_SR1_SMBALERT ((uint16_t)0x8000)
- #define I2C_SR2_MSL ((uint16_t)0x0001)
- #define I2C_SR2_BUSY ((uint16_t)0x0002)
- #define I2C_SR2_TRA ((uint16_t)0x0004)
- #define I2C_SR2_GENCALL ((uint16_t)0x0010)
- #define I2C_SR2_SMBDEFAULT ((uint16_t)0x0020)
- #define I2C_SR2_SMBHOST ((uint16_t)0x0040)
- #define I2C_SR2_DUALF ((uint16_t)0x0080)
- #define I2C_SR2_PEC ((uint16_t)0xFF00)

- #define I2C_CCR_CCR ((uint16_t)0x0FFF)
- #define I2C_CCR_DUTY ((uint16_t)0x4000)
- #define I2C_CCR_FS ((uint16_t)0x8000)
- #define I2C_TRISE_TRISE ((uint8_t)0x3F)
- #define I2C_FLTR_DNF ((uint8_t)0x0F)
- #define I2C_FLTR_ANOFF ((uint8_t)0x10)
- #define IWDG_KR_KEY ((uint16_t)0xFFFF)
- #define IWDG_PR_PR ((uint8_t)0x07)
- #define IWDG_PR_PR_0 ((uint8_t)0x01)
- #define IWDG_PR_PR_1 ((uint8_t)0x02)
- #define IWDG_PR_PR_2 ((uint8_t)0x04)
- #define IWDG_RLR_RL ((uint16_t)0x0FFF)
- #define IWDG_SR_PVU ((uint8_t)0x01)
- #define IWDG_SR_RVU ((uint8_t)0x02)
- #define LTDC_SSCR_VSH ((uint32_t)0x0000007FF)
- #define LTDC_SSCR_HSW ((uint32_t)0x0FFF0000)
- #define LTDC_BPCR_AVBP ((uint32_t)0x0000007FF)
- #define LTDC_BPCR_AHBP ((uint32_t)0x0FFF0000)
- #define LTDC_AWCR_AAH ((uint32_t)0x0000007FF)
- #define LTDC_AWCR_AAW ((uint32_t)0x0FFF0000)
- #define LTDC_TWCR_TOTALH ((uint32_t)0x0000007FF)
- #define LTDC_TWCR_TOTALW ((uint32_t)0x0FFF0000)
- #define LTDC_GCR_LTDCEN ((uint32_t)0x000000001)
- #define LTDC_GCR_DBW ((uint32_t)0x00000070)
- #define LTDC_GCR_DGW ((uint32_t)0x00000700)
- #define LTDC_GCR_DRW ((uint32_t)0x00007000)
- #define LTDC_GCR_DEN ((uint32_t)0x00010000)
- #define LTDC_GCR_PCPOL ((uint32_t)0x10000000)
- #define LTDC_GCR_DEPOL ((uint32_t)0x20000000)
- #define LTDC_GCR_VSPOL ((uint32_t)0x40000000)
- #define LTDC_GCR_HSPOL ((uint32_t)0x80000000)
- #define LTDC_GCR_DTEN LTDC_GCR_DEN
- #define LTDC_SRCR_IMR ((uint32_t)0x00000001)
- #define LTDC_SRCR_VBR ((uint32_t)0x00000002)
- #define LTDC_BCCR_BCBLUE ((uint32_t)0x000000FF)
- #define LTDC_BCCR_BCGREEN ((uint32_t)0x0000FF00)
- #define LTDC_BCCR_BCRED ((uint32_t)0x00FF0000)
- #define LTDC_IER_LIE ((uint32_t)0x00000001)
- #define LTDC_IER_FUIE ((uint32_t)0x00000002)
- #define LTDC_IER_TERRIE ((uint32_t)0x00000004)
- #define LTDC_IER_RRIE ((uint32_t)0x00000008)
- #define LTDC_ISR_LIF ((uint32_t)0x00000001)
- #define LTDC_ISR_FUIF ((uint32_t)0x00000002)
- #define LTDC_ISR_TERRIF ((uint32_t)0x00000004)
- #define LTDC_ISR_RRIF ((uint32_t)0x00000008)
- #define LTDC_ICR_CLIF ((uint32_t)0x00000001)
- #define LTDC_ICR_CFUIF ((uint32_t)0x00000002)
- #define LTDC_ICR_CTERRIF ((uint32_t)0x00000004)
- #define LTDC_ICR_CRRIF ((uint32_t)0x00000008)
- #define LTDC_LIPCR_LIPOS ((uint32_t)0x000007FF)
- #define LTDC_CPSR_CYPOS ((uint32_t)0x0000FFFF)
- #define LTDC_CPSR_CXPOS ((uint32_t)0xFFFF0000)
- #define LTDC_CDSR_VDES ((uint32_t)0x00000001)
- #define LTDC_CDSR_HDES ((uint32_t)0x00000002)
- #define LTDC_CDSR_VSYNCS ((uint32_t)0x00000004)

- #define LTDC_CDSR_HSYNCS ((uint32_t)0x00000008)
- #define LTDC_LxCR_LEN ((uint32_t)0x00000001)
- #define LTDC_LxCR_COLKEN ((uint32_t)0x00000002)
- #define LTDC_LxCR_CLUTEN ((uint32_t)0x00000010)
- #define LTDC_LxWHPCR_WHSTPOS ((uint32_t)0x00000FFF)
- #define LTDC_LxWHPCR_WHSPPOS ((uint32_t)0xFFFF0000)
- #define LTDC_LxWVPCR_WVSTPOS ((uint32_t)0x00000FFF)
- #define LTDC_LxWVPCR_WVSPPOS ((uint32_t)0xFFFF0000)
- #define LTDC_LxCKCR_CKBLUE ((uint32_t)0x000000FF)
- #define LTDC_LxCKCR_CKGREEN ((uint32_t)0x0000FF00)
- #define LTDC_LxCKCR_CKRED ((uint32_t)0x00FF0000)
- #define LTDC_LxPFCR_PF ((uint32_t)0x00000007)
- #define LTDC_LxCACR_CONSTA ((uint32_t)0x000000FF)
- #define LTDC_LxDCCR_DCBLUE ((uint32_t)0x000000FF)
- #define LTDC_LxDCCR_DCGREEN ((uint32_t)0x0000FF00)
- #define LTDC_LxDCCR_DCRED ((uint32_t)0x00FF0000)
- #define LTDC_LxDCCR_DCALPHA ((uint32_t)0xFF000000)
- #define LTDC_LxBFCR_BF2 ((uint32_t)0x00000007)
- #define LTDC_LxBFCR_BF1 ((uint32_t)0x00000700)
- #define LTDC_LxCFBAR_CFBADD ((uint32_t)0xFFFFFFFF)
- #define LTDC_LxCFBLR_CFBLL ((uint32_t)0x00001FFF)
- #define LTDC_LxCFBLNR_CFBLNBR ((uint32_t)0x000007FF)
- #define LTDC_LxCLUTWR_BLUE ((uint32_t)0x000000FF)
- #define LTDC_LxCLUTWR_GREEN ((uint32_t)0x0000FF00)
- #define LTDC_LxCLUTWR_RED ((uint32_t)0x00FF0000)
- #define LTDC_LxCLUTWR_CLUTADD ((uint32_t)0xFF000000)
- #define PWR_CR_LPDS ((uint32_t)0x00000001)
- #define PWR_CR_PDDS ((uint32_t)0x00000002)
- #define PWR_CR_CWUF ((uint32_t)0x00000004)
- #define PWR_CR_CSBF ((uint32_t)0x00000008)
- #define PWR_CR_PVDE ((uint32_t)0x00000010)
- #define PWR_CR_PLS ((uint32_t)0x000000E0)
- #define PWR_CR_PLS_0 ((uint32_t)0x00000020)
- #define PWR_CR_PLS_1 ((uint32_t)0x00000040)
- #define PWR_CR_PLS_2 ((uint32_t)0x00000080)
- #define PWR_CR_PLS_LEVO ((uint32_t)0x00000000)
- #define PWR_CR_PLSLEV1 ((uint32_t)0x00000020)
- #define PWR_CR_PLSLEV2 ((uint32_t)0x00000040)
- #define PWR_CR_PLSLEV3 ((uint32_t)0x00000060)
- #define PWR_CR_PLSLEV4 ((uint32_t)0x00000080)
- #define PWR_CR_PLSLEV5 ((uint32_t)0x000000A0)
- #define PWR_CR_PLSLEV6 ((uint32_t)0x000000C0)
- #define PWR_CR_PLSLEV7 ((uint32_t)0x000000E0)
- #define PWR_CR_DBP ((uint32_t)0x00000100)
- #define PWR_CR_FPDS ((uint32_t)0x00000200)
- #define PWR_CR_LPUDS ((uint32_t)0x00000400)
- #define PWR_CR_MRUDS ((uint32_t)0x00000800)
- #define PWR_CR_LPLVDS ((uint32_t)0x00000400)
- #define PWR_CR_MRLVDS ((uint32_t)0x00000800)
- #define PWR_CR_ADCDC1 ((uint32_t)0x00002000)
- #define PWR_CR_VOS ((uint32_t)0x0000C000)
- #define PWR_CR_VOS_0 ((uint32_t)0x00004000)
- #define PWR_CR_VOS_1 ((uint32_t)0x00008000)
- #define PWR_CR_ODEN ((uint32_t)0x00010000)

- #define PWR_CR_ODSWEN ((uint32_t)0x00020000)
- #define PWR_CR_UDEN ((uint32_t)0x000C0000)
- #define PWR_CR_UDEN_0 ((uint32_t)0x00040000)
- #define PWR_CR_UDEN_1 ((uint32_t)0x00080000)
- #define PWR_CR_FMSSR ((uint32_t)0x00100000)
- #define PWR_CR_FISSR ((uint32_t)0x00200000)
- #define PWR_CR_PMODE PWR_CR_VOS
- #define PWR_CSR_WUF ((uint32_t)0x00000001)
- #define PWR_CSR_SBF ((uint32_t)0x00000002)
- #define PWR_CSR_PVDO ((uint32_t)0x00000004)
- #define PWR_CSR_BRR ((uint32_t)0x00000008)
- #define PWR_CSR_WUPP ((uint32_t)0x00000080)
- #define PWR_CSR_EWUP ((uint32_t)0x00000100)
- #define PWR_CSR_BRE ((uint32_t)0x00000200)
- #define PWR_CSR_VOSRDY ((uint32_t)0x00004000)
- #define PWR_CSR_ODRDY ((uint32_t)0x00010000)
- #define PWR_CSR_ODSWRDY ((uint32_t)0x00020000)
- #define PWR_CSR_UDSWRDY ((uint32_t)0x000C0000)
- #define PWR_CSR_REGRDY PWR_CSR_VOSRDY
- #define RCC_CR_HSION ((uint32_t)0x00000001)
- #define RCC_CR_HSIRDY ((uint32_t)0x00000002)
- #define RCC_CR_HSITRIM ((uint32_t)0x000000F8)
- #define RCC_CR_HSITRIM_0 ((uint32_t)0x00000008)
- #define RCC_CR_HSITRIM_1 ((uint32_t)0x00000010)
- #define RCC_CR_HSITRIM_2 ((uint32_t)0x00000020)
- #define RCC_CR_HSITRIM_3 ((uint32_t)0x00000040)
- #define RCC_CR_HSITRIM_4 ((uint32_t)0x00000080)
- #define RCC_CR_HSICAL ((uint32_t)0x0000FF00)
- #define RCC_CR_HSICAL_0 ((uint32_t)0x00000100)
- #define RCC_CR_HSICAL_1 ((uint32_t)0x00000200)
- #define RCC_CR_HSICAL_2 ((uint32_t)0x00000400)
- #define RCC_CR_HSICAL_3 ((uint32_t)0x00000800)
- #define RCC_CR_HSICAL_4 ((uint32_t)0x00001000)
- #define RCC_CR_HSICAL_5 ((uint32_t)0x00002000)
- #define RCC_CR_HSICAL_6 ((uint32_t)0x00004000)
- #define RCC_CR_HSICAL_7 ((uint32_t)0x00008000)
- #define RCC_CR_HSEON ((uint32_t)0x00010000)
- #define RCC_CR_HSERDY ((uint32_t)0x00020000)
- #define RCC_CR_HSEBYP ((uint32_t)0x00040000)
- #define RCC_CR_CSSON ((uint32_t)0x00080000)
- #define RCC_CR_PLLON ((uint32_t)0x01000000)
- #define RCC_CR_PLLRDY ((uint32_t)0x02000000)
- #define RCC_CR_PLLI2SON ((uint32_t)0x04000000)
- #define RCC_CR_PLLI2SRDY ((uint32_t)0x08000000)
- #define RCC_CR_PLLSAION ((uint32_t)0x10000000)
- #define RCC_CR_PLLSAIRDY ((uint32_t)0x20000000)
- #define RCC_PLLCFG_R_PLLM ((uint32_t)0x0000003F)
- #define RCC_PLLCFG_R_PLLM_0 ((uint32_t)0x00000001)
- #define RCC_PLLCFG_R_PLLM_1 ((uint32_t)0x00000002)
- #define RCC_PLLCFG_R_PLLM_2 ((uint32_t)0x00000004)
- #define RCC_PLLCFG_R_PLLM_3 ((uint32_t)0x00000008)
- #define RCC_PLLCFG_R_PLLM_4 ((uint32_t)0x00000010)
- #define RCC_PLLCFG_R_PLLM_5 ((uint32_t)0x00000020)
- #define RCC_PLLCFG_R_PLLN ((uint32_t)0x00007FC0)
- #define RCC_PLLCFG_R_PLLN_0 ((uint32_t)0x00000040)

- #define **RCC_PLLCFGR_PLLN_1** ((uint32_t)0x00000080)
- #define **RCC_PLLCFGR_PLLN_2** ((uint32_t)0x00000100)
- #define **RCC_PLLCFGR_PLLN_3** ((uint32_t)0x00000200)
- #define **RCC_PLLCFGR_PLLN_4** ((uint32_t)0x00000400)
- #define **RCC_PLLCFGR_PLLN_5** ((uint32_t)0x00000800)
- #define **RCC_PLLCFGR_PLLN_6** ((uint32_t)0x00001000)
- #define **RCC_PLLCFGR_PLLN_7** ((uint32_t)0x00002000)
- #define **RCC_PLLCFGR_PLLN_8** ((uint32_t)0x00004000)
- #define **RCC_PLLCFGR_PLLP** ((uint32_t)0x00030000)
- #define **RCC_PLLCFGR_PLLP_0** ((uint32_t)0x00010000)
- #define **RCC_PLLCFGR_PLLP_1** ((uint32_t)0x00020000)
- #define **RCC_PLLCFGR_PLLSRC** ((uint32_t)0x00400000)
- #define **RCC_PLLCFGR_PLLSRC_HSE** ((uint32_t)0x0040000000)
- #define **RCC_PLLCFGR_PLLSRC_HSI** ((uint32_t)0x00000000)
- #define **RCC_PLLCFGR_PLLQ** ((uint32_t)0xF0000000)
- #define **RCC_PLLCFGR_PLLQ_0** ((uint32_t)0x01000000)
- #define **RCC_PLLCFGR_PLLQ_1** ((uint32_t)0x02000000)
- #define **RCC_PLLCFGR_PLLQ_2** ((uint32_t)0x04000000)
- #define **RCC_PLLCFGR_PLLQ_3** ((uint32_t)0x08000000)
- #define **RCC_CFGR_SW** ((uint32_t)0x00000003)
- #define **RCC_CFGR_SW_0** ((uint32_t)0x00000001)
- #define **RCC_CFGR_SW_1** ((uint32_t)0x00000002)
- #define **RCC_CFGR_SW_HSI** ((uint32_t)0x00000000)
- #define **RCC_CFGR_SW_HSE** ((uint32_t)0x00000001)
- #define **RCC_CFGR_SW_PLL** ((uint32_t)0x00000002)
- #define **RCC_CFGR_SWS** ((uint32_t)0x0000000C)
- #define **RCC_CFGR_SWS_0** ((uint32_t)0x00000004)
- #define **RCC_CFGR_SWS_1** ((uint32_t)0x00000008)
- #define **RCC_CFGR_SWS_HSI** ((uint32_t)0x00000000)
- #define **RCC_CFGR_SWS_HSE** ((uint32_t)0x00000004)
- #define **RCC_CFGR_SWS_PLL** ((uint32_t)0x00000008)
- #define **RCC_CFGR_HPRE** ((uint32_t)0x000000F0)
- #define **RCC_CFGR_HPRE_0** ((uint32_t)0x00000010)
- #define **RCC_CFGR_HPRE_1** ((uint32_t)0x00000020)
- #define **RCC_CFGR_HPRE_2** ((uint32_t)0x00000040)
- #define **RCC_CFGR_HPRE_3** ((uint32_t)0x00000080)
- #define **RCC_CFGR_HPRE_DIV1** ((uint32_t)0x00000000)
- #define **RCC_CFGR_HPRE_DIV2** ((uint32_t)0x00000080)
- #define **RCC_CFGR_HPRE_DIV4** ((uint32_t)0x00000090)
- #define **RCC_CFGR_HPRE_DIV8** ((uint32_t)0x000000A0)
- #define **RCC_CFGR_HPRE_DIV16** ((uint32_t)0x000000B0)
- #define **RCC_CFGR_HPRE_DIV64** ((uint32_t)0x000000C0)
- #define **RCC_CFGR_HPRE_DIV128** ((uint32_t)0x000000D0)
- #define **RCC_CFGR_HPRE_DIV256** ((uint32_t)0x000000E0)
- #define **RCC_CFGR_HPRE_DIV512** ((uint32_t)0x000000F0)
- #define **RCC_CFGR_PPREG1** ((uint32_t)0x00001C00)
- #define **RCC_CFGR_PPREG1_0** ((uint32_t)0x00000400)
- #define **RCC_CFGR_PPREG1_1** ((uint32_t)0x00000800)
- #define **RCC_CFGR_PPREG1_2** ((uint32_t)0x00001000)
- #define **RCC_CFGR_PPREG1_DIV1** ((uint32_t)0x00000000)
- #define **RCC_CFGR_PPREG1_DIV2** ((uint32_t)0x00001000)
- #define **RCC_CFGR_PPREG1_DIV4** ((uint32_t)0x00001400)
- #define **RCC_CFGR_PPREG1_DIV8** ((uint32_t)0x00001800)
- #define **RCC_CFGR_PPREG1_DIV16** ((uint32_t)0x00001C00)
- #define **RCC_CFGR_PPREG2** ((uint32_t)0x0000E000)

- #define **RCC_CFGR_PPREG_0** ((uint32_t)0x00002000)
- #define **RCC_CFGR_PPREG_1** ((uint32_t)0x00004000)
- #define **RCC_CFGR_PPREG_2** ((uint32_t)0x00008000)
- #define **RCC_CFGR_PPREG_DIV1** ((uint32_t)0x00000000)
- #define **RCC_CFGR_PPREG_DIV2** ((uint32_t)0x00008000)
- #define **RCC_CFGR_PPREG_DIV4** ((uint32_t)0x0000A000)
- #define **RCC_CFGR_PPREG_DIV8** ((uint32_t)0x0000C000)
- #define **RCC_CFGR_PPREG_DIV16** ((uint32_t)0x0000E000)
- #define **RCC_CFGR_RTCPRE** ((uint32_t)0x001F0000)
- #define **RCC_CFGR_RTCPRE_0** ((uint32_t)0x00010000)
- #define **RCC_CFGR_RTCPRE_1** ((uint32_t)0x00020000)
- #define **RCC_CFGR_RTCPRE_2** ((uint32_t)0x00040000)
- #define **RCC_CFGR_RTCPRE_3** ((uint32_t)0x00080000)
- #define **RCC_CFGR_RTCPRE_4** ((uint32_t)0x00100000)
- #define **RCC_CFGR_MCO1** ((uint32_t)0x00600000)
- #define **RCC_CFGR_MCO1_0** ((uint32_t)0x00200000)
- #define **RCC_CFGR_MCO1_1** ((uint32_t)0x00400000)
- #define **RCC_CFGR_I2SSRC** ((uint32_t)0x00800000)
- #define **RCC_CFGR_MCO1PRE** ((uint32_t)0x07000000)
- #define **RCC_CFGR_MCO1PRE_0** ((uint32_t)0x01000000)
- #define **RCC_CFGR_MCO1PRE_1** ((uint32_t)0x02000000)
- #define **RCC_CFGR_MCO1PRE_2** ((uint32_t)0x04000000)
- #define **RCC_CFGR_MCO2PRE** ((uint32_t)0x38000000)
- #define **RCC_CFGR_MCO2PRE_0** ((uint32_t)0x08000000)
- #define **RCC_CFGR_MCO2PRE_1** ((uint32_t)0x10000000)
- #define **RCC_CFGR_MCO2PRE_2** ((uint32_t)0x20000000)
- #define **RCC_CFGR_MCO2** ((uint32_t)0xC0000000)
- #define **RCC_CFGR_MCO2_0** ((uint32_t)0x40000000)
- #define **RCC_CFGR_MCO2_1** ((uint32_t)0x80000000)
- #define **RCC_CIR_LSIRDYF** ((uint32_t)0x00000001)
- #define **RCC_CIR_LSERDYF** ((uint32_t)0x00000002)
- #define **RCC_CIR_HSIRDYF** ((uint32_t)0x00000004)
- #define **RCC_CIR_HSERDYF** ((uint32_t)0x00000008)
- #define **RCC_CIR_PLLRDYF** ((uint32_t)0x00000010)
- #define **RCC_CIR_PLLI2SRDYF** ((uint32_t)0x00000020)
- #define **RCC_CIR_PLLSAIRDYF** ((uint32_t)0x00000040)
- #define **RCC_CIR_CSSF** ((uint32_t)0x00000080)
- #define **RCC_CIR_LSIRDYIE** ((uint32_t)0x00000100)
- #define **RCC_CIR_LSERDYIE** ((uint32_t)0x00000200)
- #define **RCC_CIR_HSIRDYIE** ((uint32_t)0x00000400)
- #define **RCC_CIR_HSERDYIE** ((uint32_t)0x00000800)
- #define **RCC_CIR_PLLRDYIE** ((uint32_t)0x00001000)
- #define **RCC_CIR_PLLI2SRDYIE** ((uint32_t)0x00002000)
- #define **RCC_CIR_PLLSAIRDYIE** ((uint32_t)0x00004000)
- #define **RCC_CIR_LSIRDYC** ((uint32_t)0x00010000)
- #define **RCC_CIR_LSERDYC** ((uint32_t)0x00020000)
- #define **RCC_CIR_HSIRDYC** ((uint32_t)0x00040000)
- #define **RCC_CIR_HSERDYC** ((uint32_t)0x00080000)
- #define **RCC_CIR_PLLRDYC** ((uint32_t)0x00100000)
- #define **RCC_CIR_PLLI2SRDYC** ((uint32_t)0x00200000)
- #define **RCC_CIR_PLLSAIRDYC** ((uint32_t)0x00400000)
- #define **RCC_CIR_CSSC** ((uint32_t)0x00800000)
- #define **RCC_AHB1RSTR_GPIOARST** ((uint32_t)0x00000001)
- #define **RCC_AHB1RSTR_GPIOBRST** ((uint32_t)0x00000002)
- #define **RCC_AHB1RSTR_GPIOCRST** ((uint32_t)0x00000004)

- #define **RCC_AHB1RSTR_GPIODRST** ((uint32_t)0x00000008)
- #define **RCC_AHB1RSTR_GPIOERST** ((uint32_t)0x00000010)
- #define **RCC_AHB1RSTR_GPIOFRST** ((uint32_t)0x00000020)
- #define **RCC_AHB1RSTR_GPIOGRST** ((uint32_t)0x00000040)
- #define **RCC_AHB1RSTR_GPIOHRST** ((uint32_t)0x00000080)
- #define **RCC_AHB1RSTR_GPIOIRST** ((uint32_t)0x00000100)
- #define **RCC_AHB1RSTR_GPIOJRST** ((uint32_t)0x00000200)
- #define **RCC_AHB1RSTR_GPIOKRST** ((uint32_t)0x00000400)
- #define **RCC_AHB1RSTR_CRCRST** ((uint32_t)0x00001000)
- #define **RCC_AHB1RSTR_DMA1RST** ((uint32_t)0x00200000)
- #define **RCC_AHB1RSTR_DMA2RST** ((uint32_t)0x00400000)
- #define **RCC_AHB1RSTR_DMA2DRST** ((uint32_t)0x00800000)
- #define **RCC_AHB1RSTR_ETHMACRST** ((uint32_t)0x02000000)
- #define **RCC_AHB1RSTR_OTGHRST** ((uint32_t)0x10000000)
- #define **RCC_AHB2RSTR_DCMIRST** ((uint32_t)0x00000001)
- #define **RCC_AHB2RSTR_CRYPRST** ((uint32_t)0x00000010)
- #define **RCC_AHB2RSTR_HASHRST** ((uint32_t)0x00000020)
- #define **RCC_AHB2RSTR_HSAHRST** RCC_AHB2RSTR_HASHRST
- #define **RCC_AHB2RSTR_RNGRST** ((uint32_t)0x00000040)
- #define **RCC_AHB2RSTR_OTGFSRST** ((uint32_t)0x00000080)
- #define **RCC_APB1RSTR_TIM2RST** ((uint32_t)0x00000001)
- #define **RCC_APB1RSTR_TIM3RST** ((uint32_t)0x00000002)
- #define **RCC_APB1RSTR_TIM4RST** ((uint32_t)0x00000004)
- #define **RCC_APB1RSTR_TIM5RST** ((uint32_t)0x00000008)
- #define **RCC_APB1RSTR_TIM6RST** ((uint32_t)0x00000010)
- #define **RCC_APB1RSTR_TIM7RST** ((uint32_t)0x00000020)
- #define **RCC_APB1RSTR_TIM12RST** ((uint32_t)0x00000040)
- #define **RCC_APB1RSTR_TIM13RST** ((uint32_t)0x00000080)
- #define **RCC_APB1RSTR_TIM14RST** ((uint32_t)0x00000100)
- #define **RCC_APB1RSTR_WWDGRST** ((uint32_t)0x00000800)
- #define **RCC_APB1RSTR_SPI2RST** ((uint32_t)0x00004000)
- #define **RCC_APB1RSTR_SPI3RST** ((uint32_t)0x00008000)
- #define **RCC_APB1RSTR_USART2RST** ((uint32_t)0x00020000)
- #define **RCC_APB1RSTR_USART3RST** ((uint32_t)0x00040000)
- #define **RCC_APB1RSTR_UART4RST** ((uint32_t)0x00080000)
- #define **RCC_APB1RSTR_UART5RST** ((uint32_t)0x00100000)
- #define **RCC_APB1RSTR_I2C1RST** ((uint32_t)0x00200000)
- #define **RCC_APB1RSTR_I2C2RST** ((uint32_t)0x00400000)
- #define **RCC_APB1RSTR_I2C3RST** ((uint32_t)0x00800000)
- #define **RCC_APB1RSTR_CAN1RST** ((uint32_t)0x02000000)
- #define **RCC_APB1RSTR_CAN2RST** ((uint32_t)0x04000000)
- #define **RCC_APB1RSTR_PWRRST** ((uint32_t)0x10000000)
- #define **RCC_APB1RSTR_DACRST** ((uint32_t)0x20000000)
- #define **RCC_APB1RSTR_UART7RST** ((uint32_t)0x40000000)
- #define **RCC_APB1RSTR_UART8RST** ((uint32_t)0x80000000)
- #define **RCC_APB2RSTR_TIM1RST** ((uint32_t)0x00000001)
- #define **RCC_APB2RSTR_TIM8RST** ((uint32_t)0x00000002)
- #define **RCC_APB2RSTR_USART1RST** ((uint32_t)0x00000010)
- #define **RCC_APB2RSTR_USART6RST** ((uint32_t)0x00000020)
- #define **RCC_APB2RSTR_USART9RST** ((uint32_t)0x00000040)
- #define **RCC_APB2RSTR_USART10RST** ((uint32_t)0x00000080)
- #define **RCC_APB2RSTR_ADCRST** ((uint32_t)0x00000100)
- #define **RCC_APB2RSTR_SDIORST** ((uint32_t)0x00000800)
- #define **RCC_APB2RSTR_SPI1RST** ((uint32_t)0x00001000)
- #define **RCC_APB2RSTR_SPI4RST** ((uint32_t)0x00002000)

- #define **RCC_APB2RSTR_SYSCFGRST** ((uint32_t)0x00004000)
- #define **RCC_APB2RSTR_TIM9RST** ((uint32_t)0x00010000)
- #define **RCC_APB2RSTR_TIM10RST** ((uint32_t)0x00020000)
- #define **RCC_APB2RSTR_TIM11RST** ((uint32_t)0x00040000)
- #define **RCC_APB2RSTR_SPI5RST** ((uint32_t)0x00100000)
- #define **RCC_APB2RSTR_SPI6RST** ((uint32_t)0x00200000)
- #define **RCC_APB2RSTR_SAI1RST** ((uint32_t)0x00400000)
- #define **RCC_APB2RSTR_LTDCRST** ((uint32_t)0x04000000)
- #define **RCC_APB2RSTR_SPI1** RCC_APB2RSTR_SPI1RST
- #define **RCC_APB2RSTR_DFSDMRST** RCC_APB2RSTR_DFSDM1RST
- #define **RCC_AHB1ENR_GPIOAEN** ((uint32_t)0x00000001)
- #define **RCC_AHB1ENR_GPIOBEN** ((uint32_t)0x00000002)
- #define **RCC_AHB1ENR_GPIOCEN** ((uint32_t)0x00000004)
- #define **RCC_AHB1ENR_GPIODEN** ((uint32_t)0x00000008)
- #define **RCC_AHB1ENR_GPIOEEN** ((uint32_t)0x00000010)
- #define **RCC_AHB1ENR_GPIOFEN** ((uint32_t)0x00000020)
- #define **RCC_AHB1ENR_GPIOGEN** ((uint32_t)0x00000040)
- #define **RCC_AHB1ENR_GPIOHEN** ((uint32_t)0x00000080)
- #define **RCC_AHB1ENR_GPIOIEN** ((uint32_t)0x00000100)
- #define **RCC_AHB1ENR_GPIOJEN** ((uint32_t)0x00000200)
- #define **RCC_AHB1ENR_GPIOKEN** ((uint32_t)0x00000400)
- #define **RCC_AHB1ENR_CRCEN** ((uint32_t)0x00001000)
- #define **RCC_AHB1ENR_BKPSRAMEN** ((uint32_t)0x00040000)
- #define **RCC_AHB1ENR_CCMDATARAMEN** ((uint32_t)0x00100000)
- #define **RCC_AHB1ENR_DMA1EN** ((uint32_t)0x00200000)
- #define **RCC_AHB1ENR_DMA2EN** ((uint32_t)0x00400000)
- #define **RCC_AHB1ENR_DMA2DEN** ((uint32_t)0x00800000)
- #define **RCC_AHB1ENR_ETHMACEN** ((uint32_t)0x02000000)
- #define **RCC_AHB1ENR_ETHMACTXEN** ((uint32_t)0x04000000)
- #define **RCC_AHB1ENR_ETHMACRXEN** ((uint32_t)0x08000000)
- #define **RCC_AHB1ENR_ETHMACPTPEN** ((uint32_t)0x10000000)
- #define **RCC_AHB1ENR_OTGHSEN** ((uint32_t)0x20000000)
- #define **RCC_AHB1ENR_OTGHSULPIEN** ((uint32_t)0x40000000)
- #define **RCC_AHB2ENR_DCMIEN** ((uint32_t)0x00000001)
- #define **RCC_AHB2ENR_CRYPEN** ((uint32_t)0x00000010)
- #define **RCC_AHB2ENR_HASHEN** ((uint32_t)0x00000020)
- #define **RCC_AHB2ENR_RNGEN** ((uint32_t)0x00000040)
- #define **RCC_AHB2ENR_OTGFSEN** ((uint32_t)0x00000080)
- #define **RCC_APB1ENR_TIM2EN** ((uint32_t)0x00000001)
- #define **RCC_APB1ENR_TIM3EN** ((uint32_t)0x00000002)
- #define **RCC_APB1ENR_TIM4EN** ((uint32_t)0x00000004)
- #define **RCC_APB1ENR_TIM5EN** ((uint32_t)0x00000008)
- #define **RCC_APB1ENR_TIM6EN** ((uint32_t)0x00000010)
- #define **RCC_APB1ENR_TIM7EN** ((uint32_t)0x00000020)
- #define **RCC_APB1ENR_TIM12EN** ((uint32_t)0x00000040)
- #define **RCC_APB1ENR_TIM13EN** ((uint32_t)0x00000080)
- #define **RCC_APB1ENR_TIM14EN** ((uint32_t)0x00000100)
- #define **RCC_APB1ENR_WWDGEN** ((uint32_t)0x00000800)
- #define **RCC_APB1ENR_SPI2EN** ((uint32_t)0x00004000)
- #define **RCC_APB1ENR_SPI3EN** ((uint32_t)0x00008000)
- #define **RCC_APB1ENR_USART2EN** ((uint32_t)0x00020000)
- #define **RCC_APB1ENR_USART3EN** ((uint32_t)0x00040000)
- #define **RCC_APB1ENR_UART4EN** ((uint32_t)0x00080000)
- #define **RCC_APB1ENR_UART5EN** ((uint32_t)0x00100000)
- #define **RCC_APB1ENR_I2C1EN** ((uint32_t)0x00200000)

- #define **RCC_APB1ENR_I2C2EN** ((uint32_t)0x00400000)
- #define **RCC_APB1ENR_I2C3EN** ((uint32_t)0x00800000)
- #define **RCC_APB1ENR_CAN1EN** ((uint32_t)0x02000000)
- #define **RCC_APB1ENR_CAN2EN** ((uint32_t)0x04000000)
- #define **RCC_APB1ENR_PWREN** ((uint32_t)0x10000000)
- #define **RCC_APB1ENR_DACEN** ((uint32_t)0x20000000)
- #define **RCC_APB1ENR_UART7EN** ((uint32_t)0x40000000)
- #define **RCC_APB1ENR_UART8EN** ((uint32_t)0x80000000)
- #define **RCC_APB2ENR_TIM1EN** ((uint32_t)0x00000001)
- #define **RCC_APB2ENR_TIM8EN** ((uint32_t)0x00000002)
- #define **RCC_APB2ENR_USART1EN** ((uint32_t)0x00000010)
- #define **RCC_APB2ENR_USART6EN** ((uint32_t)0x00000020)
- #define **RCC_APB2ENR_UART9EN** ((uint32_t)0x00000040)
- #define **RCC_APB2ENR_UART10EN** ((uint32_t)0x00000080)
- #define **RCC_APB2ENR_ADC1EN** ((uint32_t)0x00000100)
- #define **RCC_APB2ENR_ADC2EN** ((uint32_t)0x00000200)
- #define **RCC_APB2ENR_ADC3EN** ((uint32_t)0x00000400)
- #define **RCC_APB2ENR_SDIOEN** ((uint32_t)0x00000800)
- #define **RCC_APB2ENR_SPI1EN** ((uint32_t)0x00001000)
- #define **RCC_APB2ENR_SPI4EN** ((uint32_t)0x00002000)
- #define **RCC_APB2ENR_SYSCFGEN** ((uint32_t)0x00004000)
- #define **RCC_APB2ENR_EXTIEN** ((uint32_t)0x00008000)
- #define **RCC_APB2ENR_TIM9EN** ((uint32_t)0x00010000)
- #define **RCC_APB2ENR_TIM10EN** ((uint32_t)0x00020000)
- #define **RCC_APB2ENR_TIM11EN** ((uint32_t)0x00040000)
- #define **RCC_APB2ENR_SPI5EN** ((uint32_t)0x00100000)
- #define **RCC_APB2ENR_SPI6EN** ((uint32_t)0x00200000)
- #define **RCC_APB2ENR_SAI1EN** ((uint32_t)0x00400000)
- #define **RCC_APB2ENR_LTDCEN** ((uint32_t)0x04000000)
- #define **RCC_AHB1LPENR_GPIOALPEN** ((uint32_t)0x00000001)
- #define **RCC_AHB1LPENR_GPIOBLPEN** ((uint32_t)0x00000002)
- #define **RCC_AHB1LPENR_GPIOCLPEN** ((uint32_t)0x00000004)
- #define **RCC_AHB1LPENR_GPIOODLPEN** ((uint32_t)0x00000008)
- #define **RCC_AHB1LPENR_GPIOELPEN** ((uint32_t)0x00000010)
- #define **RCC_AHB1LPENR_GPIOFLPEN** ((uint32_t)0x00000020)
- #define **RCC_AHB1LPENR_GPIOGLPEN** ((uint32_t)0x00000040)
- #define **RCC_AHB1LPENR_GPIOHLPEN** ((uint32_t)0x00000080)
- #define **RCC_AHB1LPENR_GPIOILPEN** ((uint32_t)0x00000100)
- #define **RCC_AHB1LPENR_GPIOJLPEN** ((uint32_t)0x00000200)
- #define **RCC_AHB1LPENR_GPIOOKLPEN** ((uint32_t)0x00000400)
- #define **RCC_AHB1LPENR_CRCLPEN** ((uint32_t)0x00001000)
- #define **RCC_AHB1LPENR_FLITFLPEN** ((uint32_t)0x00008000)
- #define **RCC_AHB1LPENR_SRAM1LPEN** ((uint32_t)0x00010000)
- #define **RCC_AHB1LPENR_SRAM2LPEN** ((uint32_t)0x00020000)
- #define **RCC_AHB1LPENR_BKPSRAMLPEN** ((uint32_t)0x00040000)
- #define **RCC_AHB1LPENR_SRAM3LPEN** ((uint32_t)0x00080000)
- #define **RCC_AHB1LPENR_DMA1LPEN** ((uint32_t)0x00200000)
- #define **RCC_AHB1LPENR_DMA2LPEN** ((uint32_t)0x00400000)
- #define **RCC_AHB1LPENR_DMA2DLPEN** ((uint32_t)0x00800000)
- #define **RCC_AHB1LPENR_ETHMACLPEN** ((uint32_t)0x02000000)
- #define **RCC_AHB1LPENR_ETHMACTXLPEN** ((uint32_t)0x04000000)
- #define **RCC_AHB1LPENR_ETHMACRXLPEN** ((uint32_t)0x08000000)
- #define **RCC_AHB1LPENR_ETHMACPTPLPEN** ((uint32_t)0x10000000)
- #define **RCC_AHB1LPENR_OTGHSLPEN** ((uint32_t)0x20000000)
- #define **RCC_AHB1LPENR_OTGHSULPILPEN** ((uint32_t)0x40000000)

- #define **RCC_AHB2LPENR_DCMILPEN** ((uint32_t)0x00000001)
- #define **RCC_AHB2LPENR_CRYPLPEN** ((uint32_t)0x00000010)
- #define **RCC_AHB2LPENR_HASHLPEN** ((uint32_t)0x00000020)
- #define **RCC_AHB2LPENR_RNGLPEN** ((uint32_t)0x00000040)
- #define **RCC_AHB2LPENR_OTGFSLPEN** ((uint32_t)0x00000080)
- #define **RCC_APB1LPENR_TIM2LPEN** ((uint32_t)0x00000001)
- #define **RCC_APB1LPENR_TIM3LPEN** ((uint32_t)0x00000002)
- #define **RCC_APB1LPENR_TIM4LPEN** ((uint32_t)0x00000004)
- #define **RCC_APB1LPENR_TIM5LPEN** ((uint32_t)0x00000008)
- #define **RCC_APB1LPENR_TIM6LPEN** ((uint32_t)0x00000010)
- #define **RCC_APB1LPENR_TIM7LPEN** ((uint32_t)0x00000020)
- #define **RCC_APB1LPENR_TIM12LPEN** ((uint32_t)0x00000040)
- #define **RCC_APB1LPENR_TIM13LPEN** ((uint32_t)0x00000080)
- #define **RCC_APB1LPENR_TIM14LPEN** ((uint32_t)0x00000100)
- #define **RCC_APB1LPENR_WWDGLPEN** ((uint32_t)0x00000800)
- #define **RCC_APB1LPENR_SPI2LPEN** ((uint32_t)0x00004000)
- #define **RCC_APB1LPENR_SPI3LPEN** ((uint32_t)0x00008000)
- #define **RCC_APB1LPENR_USART2LPEN** ((uint32_t)0x00020000)
- #define **RCC_APB1LPENR_USART3LPEN** ((uint32_t)0x00040000)
- #define **RCC_APB1LPENR_UART4LPEN** ((uint32_t)0x00080000)
- #define **RCC_APB1LPENR_UART5LPEN** ((uint32_t)0x00100000)
- #define **RCC_APB1LPENR_I2C1LPEN** ((uint32_t)0x00200000)
- #define **RCC_APB1LPENR_I2C2LPEN** ((uint32_t)0x00400000)
- #define **RCC_APB1LPENR_I2C3LPEN** ((uint32_t)0x00800000)
- #define **RCC_APB1LPENR_CAN1LPEN** ((uint32_t)0x02000000)
- #define **RCC_APB1LPENR_CAN2LPEN** ((uint32_t)0x04000000)
- #define **RCC_APB1LPENR_PWRLPEN** ((uint32_t)0x10000000)
- #define **RCC_APB1LPENR_DACLPEN** ((uint32_t)0x20000000)
- #define **RCC_APB1LPENR_UART7LPEN** ((uint32_t)0x40000000)
- #define **RCC_APB1LPENR_UART8LPEN** ((uint32_t)0x80000000)
- #define **RCC_APB2LPENR_TIM1LPEN** ((uint32_t)0x00000001)
- #define **RCC_APB2LPENR_TIM8LPEN** ((uint32_t)0x00000002)
- #define **RCC_APB2LPENR_USART1LPEN** ((uint32_t)0x00000010)
- #define **RCC_APB2LPENR_USART6LPEN** ((uint32_t)0x00000020)
- #define **RCC_APB2LPENR_UART9LPEN** ((uint32_t)0x00000040)
- #define **RCC_APB2LPENR_UART10LPEN** ((uint32_t)0x00000080)
- #define **RCC_APB2LPENR_ADC1LPEN** ((uint32_t)0x00000100)
- #define **RCC_APB2LPENR_ADC2PEN** ((uint32_t)0x00000200)
- #define **RCC_APB2LPENR_ADC3LPEN** ((uint32_t)0x00000400)
- #define **RCC_APB2LPENR_SDIOLPEN** ((uint32_t)0x00000800)
- #define **RCC_APB2LPENR_SPI1LPEN** ((uint32_t)0x00001000)
- #define **RCC_APB2LPENR_SPI4LPEN** ((uint32_t)0x00002000)
- #define **RCC_APB2LPENR_SYSCFGLPEN** ((uint32_t)0x00004000)
- #define **RCC_APB2LPENR_TIM9LPEN** ((uint32_t)0x00010000)
- #define **RCC_APB2LPENR_TIM10LPEN** ((uint32_t)0x00020000)
- #define **RCC_APB2LPENR_TIM11LPEN** ((uint32_t)0x00040000)
- #define **RCC_APB2LPENR_SPI5LPEN** ((uint32_t)0x00100000)
- #define **RCC_APB2LPENR_SPI6LPEN** ((uint32_t)0x00200000)
- #define **RCC_APB2LPENR_SAI1LPEN** ((uint32_t)0x00400000)
- #define **RCC_APB2LPENR_LTDCLPEN** ((uint32_t)0x04000000)
- #define **RCC_BDCR_LSEON** ((uint32_t)0x00000001)
- #define **RCC_BDCR_LSERDY** ((uint32_t)0x00000002)
- #define **RCC_BDCR_LSEBYP** ((uint32_t)0x00000004)
- #define **RCC_BDCR_LSEMOP** ((uint32_t)0x00000008)
- #define **RCC_BDCR_RTCSEL** ((uint32_t)0x00000300)

- #define **RCC_BDCR_RTCSEL_0** ((uint32_t)0x00000100)
- #define **RCC_BDCR_RTCSEL_1** ((uint32_t)0x00000200)
- #define **RCC_BDCR_RTCEN** ((uint32_t)0x00008000)
- #define **RCC_BDCR_BDRST** ((uint32_t)0x00010000)
- #define **RCC_CSR_LSION** ((uint32_t)0x00000001)
- #define **RCC_CSR_LSIRDY** ((uint32_t)0x00000002)
- #define **RCC_CSR_RMVF** ((uint32_t)0x01000000)
- #define **RCC_CSR_BORRSTF** ((uint32_t)0x02000000)
- #define **RCC_CSR_PADRSTF** ((uint32_t)0x04000000)
- #define **RCC_CSR_PORRSTF** ((uint32_t)0x08000000)
- #define **RCC_CSR_SFTRSTF** ((uint32_t)0x10000000)
- #define **RCC_CSR_WDGRSTF** ((uint32_t)0x20000000)
- #define **RCC_CSR_WWDGRSTF** ((uint32_t)0x40000000)
- #define **RCC_CSR_LPWRRSTF** ((uint32_t)0x80000000)
- #define **RCC_SSCGR_MODPER** ((uint32_t)0x00001FFF)
- #define **RCC_SSCGR_INCSTEP** ((uint32_t)0x0FFE000)
- #define **RCC_SSCGR_SPREADSEL** ((uint32_t)0x40000000)
- #define **RCC_SSCGR_SSCGEN** ((uint32_t)0x80000000)
- #define **RCC_PLLI2SCFGR_PLLI2SM** ((uint32_t)0x0000003F)
- #define **RCC_PLLI2SCFGR_PLLI2SM_0** ((uint32_t)0x00000001)
- #define **RCC_PLLI2SCFGR_PLLI2SM_1** ((uint32_t)0x00000002)
- #define **RCC_PLLI2SCFGR_PLLI2SM_2** ((uint32_t)0x00000004)
- #define **RCC_PLLI2SCFGR_PLLI2SM_3** ((uint32_t)0x00000008)
- #define **RCC_PLLI2SCFGR_PLLI2SM_4** ((uint32_t)0x00000010)
- #define **RCC_PLLI2SCFGR_PLLI2SM_5** ((uint32_t)0x00000020)
- #define **RCC_PLLI2SCFGR_PLLI2SN** ((uint32_t)0x00007FC0)
- #define **RCC_PLLI2SCFGR_PLLI2SN_0** ((uint32_t)0x00000040)
- #define **RCC_PLLI2SCFGR_PLLI2SN_1** ((uint32_t)0x00000080)
- #define **RCC_PLLI2SCFGR_PLLI2SN_2** ((uint32_t)0x00000100)
- #define **RCC_PLLI2SCFGR_PLLI2SN_3** ((uint32_t)0x00000200)
- #define **RCC_PLLI2SCFGR_PLLI2SN_4** ((uint32_t)0x00000400)
- #define **RCC_PLLI2SCFGR_PLLI2SN_5** ((uint32_t)0x00000800)
- #define **RCC_PLLI2SCFGR_PLLI2SN_6** ((uint32_t)0x00001000)
- #define **RCC_PLLI2SCFGR_PLLI2SN_7** ((uint32_t)0x00002000)
- #define **RCC_PLLI2SCFGR_PLLI2SN_8** ((uint32_t)0x00004000)
- #define **RCC_PLLI2SCFGR_PLLI2SQ** ((uint32_t)0x0F000000)
- #define **RCC_PLLI2SCFGR_PLLI2SQ_0** ((uint32_t)0x01000000)
- #define **RCC_PLLI2SCFGR_PLLI2SQ_1** ((uint32_t)0x02000000)
- #define **RCC_PLLI2SCFGR_PLLI2SQ_2** ((uint32_t)0x04000000)
- #define **RCC_PLLI2SCFGR_PLLI2SQ_3** ((uint32_t)0x08000000)
- #define **RCC_PLLI2SCFGR_PLLI2SR** ((uint32_t)0x70000000)
- #define **RCC_PLLI2SCFGR_PLLI2SR_0** ((uint32_t)0x10000000)
- #define **RCC_PLLI2SCFGR_PLLI2SR_1** ((uint32_t)0x20000000)
- #define **RCC_PLLI2SCFGR_PLLI2SR_2** ((uint32_t)0x40000000)
- #define **RCC_PLLSAICFGR_PLLSAIN** ((uint32_t)0x00007FC0)
- #define **RCC_PLLSAICFGR_PLLSAIN_0** ((uint32_t)0x00000040)
- #define **RCC_PLLSAICFGR_PLLSAIN_1** ((uint32_t)0x00000080)
- #define **RCC_PLLSAICFGR_PLLSAIN_2** ((uint32_t)0x00000100)
- #define **RCC_PLLSAICFGR_PLLSAIN_3** ((uint32_t)0x00000200)
- #define **RCC_PLLSAICFGR_PLLSAIN_4** ((uint32_t)0x00000400)
- #define **RCC_PLLSAICFGR_PLLSAIN_5** ((uint32_t)0x00000800)
- #define **RCC_PLLSAICFGR_PLLSAIN_6** ((uint32_t)0x00001000)
- #define **RCC_PLLSAICFGR_PLLSAIN_7** ((uint32_t)0x00002000)
- #define **RCC_PLLSAICFGR_PLLSAIN_8** ((uint32_t)0x00004000)
- #define **RCC_PLLSAICFGR_PLLSAIQ** ((uint32_t)0x0F000000)

- #define **RCC_PLLSAICFGR_PLLSAIQ_0** ((uint32_t)0x01000000)
- #define **RCC_PLLSAICFGR_PLLSAIQ_1** ((uint32_t)0x02000000)
- #define **RCC_PLLSAICFGR_PLLSAIQ_2** ((uint32_t)0x04000000)
- #define **RCC_PLLSAICFGR_PLLSAIQ_3** ((uint32_t)0x08000000)
- #define **RCC_PLLSAICFGR_PLLSAIR** ((uint32_t)0x70000000)
- #define **RCC_PLLSAICFGR_PLLSAIR_0** ((uint32_t)0x10000000)
- #define **RCC_PLLSAICFGR_PLLSAIR_1** ((uint32_t)0x20000000)
- #define **RCC_PLLSAICFGR_PLLSAIR_2** ((uint32_t)0x40000000)
- #define **RCC_DCKCFGR_PLLI2SDIVQ** ((uint32_t)0x00000001F)
- #define **RCC_DCKCFGR_PLLSAIDIVQ** ((uint32_t)0x00001F00)
- #define **RCC_DCKCFGR_PLLSAIDIVR** ((uint32_t)0x00030000)
- #define **RCC_DCKCFGR_SAI1ASRC** ((uint32_t)0x00300000)
- #define **RCC_DCKCFGR_SAI1ASRC_0** ((uint32_t)0x00100000)
- #define **RCC_DCKCFGR_SAI1ASRC_1** ((uint32_t)0x00200000)
- #define **RCC_DCKCFGR_SAI1BSRC** ((uint32_t)0x00C00000)
- #define **RCC_DCKCFGR_SAI1BSRC_0** ((uint32_t)0x00400000)
- #define **RCC_DCKCFGR_SAI1BSRC_1** ((uint32_t)0x00800000)
- #define **RCC_DCKCFGR_TIMPRE** ((uint32_t)0x01000000)
- #define **RNG_CR_RNGEN** ((uint32_t)0x00000004)
- #define **RNG_CR_IE** ((uint32_t)0x00000008)
- #define **RNG_SR_DRDY** ((uint32_t)0x00000001)
- #define **RNG_SR_CECS** ((uint32_t)0x00000002)
- #define **RNG_SR_SECS** ((uint32_t)0x00000004)
- #define **RNG_SR_CEIS** ((uint32_t)0x00000020)
- #define **RNG_SR_SEIS** ((uint32_t)0x00000040)
- #define **RTC_TR_PM** ((uint32_t)0x00400000)
- #define **RTC_TR_HT** ((uint32_t)0x00300000)
- #define **RTC_TR_HT_0** ((uint32_t)0x00100000)
- #define **RTC_TR_HT_1** ((uint32_t)0x00200000)
- #define **RTC_TR_HU** ((uint32_t)0x000F0000)
- #define **RTC_TR_HU_0** ((uint32_t)0x00010000)
- #define **RTC_TR_HU_1** ((uint32_t)0x00020000)
- #define **RTC_TR_HU_2** ((uint32_t)0x00040000)
- #define **RTC_TR_HU_3** ((uint32_t)0x00080000)
- #define **RTC_TR_MNT** ((uint32_t)0x00007000)
- #define **RTC_TR_MNT_0** ((uint32_t)0x00001000)
- #define **RTC_TR_MNT_1** ((uint32_t)0x00002000)
- #define **RTC_TR_MNT_2** ((uint32_t)0x00004000)
- #define **RTC_TR_MNU** ((uint32_t)0x00000F00)
- #define **RTC_TR_MNU_0** ((uint32_t)0x00000100)
- #define **RTC_TR_MNU_1** ((uint32_t)0x00000200)
- #define **RTC_TR_MNU_2** ((uint32_t)0x00000400)
- #define **RTC_TR_MNU_3** ((uint32_t)0x00000800)
- #define **RTC_TR_ST** ((uint32_t)0x00000070)
- #define **RTC_TR_ST_0** ((uint32_t)0x00000010)
- #define **RTC_TR_ST_1** ((uint32_t)0x00000020)
- #define **RTC_TR_ST_2** ((uint32_t)0x00000040)
- #define **RTC_TR_SU** ((uint32_t)0x0000000F)
- #define **RTC_TR_SU_0** ((uint32_t)0x00000001)
- #define **RTC_TR_SU_1** ((uint32_t)0x00000002)
- #define **RTC_TR_SU_2** ((uint32_t)0x00000004)
- #define **RTC_TR_SU_3** ((uint32_t)0x00000008)
- #define **RTC_DR_YT** ((uint32_t)0x00F00000)
- #define **RTC_DR_YT_0** ((uint32_t)0x00100000)
- #define **RTC_DR_YT_1** ((uint32_t)0x00200000)

- #define **RTC_DR_YT_2** ((uint32_t)0x00400000)
- #define **RTC_DR_YT_3** ((uint32_t)0x00800000)
- #define **RTC_DR_YU** ((uint32_t)0x000F0000)
- #define **RTC_DR_YU_0** ((uint32_t)0x00010000)
- #define **RTC_DR_YU_1** ((uint32_t)0x00020000)
- #define **RTC_DR_YU_2** ((uint32_t)0x00040000)
- #define **RTC_DR_YU_3** ((uint32_t)0x00080000)
- #define **RTC_DR_WDU** ((uint32_t)0x0000E000)
- #define **RTC_DR_WDU_0** ((uint32_t)0x00002000)
- #define **RTC_DR_WDU_1** ((uint32_t)0x00004000)
- #define **RTC_DR_WDU_2** ((uint32_t)0x00008000)
- #define **RTC_DR_MT** ((uint32_t)0x00001000)
- #define **RTC_DR_MU** ((uint32_t)0x00000F00)
- #define **RTC_DR_MU_0** ((uint32_t)0x00000100)
- #define **RTC_DR_MU_1** ((uint32_t)0x00000200)
- #define **RTC_DR_MU_2** ((uint32_t)0x00000400)
- #define **RTC_DR_MU_3** ((uint32_t)0x00000800)
- #define **RTC_DR_DT** ((uint32_t)0x00000030)
- #define **RTC_DR_DT_0** ((uint32_t)0x00000010)
- #define **RTC_DR_DT_1** ((uint32_t)0x00000020)
- #define **RTC_DR_DU** ((uint32_t)0x0000000F)
- #define **RTC_DR_DU_0** ((uint32_t)0x00000001)
- #define **RTC_DR_DU_1** ((uint32_t)0x00000002)
- #define **RTC_DR_DU_2** ((uint32_t)0x00000004)
- #define **RTC_DR_DU_3** ((uint32_t)0x00000008)
- #define **RTC_CR_COE** ((uint32_t)0x00800000)
- #define **RTC_CR_OSEL** ((uint32_t)0x00600000)
- #define **RTC_CR_OSEL_0** ((uint32_t)0x00200000)
- #define **RTC_CR_OSEL_1** ((uint32_t)0x00400000)
- #define **RTC_CR_POL** ((uint32_t)0x00100000)
- #define **RTC_CR_COSEL** ((uint32_t)0x00080000)
- #define **RTC_CR_BCK** ((uint32_t)0x00040000)
- #define **RTC_CR_SUB1H** ((uint32_t)0x00020000)
- #define **RTC_CR_ADD1H** ((uint32_t)0x00010000)
- #define **RTC_CR_TSIE** ((uint32_t)0x00008000)
- #define **RTC_CR_WUTIE** ((uint32_t)0x00004000)
- #define **RTC_CR_ALRBIE** ((uint32_t)0x00002000)
- #define **RTC_CR_ALRAIE** ((uint32_t)0x00001000)
- #define **RTC_CR_TSE** ((uint32_t)0x00000800)
- #define **RTC_CR_WUTE** ((uint32_t)0x00000400)
- #define **RTC_CR_ALRBE** ((uint32_t)0x00000200)
- #define **RTC_CR_ALRAE** ((uint32_t)0x00000100)
- #define **RTC_CR_DCE** ((uint32_t)0x00000080)
- #define **RTC_CR_FMT** ((uint32_t)0x00000040)
- #define **RTC_CR_BYPSHAD** ((uint32_t)0x00000020)
- #define **RTC_CR_REFCKON** ((uint32_t)0x00000010)
- #define **RTC_CR_TSEDGE** ((uint32_t)0x00000008)
- #define **RTC_CR_WUCKSEL** ((uint32_t)0x00000007)
- #define **RTC_CR_WUCKSEL_0** ((uint32_t)0x00000001)
- #define **RTC_CR_WUCKSEL_1** ((uint32_t)0x00000002)
- #define **RTC_CR_WUCKSEL_2** ((uint32_t)0x00000004)
- #define **RTC_ISR_RECALPF** ((uint32_t)0x00010000)
- #define **RTC_ISR_TAMP1F** ((uint32_t)0x00002000)
- #define **RTC_ISR_TAMP2F** ((uint32_t)0x00004000)
- #define **RTC_ISR_TSOVF** ((uint32_t)0x00001000)

- #define **RTC_ISR_TSF** ((uint32_t)0x00000800)
- #define **RTC_ISR_WUTF** ((uint32_t)0x00000400)
- #define **RTC_ISR_ALRBF** ((uint32_t)0x00000200)
- #define **RTC_ISR_ALRAF** ((uint32_t)0x00000100)
- #define **RTC_ISR_INIT** ((uint32_t)0x00000080)
- #define **RTC_ISR_INITF** ((uint32_t)0x00000040)
- #define **RTC_ISR_RSF** ((uint32_t)0x00000020)
- #define **RTC_ISR_INITS** ((uint32_t)0x00000010)
- #define **RTC_ISR_SHPF** ((uint32_t)0x00000008)
- #define **RTC_ISR_WUTWF** ((uint32_t)0x00000004)
- #define **RTC_ISR_ALRBWF** ((uint32_t)0x00000002)
- #define **RTC_ISR_ALRAWF** ((uint32_t)0x00000001)
- #define **RTC_PRER_PREDIV_A** ((uint32_t)0x007F0000)
- #define **RTC_PRER_PREDIV_S** ((uint32_t)0x00001FFF)
- #define **RTC_WUTR_WUT** ((uint32_t)0x0000FFFF)
- #define **RTC_CALIBR_DCS** ((uint32_t)0x00000080)
- #define **RTC_CALIBR_DC** ((uint32_t)0x0000001F)
- #define **RTC_ALRMAR_MSK4** ((uint32_t)0x80000000)
- #define **RTC_ALRMAR_WDSEL** ((uint32_t)0x40000000)
- #define **RTC_ALRMAR_DT** ((uint32_t)0x30000000)
- #define **RTC_ALRMAR_DT_0** ((uint32_t)0x10000000)
- #define **RTC_ALRMAR_DT_1** ((uint32_t)0x20000000)
- #define **RTC_ALRMAR_DU** ((uint32_t)0x0F000000)
- #define **RTC_ALRMAR_DU_0** ((uint32_t)0x01000000)
- #define **RTC_ALRMAR_DU_1** ((uint32_t)0x02000000)
- #define **RTC_ALRMAR_DU_2** ((uint32_t)0x04000000)
- #define **RTC_ALRMAR_DU_3** ((uint32_t)0x08000000)
- #define **RTC_ALRMAR_MSK3** ((uint32_t)0x00800000)
- #define **RTC_ALRMAR_PM** ((uint32_t)0x00400000)
- #define **RTC_ALRMAR_HT** ((uint32_t)0x00300000)
- #define **RTC_ALRMAR_HT_0** ((uint32_t)0x00100000)
- #define **RTC_ALRMAR_HT_1** ((uint32_t)0x00200000)
- #define **RTC_ALRMAR_HU** ((uint32_t)0x000F0000)
- #define **RTC_ALRMAR_HU_0** ((uint32_t)0x00010000)
- #define **RTC_ALRMAR_HU_1** ((uint32_t)0x00020000)
- #define **RTC_ALRMAR_HU_2** ((uint32_t)0x00040000)
- #define **RTC_ALRMAR_HU_3** ((uint32_t)0x00080000)
- #define **RTC_ALRMAR_MSK2** ((uint32_t)0x00008000)
- #define **RTC_ALRMAR_MNT** ((uint32_t)0x00007000)
- #define **RTC_ALRMAR_MNT_0** ((uint32_t)0x00001000)
- #define **RTC_ALRMAR_MNT_1** ((uint32_t)0x00002000)
- #define **RTC_ALRMAR_MNT_2** ((uint32_t)0x00004000)
- #define **RTC_ALRMAR_MNU** ((uint32_t)0x0000F00)
- #define **RTC_ALRMAR_MNU_0** ((uint32_t)0x0000100)
- #define **RTC_ALRMAR_MNU_1** ((uint32_t)0x00000200)
- #define **RTC_ALRMAR_MNU_2** ((uint32_t)0x00000400)
- #define **RTC_ALRMAR_MNU_3** ((uint32_t)0x00000800)
- #define **RTC_ALRMAR_MSK1** ((uint32_t)0x00000080)
- #define **RTC_ALRMAR_ST** ((uint32_t)0x00000070)
- #define **RTC_ALRMAR_ST_0** ((uint32_t)0x00000010)
- #define **RTC_ALRMAR_ST_1** ((uint32_t)0x00000020)
- #define **RTC_ALRMAR_ST_2** ((uint32_t)0x00000040)
- #define **RTC_ALRMAR_SU** ((uint32_t)0x0000000F)
- #define **RTC_ALRMAR_SU_0** ((uint32_t)0x00000001)
- #define **RTC_ALRMAR_SU_1** ((uint32_t)0x00000002)

- #define **RTC_ALRMAR_SU_2** ((uint32_t)0x00000004)
- #define **RTC_ALRMAR_SU_3** ((uint32_t)0x00000008)
- #define **RTC_ALRMBR_MSK4** ((uint32_t)0x80000000)
- #define **RTC_ALRMBR_WDSEL** ((uint32_t)0x40000000)
- #define **RTC_ALRMBR_DT** ((uint32_t)0x30000000)
- #define **RTC_ALRMBR_DT_0** ((uint32_t)0x10000000)
- #define **RTC_ALRMBR_DT_1** ((uint32_t)0x20000000)
- #define **RTC_ALRMBR_DU** ((uint32_t)0x0F000000)
- #define **RTC_ALRMBR_DU_0** ((uint32_t)0x01000000)
- #define **RTC_ALRMBR_DU_1** ((uint32_t)0x02000000)
- #define **RTC_ALRMBR_DU_2** ((uint32_t)0x04000000)
- #define **RTC_ALRMBR_DU_3** ((uint32_t)0x08000000)
- #define **RTC_ALRMBR_MSK3** ((uint32_t)0x00800000)
- #define **RTC_ALRMBR_PM** ((uint32_t)0x00400000)
- #define **RTC_ALRMBR_HT** ((uint32_t)0x00300000)
- #define **RTC_ALRMBR_HT_0** ((uint32_t)0x00100000)
- #define **RTC_ALRMBR_HT_1** ((uint32_t)0x00200000)
- #define **RTC_ALRMBR_HU** ((uint32_t)0x000F0000)
- #define **RTC_ALRMBR_HU_0** ((uint32_t)0x00010000)
- #define **RTC_ALRMBR_HU_1** ((uint32_t)0x00020000)
- #define **RTC_ALRMBR_HU_2** ((uint32_t)0x00040000)
- #define **RTC_ALRMBR_HU_3** ((uint32_t)0x00080000)
- #define **RTC_ALRMBR_MSK2** ((uint32_t)0x00008000)
- #define **RTC_ALRMBR_MNT** ((uint32_t)0x00007000)
- #define **RTC_ALRMBR_MNT_0** ((uint32_t)0x00001000)
- #define **RTC_ALRMBR_MNT_1** ((uint32_t)0x00002000)
- #define **RTC_ALRMBR_MNT_2** ((uint32_t)0x00004000)
- #define **RTC_ALRMBR_MNU** ((uint32_t)0x00000F00)
- #define **RTC_ALRMBR_MNU_0** ((uint32_t)0x00000100)
- #define **RTC_ALRMBR_MNU_1** ((uint32_t)0x00000200)
- #define **RTC_ALRMBR_MNU_2** ((uint32_t)0x00000400)
- #define **RTC_ALRMBR_MNU_3** ((uint32_t)0x00000800)
- #define **RTC_ALRMBR_MSK1** ((uint32_t)0x00000080)
- #define **RTC_ALRMBR_ST** ((uint32_t)0x00000070)
- #define **RTC_ALRMBR_ST_0** ((uint32_t)0x00000010)
- #define **RTC_ALRMBR_ST_1** ((uint32_t)0x00000020)
- #define **RTC_ALRMBR_ST_2** ((uint32_t)0x00000040)
- #define **RTC_ALRMBR_SU** ((uint32_t)0x0000000F)
- #define **RTC_ALRMBR_SU_0** ((uint32_t)0x00000001)
- #define **RTC_ALRMBR_SU_1** ((uint32_t)0x00000002)
- #define **RTC_ALRMBR_SU_2** ((uint32_t)0x00000004)
- #define **RTC_ALRMBR_SU_3** ((uint32_t)0x00000008)
- #define **RTC_WPR_KEY** ((uint32_t)0x000000FF)
- #define **RTC_SSR_SS** ((uint32_t)0x0000FFFF)
- #define **RTC_SHIFTR_SUBFS** ((uint32_t)0x00007FFF)
- #define **RTC_SHIFTR_ADD1S** ((uint32_t)0x80000000)
- #define **RTC_TSTR_PM** ((uint32_t)0x00400000)
- #define **RTC_TSTR_HT** ((uint32_t)0x00300000)
- #define **RTC_TSTR_HT_0** ((uint32_t)0x00100000)
- #define **RTC_TSTR_HT_1** ((uint32_t)0x00200000)
- #define **RTC_TSTR_HU** ((uint32_t)0x000F0000)
- #define **RTC_TSTR_HU_0** ((uint32_t)0x00010000)
- #define **RTC_TSTR_HU_1** ((uint32_t)0x00020000)
- #define **RTC_TSTR_HU_2** ((uint32_t)0x00040000)
- #define **RTC_TSTR_HU_3** ((uint32_t)0x00080000)

- #define **RTC_TSTR_MNT** ((uint32_t)0x00007000)
- #define **RTC_TSTR_MNT_0** ((uint32_t)0x00001000)
- #define **RTC_TSTR_MNT_1** ((uint32_t)0x00002000)
- #define **RTC_TSTR_MNT_2** ((uint32_t)0x00004000)
- #define **RTC_TSTR_MNU** ((uint32_t)0x00000F00)
- #define **RTC_TSTR_MNU_0** ((uint32_t)0x00000100)
- #define **RTC_TSTR_MNU_1** ((uint32_t)0x00000200)
- #define **RTC_TSTR_MNU_2** ((uint32_t)0x00000400)
- #define **RTC_TSTR_MNU_3** ((uint32_t)0x00000800)
- #define **RTC_TSTR_ST** ((uint32_t)0x00000070)
- #define **RTC_TSTR_ST_0** ((uint32_t)0x00000010)
- #define **RTC_TSTR_ST_1** ((uint32_t)0x00000020)
- #define **RTC_TSTR_ST_2** ((uint32_t)0x00000040)
- #define **RTC_TSTR_SU** ((uint32_t)0x0000000F)
- #define **RTC_TSTR_SU_0** ((uint32_t)0x00000001)
- #define **RTC_TSTR_SU_1** ((uint32_t)0x00000002)
- #define **RTC_TSTR_SU_2** ((uint32_t)0x00000004)
- #define **RTC_TSTR_SU_3** ((uint32_t)0x00000008)
- #define **RTC_TSDR_WDU** ((uint32_t)0x0000E000)
- #define **RTC_TSDR_WDU_0** ((uint32_t)0x00002000)
- #define **RTC_TSDR_WDU_1** ((uint32_t)0x00004000)
- #define **RTC_TSDR_WDU_2** ((uint32_t)0x00008000)
- #define **RTC_TSDR_MT** ((uint32_t)0x00001000)
- #define **RTC_TSDR_MU** ((uint32_t)0x00000F00)
- #define **RTC_TSDR_MU_0** ((uint32_t)0x00000100)
- #define **RTC_TSDR_MU_1** ((uint32_t)0x00000200)
- #define **RTC_TSDR_MU_2** ((uint32_t)0x00000400)
- #define **RTC_TSDR_MU_3** ((uint32_t)0x00000800)
- #define **RTC_TSDR_DT** ((uint32_t)0x00000030)
- #define **RTC_TSDR_DT_0** ((uint32_t)0x00000010)
- #define **RTC_TSDR_DT_1** ((uint32_t)0x00000020)
- #define **RTC_TSDR_DU** ((uint32_t)0x0000000F)
- #define **RTC_TSDR_DU_0** ((uint32_t)0x00000001)
- #define **RTC_TSDR_DU_1** ((uint32_t)0x00000002)
- #define **RTC_TSDR_DU_2** ((uint32_t)0x00000004)
- #define **RTC_TSDR_DU_3** ((uint32_t)0x00000008)
- #define **RTC_TSSSR_SS** ((uint32_t)0x0000FFFF)
- #define **RTC_CALR_CALP** ((uint32_t)0x00008000)
- #define **RTC_CALR_CALW8** ((uint32_t)0x00004000)
- #define **RTC_CALR_CALW16** ((uint32_t)0x00002000)
- #define **RTC_CALR_CALM** ((uint32_t)0x000001FF)
- #define **RTC_CALR_CALM_0** ((uint32_t)0x00000001)
- #define **RTC_CALR_CALM_1** ((uint32_t)0x00000002)
- #define **RTC_CALR_CALM_2** ((uint32_t)0x00000004)
- #define **RTC_CALR_CALM_3** ((uint32_t)0x00000008)
- #define **RTC_CALR_CALM_4** ((uint32_t)0x00000010)
- #define **RTC_CALR_CALM_5** ((uint32_t)0x00000020)
- #define **RTC_CALR_CALM_6** ((uint32_t)0x00000040)
- #define **RTC_CALR_CALM_7** ((uint32_t)0x00000080)
- #define **RTC_CALR_CALM_8** ((uint32_t)0x00000100)
- #define **RTC_TAFCR_ALARMOUTTYPE** ((uint32_t)0x00040000)
- #define **RTC_TAFCR_TSINSEL** ((uint32_t)0x00020000)
- #define **RTC_TAFCR_TAMPINSEL** ((uint32_t)0x00010000)
- #define **RTC_TAFCR_TAMPPUDIS** ((uint32_t)0x00008000)
- #define **RTC_TAFCR_TAMPPRCH** ((uint32_t)0x00006000)

- #define **RTC_TAFCR_TAMPPRCH_0** ((uint32_t)0x00002000)
- #define **RTC_TAFCR_TAMPPRCH_1** ((uint32_t)0x00004000)
- #define **RTC_TAFCR_TAMPFLT** ((uint32_t)0x00001800)
- #define **RTC_TAFCR_TAMPFLT_0** ((uint32_t)0x00000800)
- #define **RTC_TAFCR_TAMPFLT_1** ((uint32_t)0x00001000)
- #define **RTC_TAFCR_TAMPFREQ** ((uint32_t)0x00000700)
- #define **RTC_TAFCR_TAMPFREQ_0** ((uint32_t)0x00000100)
- #define **RTC_TAFCR_TAMPFREQ_1** ((uint32_t)0x00000200)
- #define **RTC_TAFCR_TAMPFREQ_2** ((uint32_t)0x00000400)
- #define **RTC_TAFCR_TAMPTS** ((uint32_t)0x00000080)
- #define **RTC_TAFCR_TAMP2TRG** ((uint32_t)0x00000010)
- #define **RTC_TAFCR_TAMP2E** ((uint32_t)0x00000008)
- #define **RTC_TAFCR_TAMPIE** ((uint32_t)0x00000004)
- #define **RTC_TAFCR_TAMP1TRG** ((uint32_t)0x00000002)
- #define **RTC_TAFCR_TAMP1E** ((uint32_t)0x00000001)
- #define **RTC_ALRMASSR_MASKSS** ((uint32_t)0x0F000000)
- #define **RTC_ALRMASSR_MASKSS_0** ((uint32_t)0x01000000)
- #define **RTC_ALRMASSR_MASKSS_1** ((uint32_t)0x02000000)
- #define **RTC_ALRMASSR_MASKSS_2** ((uint32_t)0x04000000)
- #define **RTC_ALRMASSR_MASKSS_3** ((uint32_t)0x08000000)
- #define **RTC_ALRMASSR_SS** ((uint32_t)0x00007FFF)
- #define **RTC_ALRMBSSR_MASKSS** ((uint32_t)0x0F000000)
- #define **RTC_ALRMBSSR_MASKSS_0** ((uint32_t)0x01000000)
- #define **RTC_ALRMBSSR_MASKSS_1** ((uint32_t)0x02000000)
- #define **RTC_ALRMBSSR_MASKSS_2** ((uint32_t)0x04000000)
- #define **RTC_ALRMBSSR_MASKSS_3** ((uint32_t)0x08000000)
- #define **RTC_ALRMBSSR_SS** ((uint32_t)0x00007FFF)
- #define **RTC_BKP0R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP1R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP2R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP3R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP4R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP5R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP6R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP7R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP8R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP9R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP10R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP11R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP12R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP13R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP14R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP15R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP16R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP17R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP18R** ((uint32_t)0xFFFFFFFF)
- #define **RTC_BKP19R** ((uint32_t)0xFFFFFFFF)
- #define **SAI_GCR_SYNCIN** ((uint32_t)0x00000003)
- #define **SAI_GCR_SYNCIN_0** ((uint32_t)0x00000001)
- #define **SAI_GCR_SYNCIN_1** ((uint32_t)0x00000002)
- #define **SAI_GCR_SYNCOUT** ((uint32_t)0x00000030)
- #define **SAI_GCR_SYNCOUT_0** ((uint32_t)0x00000010)
- #define **SAI_GCR_SYNCOUT_1** ((uint32_t)0x00000020)
- #define **SAI_xCR1_MODE** ((uint32_t)0x00000003)
- #define **SAI_xCR1_MODE_0** ((uint32_t)0x00000001)

```
• #define SAI_xCR1_MODE_1 ((uint32_t)0x00000002)
• #define SAI_xCR1_PRTCFG ((uint32_t)0x0000000C)
• #define SAI_xCR1_PRTCFG_0 ((uint32_t)0x00000004)
• #define SAI_xCR1_PRTCFG_1 ((uint32_t)0x00000008)
• #define SAI_xCR1_DS ((uint32_t)0x000000E0)
• #define SAI_xCR1_DS_0 ((uint32_t)0x00000020)
• #define SAI_xCR1_DS_1 ((uint32_t)0x00000040)
• #define SAI_xCR1_DS_2 ((uint32_t)0x00000080)
• #define SAI_xCR1_LSBFIRST ((uint32_t)0x00000100)
• #define SAI_xCR1_CKSTR ((uint32_t)0x00000200)
• #define SAI_xCR1_SYNCEN ((uint32_t)0x00000C00)
• #define SAI_xCR1_SYNCEN_0 ((uint32_t)0x00000400)
• #define SAI_xCR1_SYNCEN_1 ((uint32_t)0x00000800)
• #define SAI_xCR1_MONO ((uint32_t)0x00001000)
• #define SAI_xCR1_OUTDRIV ((uint32_t)0x00002000)
• #define SAI_xCR1_SAIEN ((uint32_t)0x00010000)
• #define SAI_xCR1_DMAEN ((uint32_t)0x00020000)
• #define SAI_xCR1_NODIV ((uint32_t)0x00080000)
• #define SAI_xCR1_MCKDIV ((uint32_t)0x00780000)
• #define SAI_xCR1_MCKDIV_0 ((uint32_t)0x00080000)
• #define SAI_xCR1_MCKDIV_1 ((uint32_t)0x00100000)
• #define SAI_xCR1_MCKDIV_2 ((uint32_t)0x00200000)
• #define SAI_xCR1_MCKDIV_3 ((uint32_t)0x00400000)
• #define SAI_xCR2_FTH ((uint32_t)0x00000003)
• #define SAI_xCR2_FTH_0 ((uint32_t)0x00000001)
• #define SAI_xCR2_FTH_1 ((uint32_t)0x00000002)
• #define SAI_xCR2_FFLUSH ((uint32_t)0x00000008)
• #define SAI_xCR2_TRIS ((uint32_t)0x00000010)
• #define SAI_xCR2_MUTE ((uint32_t)0x00000020)
• #define SAI_xCR2_MUTEVAL ((uint32_t)0x00000040)
• #define SAI_xCR2_MUTECNT ((uint32_t)0x00001F80)
• #define SAI_xCR2_MUTECNT_0 ((uint32_t)0x00000080)
• #define SAI_xCR2_MUTECNT_1 ((uint32_t)0x00000100)
• #define SAI_xCR2_MUTECNT_2 ((uint32_t)0x00000200)
• #define SAI_xCR2_MUTECNT_3 ((uint32_t)0x00000400)
• #define SAI_xCR2_MUTECNT_4 ((uint32_t)0x00000800)
• #define SAI_xCR2_MUTECNT_5 ((uint32_t)0x00001000)
• #define SAI_xCR2_CPL ((uint32_t)0x00002000)
• #define SAI_xCR2_COMP ((uint32_t)0x0000C000)
• #define SAI_xCR2_COMP_0 ((uint32_t)0x00004000)
• #define SAI_xCR2_COMP_1 ((uint32_t)0x00008000)
• #define SAI_xFRCR_FRL ((uint32_t)0x000000FF)
• #define SAI_xFRCR_FRL_0 ((uint32_t)0x00000001)
• #define SAI_xFRCR_FRL_1 ((uint32_t)0x00000002)
• #define SAI_xFRCR_FRL_2 ((uint32_t)0x00000004)
• #define SAI_xFRCR_FRL_3 ((uint32_t)0x00000008)
• #define SAI_xFRCR_FRL_4 ((uint32_t)0x00000010)
• #define SAI_xFRCR_FRL_5 ((uint32_t)0x00000020)
• #define SAI_xFRCR_FRL_6 ((uint32_t)0x00000040)
• #define SAI_xFRCR_FRL_7 ((uint32_t)0x00000080)
• #define SAI_xFRCR_FSALL ((uint32_t)0x00007F00)
• #define SAI_xFRCR_FSALL_0 ((uint32_t)0x00000100)
• #define SAI_xFRCR_FSALL_1 ((uint32_t)0x00000200)
• #define SAI_xFRCR_FSALL_2 ((uint32_t)0x00000400)
• #define SAI_xFRCR_FSALL_3 ((uint32_t)0x00000800)
```

- #define SAI_xFRCR_FSALL_4 ((uint32_t)0x00001000)
- #define SAI_xFRCR_FSALL_5 ((uint32_t)0x00002000)
- #define SAI_xFRCR_FSALL_6 ((uint32_t)0x00004000)
- #define SAI_xFRCR_FSDEF ((uint32_t)0x00010000)
- #define SAI_xFRCR_FSPOL ((uint32_t)0x00020000)
- #define SAI_xFRCR_FSOFF ((uint32_t)0x00040000)
- #define SAI_xFRCR_FSPO SAI_xFRCR_FSPOL
- #define SAI_xSLOTR_FBOFF ((uint32_t)0x00000001F)
- #define SAI_xSLOTR_FBOFF_0 ((uint32_t)0x000000001)
- #define SAI_xSLOTR_FBOFF_1 ((uint32_t)0x000000002)
- #define SAI_xSLOTR_FBOFF_2 ((uint32_t)0x000000004)
- #define SAI_xSLOTR_FBOFF_3 ((uint32_t)0x000000008)
- #define SAI_xSLOTR_FBOFF_4 ((uint32_t)0x000000010)
- #define SAI_xSLOTR_SLOTsz ((uint32_t)0x000000C0)
- #define SAI_xSLOTR_SLOTsz_0 ((uint32_t)0x00000040)
- #define SAI_xSLOTR_SLOTsz_1 ((uint32_t)0x00000080)
- #define SAI_xSLOTR_NBSLOT ((uint32_t)0x000000F00)
- #define SAI_xSLOTR_NBSLOT_0 ((uint32_t)0x00000100)
- #define SAI_xSLOTR_NBSLOT_1 ((uint32_t)0x00000200)
- #define SAI_xSLOTR_NBSLOT_2 ((uint32_t)0x00000400)
- #define SAI_xSLOTR_NBSLOT_3 ((uint32_t)0x00000800)
- #define SAI_xSLOTR_SLOTEN ((uint32_t)0xFFFF0000)
- #define SAI_xIMR_OVRUDRIE ((uint32_t)0x00000001)
- #define SAI_xIMR_MUTEDETIE ((uint32_t)0x00000002)
- #define SAI_xIMR_WCKCFGIE ((uint32_t)0x00000004)
- #define SAI_xIMR_FREQIE ((uint32_t)0x00000008)
- #define SAI_xIMR_CNRDYIE ((uint32_t)0x00000010)
- #define SAI_xIMR_AFSDETIE ((uint32_t)0x00000020)
- #define SAI_xIMR_LFSDETIE ((uint32_t)0x00000040)
- #define SAI_xSR_OVRUDR ((uint32_t)0x00000001)
- #define SAI_xSR_MUTEDET ((uint32_t)0x00000002)
- #define SAI_xSR_WCKCFG ((uint32_t)0x00000004)
- #define SAI_xSR_FREQ ((uint32_t)0x00000008)
- #define SAI_xSR_CNRDY ((uint32_t)0x00000010)
- #define SAI_xSR_AFSDET ((uint32_t)0x00000020)
- #define SAI_xSR_LFSDET ((uint32_t)0x00000040)
- #define SAI_xSR_FLVL ((uint32_t)0x00070000)
- #define SAI_xSR_FLVL_0 ((uint32_t)0x00010000)
- #define SAI_xSR_FLVL_1 ((uint32_t)0x00020000)
- #define SAI_xSR_FLVL_2 ((uint32_t)0x00030000)
- #define SAI_xCLRFR_COVRUDR ((uint32_t)0x00000001)
- #define SAI_xCLRFR_CMUTEDET ((uint32_t)0x00000002)
- #define SAI_xCLRFR_CWCKCFG ((uint32_t)0x00000004)
- #define SAI_xCLRFR_CFREQ ((uint32_t)0x00000008)
- #define SAI_xCLRFR_CCNRDY ((uint32_t)0x00000010)
- #define SAI_xCLRFR_CAFSDET ((uint32_t)0x00000020)
- #define SAI_xCLRFR_CLFSDET ((uint32_t)0x00000040)
- #define SAI_xDR_DATA ((uint32_t)0xFFFFFFFF)
- #define SDIO_POWER_PWRCTRL ((uint8_t)0x03)
- #define SDIO_POWER_PWRCTRL_0 ((uint8_t)0x01)
- #define SDIO_POWER_PWRCTRL_1 ((uint8_t)0x02)
- #define SDIO_CLKCR_CLKDIV ((uint16_t)0x00FF)
- #define SDIO_CLKCR_CLKEN ((uint16_t)0x0100)
- #define SDIO_CLKCR_PWRSAV ((uint16_t)0x0200)
- #define SDIO_CLKCR_BYPASS ((uint16_t)0x0400)

- #define SDIO_CLKCR_WIDBUS ((uint16_t)0x1800)
- #define SDIO_CLKCR_WIDBUS_0 ((uint16_t)0x0800)
- #define SDIO_CLKCR_WIDBUS_1 ((uint16_t)0x1000)
- #define SDIO_CLKCR_NEGEDGE ((uint16_t)0x2000)
- #define SDIO_CLKCR_HWFC_EN ((uint16_t)0x4000)
- #define SDIO_ARG_CMDARG ((uint32_t)0xFFFFFFFF)
- #define SDIO_CMD_CMDINDEX ((uint16_t)0x003F)
- #define SDIO_CMD_WAITRESP ((uint16_t)0x00C0)
- #define SDIO_CMD_WAITRESP_0 ((uint16_t)0x0040)
- #define SDIO_CMD_WAITRESP_1 ((uint16_t)0x0080)
- #define SDIO_CMD_WAITINT ((uint16_t)0x0100)
- #define SDIO_CMD_WAITPEND ((uint16_t)0x0200)
- #define SDIO_CMD_CPSMEN ((uint16_t)0x0400)
- #define SDIO_CMD_SDIOSUSPEND ((uint16_t)0x0800)
- #define SDIO_CMD_ENCMDCOMPL ((uint16_t)0x1000)
- #define SDIO_CMD_NIEN ((uint16_t)0x2000)
- #define SDIO_CMD_CEATACMD ((uint16_t)0x4000)
- #define SDIO_RESPCMD_RESPCMD ((uint8_t)0x3F)
- #define SDIO_RESP0_CARDSTATUS0 ((uint32_t)0xFFFFFFFF)
- #define SDIO_RESP1_CARDSTATUS1 ((uint32_t)0xFFFFFFFF)
- #define SDIO_RESP2_CARDSTATUS2 ((uint32_t)0xFFFFFFFF)
- #define SDIO_RESP3_CARDSTATUS3 ((uint32_t)0xFFFFFFFF)
- #define SDIO_RESP4_CARDSTATUS4 ((uint32_t)0xFFFFFFFF)
- #define SDIO_DTIMER_DATETIME ((uint32_t)0xFFFFFFFF)
- #define SDIO_DLEN_DATALENGTH ((uint32_t)0x01FFFFFF)
- #define SDIO_DCTRL_DTEN ((uint16_t)0x0001)
- #define SDIO_DCTRL_DDIR ((uint16_t)0x0002)
- #define SDIO_DCTRL_DTMODE ((uint16_t)0x0004)
- #define SDIO_DCTRL_DMAEN ((uint16_t)0x0008)
- #define SDIO_DCTRL_DBLOCKSIZE ((uint16_t)0x00F0)
- #define SDIO_DCTRL_DBLOCKSIZE_0 ((uint16_t)0x0010)
- #define SDIO_DCTRL_DBLOCKSIZE_1 ((uint16_t)0x0020)
- #define SDIO_DCTRL_DBLOCKSIZE_2 ((uint16_t)0x0040)
- #define SDIO_DCTRL_DBLOCKSIZE_3 ((uint16_t)0x0080)
- #define SDIO_DCTRL_RWSTART ((uint16_t)0x0100)
- #define SDIO_DCTRL_RWSTOP ((uint16_t)0x0200)
- #define SDIO_DCTRL_RWMOD ((uint16_t)0x0400)
- #define SDIO_DCTRL_SDIOEN ((uint16_t)0x0800)
- #define SDIO_DCOUNT_DATACOUNT ((uint32_t)0x01FFFFFF)
- #define SDIO_STA_CCRCFAIL ((uint32_t)0x00000001)
- #define SDIO_STA_DCRCFAIL ((uint32_t)0x00000002)
- #define SDIO_STA_CTIMEOUT ((uint32_t)0x00000004)
- #define SDIO_STA_DTIMEOUT ((uint32_t)0x00000008)
- #define SDIO_STA_TXUNDERR ((uint32_t)0x00000010)
- #define SDIO_STA_RXOVERR ((uint32_t)0x00000020)
- #define SDIO_STA_CMDREND ((uint32_t)0x00000040)
- #define SDIO_STA_CMDSENT ((uint32_t)0x00000080)
- #define SDIO_STA_DATAEND ((uint32_t)0x00000100)
- #define SDIO_STA_STBITERR ((uint32_t)0x00000200)
- #define SDIO_STA_DBCKEND ((uint32_t)0x00000400)
- #define SDIO_STA_CMDACT ((uint32_t)0x00000800)
- #define SDIO_STA_TXACT ((uint32_t)0x00001000)
- #define SDIO_STA_RXACT ((uint32_t)0x00002000)
- #define SDIO_STA_TXFIFOHE ((uint32_t)0x00004000)
- #define SDIO_STA_RXFIFOHF ((uint32_t)0x00008000)

- #define SDIO_STA_TXFIFOF ((uint32_t)0x00010000)
- #define SDIO_STA_RXFIFOF ((uint32_t)0x00020000)
- #define SDIO_STA_TXFIFOE ((uint32_t)0x00040000)
- #define SDIO_STA_RXFIFOE ((uint32_t)0x00080000)
- #define SDIO_STA_TXDAVL ((uint32_t)0x00100000)
- #define SDIO_STA_RXDAVL ((uint32_t)0x00200000)
- #define SDIO_STA_SDIOIT ((uint32_t)0x00400000)
- #define SDIO_STA_CEATAEND ((uint32_t)0x00800000)
- #define SDIO_ICR_CCRCFAILC ((uint32_t)0x00000001)
- #define SDIO_ICR_DCRCFAILC ((uint32_t)0x00000002)
- #define SDIO_ICR_CTIMEOUTC ((uint32_t)0x00000004)
- #define SDIO_ICR_DTIMEOUTC ((uint32_t)0x00000008)
- #define SDIO_ICR_TXUNDERRC ((uint32_t)0x00000010)
- #define SDIO_ICR_RXOVERRC ((uint32_t)0x00000020)
- #define SDIO_ICR_CMDRENDC ((uint32_t)0x00000040)
- #define SDIO_ICR_CMDSENTC ((uint32_t)0x00000080)
- #define SDIO_ICR_DATAENDC ((uint32_t)0x00000100)
- #define SDIO_ICR_STBITERRC ((uint32_t)0x00000200)
- #define SDIO_ICR_DBCKENDC ((uint32_t)0x00000400)
- #define SDIO_ICR_SDIOITC ((uint32_t)0x00400000)
- #define SDIO_ICR_CEATAENDC ((uint32_t)0x00800000)
- #define SDIO_MASK_CCRCFAILIE ((uint32_t)0x00000001)
- #define SDIO_MASK_DCRCFAILIE ((uint32_t)0x00000002)
- #define SDIO_MASK_CTIMEOUTIE ((uint32_t)0x00000004)
- #define SDIO_MASK_DTIMEOUTIE ((uint32_t)0x00000008)
- #define SDIO_MASK_TXUNDERRIE ((uint32_t)0x00000010)
- #define SDIO_MASK_RXOVERRIE ((uint32_t)0x00000020)
- #define SDIO_MASK_CMDRENDE ((uint32_t)0x00000040)
- #define SDIO_MASK_CMDSENTIE ((uint32_t)0x00000080)
- #define SDIO_MASK_DATAENDIE ((uint32_t)0x00000100)
- #define SDIO_MASK_STBITERRIE ((uint32_t)0x00000200)
- #define SDIO_MASK_DBCKENDIE ((uint32_t)0x00000400)
- #define SDIO_MASK_CMDACTIE ((uint32_t)0x00000800)
- #define SDIO_MASK_TXACTIE ((uint32_t)0x00001000)
- #define SDIO_MASK_RXACTIE ((uint32_t)0x00002000)
- #define SDIO_MASK_TXFIFOHEIE ((uint32_t)0x00004000)
- #define SDIO_MASK_RXFIFOHFIE ((uint32_t)0x00008000)
- #define SDIO_MASK_TXFIFOFIE ((uint32_t)0x00010000)
- #define SDIO_MASK_RXFIFOFIE ((uint32_t)0x00020000)
- #define SDIO_MASK_TXFIFOEIE ((uint32_t)0x00040000)
- #define SDIO_MASK_RXFIFOEIE ((uint32_t)0x00080000)
- #define SDIO_MASK_TXDAVLIE ((uint32_t)0x00100000)
- #define SDIO_MASK_RXDAVLIE ((uint32_t)0x00200000)
- #define SDIO_MASK_SDIOITIE ((uint32_t)0x00400000)
- #define SDIO_MASK_CEATAENDIE ((uint32_t)0x00800000)
- #define SDIO_FIFO_CNT_FIFOCOUNT ((uint32_t)0x00FFFFFF)
- #define SDIO_FIFO_FIFODATA ((uint32_t)0xFFFFFFFF)
- #define SPI_CR1_CPHA ((uint16_t)0x0001)
- #define SPI_CR1_CPOL ((uint16_t)0x0002)
- #define SPI_CR1_MSTR ((uint16_t)0x0004)
- #define SPI_CR1_BR ((uint16_t)0x0038)
- #define SPI_CR1_BR_0 ((uint16_t)0x0008)
- #define SPI_CR1_BR_1 ((uint16_t)0x0010)
- #define SPI_CR1_BR_2 ((uint16_t)0x0020)
- #define SPI_CR1_SPE ((uint16_t)0x0040)

- #define SPI_CR1_LSBFIRST ((uint16_t)0x0080)
- #define SPI_CR1_SSI ((uint16_t)0x0100)
- #define SPI_CR1_SSM ((uint16_t)0x0200)
- #define SPI_CR1_RXONLY ((uint16_t)0x0400)
- #define SPI_CR1_DFF ((uint16_t)0x0800)
- #define SPI_CR1_CRCNEXT ((uint16_t)0x1000)
- #define SPI_CR1_CRCEN ((uint16_t)0x2000)
- #define SPI_CR1_BIDIOE ((uint16_t)0x4000)
- #define SPI_CR1_BIDIMODE ((uint16_t)0x8000)
- #define SPI_CR2_RXDMAEN ((uint8_t)0x01)
- #define SPI_CR2_TXDMAEN ((uint8_t)0x02)
- #define SPI_CR2_SSOE ((uint8_t)0x04)
- #define SPI_CR2_ERRIE ((uint8_t)0x20)
- #define SPI_CR2_RXNEIE ((uint8_t)0x40)
- #define SPI_CR2_TXEIE ((uint8_t)0x80)
- #define SPI_SR_RXNE ((uint8_t)0x01)
- #define SPI_SR_TXE ((uint8_t)0x02)
- #define SPI_SR_CHSIDE ((uint8_t)0x04)
- #define SPI_SR_UDR ((uint8_t)0x08)
- #define SPI_SR_CRCERR ((uint8_t)0x10)
- #define SPI_SR_MODF ((uint8_t)0x20)
- #define SPI_SR_OVR ((uint8_t)0x40)
- #define SPI_SR_BSY ((uint8_t)0x80)
- #define SPI_DR_DR ((uint16_t)0xFFFF)
- #define SPI_CRCPR_CRCPOLY ((uint16_t)0xFFFF)
- #define SPI_RXCRCR_RXCRC ((uint16_t)0xFFFF)
- #define SPI_TXCRCR_TXCRC ((uint16_t)0xFFFF)
- #define SPI_I2SCFGR_CHLEN ((uint16_t)0x0001)
- #define SPI_I2SCFGR_DATLEN ((uint16_t)0x0006)
- #define SPI_I2SCFGR_DATLEN_0 ((uint16_t)0x0002)
- #define SPI_I2SCFGR_DATLEN_1 ((uint16_t)0x0004)
- #define SPI_I2SCFGR_CKPOL ((uint16_t)0x0008)
- #define SPI_I2SCFGR_I2SSTD ((uint16_t)0x0030)
- #define SPI_I2SCFGR_I2SSTD_0 ((uint16_t)0x0010)
- #define SPI_I2SCFGR_I2SSTD_1 ((uint16_t)0x0020)
- #define SPI_I2SCFGR_PCMSYNC ((uint16_t)0x0080)
- #define SPI_I2SCFGR_I2SCFG ((uint16_t)0x0300)
- #define SPI_I2SCFGR_I2SCFG_0 ((uint16_t)0x0100)
- #define SPI_I2SCFGR_I2SCFG_1 ((uint16_t)0x0200)
- #define SPI_I2SCFGR_I2SE ((uint16_t)0x0400)
- #define SPI_I2SCFGR_I2SMOD ((uint16_t)0x0800)
- #define SPI_I2SPR_I2SDIV ((uint16_t)0x00FF)
- #define SPI_I2SPR_ODD ((uint16_t)0x0100)
- #define SPI_I2SPR_MCKOE ((uint16_t)0x0200)
- #define SYSCFG_MEMRMP_MEM_MODE ((uint32_t)0x00000007)
- #define SYSCFG_MEMRMP_MEM_MODE_0 ((uint32_t)0x00000001)
- #define SYSCFG_MEMRMP_MEM_MODE_1 ((uint32_t)0x00000002)
- #define SYSCFG_MEMRMP_MEM_MODE_2 ((uint32_t)0x00000004)
- #define SYSCFG_MEMRMP_FB_MODE ((uint32_t)0x00000100)
- #define SYSCFG_MEMRMP_SWP_FMC ((uint32_t)0x00000C00)
- #define SYSCFG_MEMRMP_SWP_FMC_0 ((uint32_t)0x00000400)
- #define SYSCFG_MEMRMP_SWP_FMC_1 ((uint32_t)0x00000800)
- #define SYSCFG_PMC_ADCxDC2 ((uint32_t)0x00070000)
- #define SYSCFG_PMC_ADC1DC2 ((uint32_t)0x00010000)
- #define SYSCFG_PMC_ADC2DC2 ((uint32_t)0x00020000)

- #define SYSCFG_PMC_ADC3DC2 ((uint32_t)0x00040000)
- #define SYSCFG_PMC_MII_RMII_SEL ((uint32_t)0x00800000)
- #define **SYSCFG_PMC_MII_RMII** SYSCFG_PMC_MII_RMII_SEL
- #define SYSCFG_EXTICR1_EXTI0 ((uint16_t)0x000F)
- #define SYSCFG_EXTICR1_EXTI1 ((uint16_t)0x00F0)
- #define SYSCFG_EXTICR1_EXTI2 ((uint16_t)0x0F00)
- #define SYSCFG_EXTICR1_EXTI3 ((uint16_t)0xF000)
- #define SYSCFG_EXTICR1_EXTI0_PA ((uint16_t)0x0000)

EXTI0 configuration

- #define SYSCFG_EXTICR1_EXTI0_PB ((uint16_t)0x0001)
- #define SYSCFG_EXTICR1_EXTI0_PC ((uint16_t)0x0002)
- #define SYSCFG_EXTICR1_EXTI0_PD ((uint16_t)0x0003)
- #define SYSCFG_EXTICR1_EXTI0_PE ((uint16_t)0x0004)
- #define SYSCFG_EXTICR1_EXTI0_PF ((uint16_t)0x0005)
- #define SYSCFG_EXTICR1_EXTI0_PG ((uint16_t)0x0006)
- #define SYSCFG_EXTICR1_EXTI0_PH ((uint16_t)0x0007)
- #define SYSCFG_EXTICR1_EXTI0_PI ((uint16_t)0x0008)
- #define SYSCFG_EXTICR1_EXTI0_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR1_EXTI0_PK ((uint16_t)0x000A)
- #define SYSCFG_EXTICR1_EXTI1_PA ((uint16_t)0x0000)

EXTI1 configuration

- #define SYSCFG_EXTICR1_EXTI1_PB ((uint16_t)0x0010)
- #define SYSCFG_EXTICR1_EXTI1_PC ((uint16_t)0x0020)
- #define SYSCFG_EXTICR1_EXTI1_PD ((uint16_t)0x0030)
- #define SYSCFG_EXTICR1_EXTI1_PE ((uint16_t)0x0040)
- #define SYSCFG_EXTICR1_EXTI1_PF ((uint16_t)0x0050)
- #define SYSCFG_EXTICR1_EXTI1_PG ((uint16_t)0x0060)
- #define SYSCFG_EXTICR1_EXTI1_PH ((uint16_t)0x0070)
- #define SYSCFG_EXTICR1_EXTI1_PI ((uint16_t)0x0080)
- #define SYSCFG_EXTICR1_EXTI1_PJ ((uint16_t)0x0090)
- #define SYSCFG_EXTICR1_EXTI1_PK ((uint16_t)0x00A0)
- #define SYSCFG_EXTICR1_EXTI2_PA ((uint16_t)0x0000)

EXTI2 configuration

- #define SYSCFG_EXTICR1_EXTI2_PB ((uint16_t)0x0100)
- #define SYSCFG_EXTICR1_EXTI2_PC ((uint16_t)0x0200)
- #define SYSCFG_EXTICR1_EXTI2_PD ((uint16_t)0x0300)
- #define SYSCFG_EXTICR1_EXTI2_PE ((uint16_t)0x0400)
- #define SYSCFG_EXTICR1_EXTI2_PF ((uint16_t)0x0500)
- #define SYSCFG_EXTICR1_EXTI2_PG ((uint16_t)0x0600)
- #define SYSCFG_EXTICR1_EXTI2_PH ((uint16_t)0x0700)
- #define SYSCFG_EXTICR1_EXTI2_PI ((uint16_t)0x0800)
- #define SYSCFG_EXTICR1_EXTI2_PJ ((uint16_t)0x0900)
- #define SYSCFG_EXTICR1_EXTI2_PK ((uint16_t)0x0A00)
- #define SYSCFG_EXTICR1_EXTI3_PA ((uint16_t)0x0000)

EXTI3 configuration

- #define SYSCFG_EXTICR1_EXTI3_PB ((uint16_t)0x1000)
- #define SYSCFG_EXTICR1_EXTI3_PC ((uint16_t)0x2000)
- #define SYSCFG_EXTICR1_EXTI3_PD ((uint16_t)0x3000)
- #define SYSCFG_EXTICR1_EXTI3_PE ((uint16_t)0x4000)
- #define SYSCFG_EXTICR1_EXTI3_PF ((uint16_t)0x5000)

- #define SYSCFG_EXTICR1_EXTI3_PG ((uint16_t)0x6000)
- #define SYSCFG_EXTICR1_EXTI3_PH ((uint16_t)0x7000)
- #define SYSCFG_EXTICR1_EXTI3_PI ((uint16_t)0x8000)
- #define SYSCFG_EXTICR1_EXTI3_PJ ((uint16_t)0x9000)
- #define SYSCFG_EXTICR1_EXTI3_PK ((uint16_t)0xA000)
- #define SYSCFG_EXTICR2_EXTI4 ((uint16_t)0x000F)
- #define SYSCFG_EXTICR2_EXTI5 ((uint16_t)0x00F0)
- #define SYSCFG_EXTICR2_EXTI6 ((uint16_t)0x0F00)
- #define SYSCFG_EXTICR2_EXTI7 ((uint16_t)0xF000)
- #define SYSCFG_EXTICR2_EXTI4_PA ((uint16_t)0x0000)

EXTI4 configuration

- #define SYSCFG_EXTICR2_EXTI4_PB ((uint16_t)0x0001)
- #define SYSCFG_EXTICR2_EXTI4_PC ((uint16_t)0x0002)
- #define SYSCFG_EXTICR2_EXTI4_PD ((uint16_t)0x0003)
- #define SYSCFG_EXTICR2_EXTI4_PE ((uint16_t)0x0004)
- #define SYSCFG_EXTICR2_EXTI4_PF ((uint16_t)0x0005)
- #define SYSCFG_EXTICR2_EXTI4_PG ((uint16_t)0x0006)
- #define SYSCFG_EXTICR2_EXTI4_PH ((uint16_t)0x0007)
- #define SYSCFG_EXTICR2_EXTI4_PI ((uint16_t)0x0008)
- #define SYSCFG_EXTICR2_EXTI4_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR2_EXTI4_PK ((uint16_t)0x000A)
- #define SYSCFG_EXTICR2_EXTI5_PA ((uint16_t)0x0000)

EXTI5 configuration

- #define SYSCFG_EXTICR2_EXTI5_PB ((uint16_t)0x0010)
- #define SYSCFG_EXTICR2_EXTI5_PC ((uint16_t)0x0020)
- #define SYSCFG_EXTICR2_EXTI5_PD ((uint16_t)0x0030)
- #define SYSCFG_EXTICR2_EXTI5_PE ((uint16_t)0x0040)
- #define SYSCFG_EXTICR2_EXTI5_PF ((uint16_t)0x0050)
- #define SYSCFG_EXTICR2_EXTI5_PG ((uint16_t)0x0060)
- #define SYSCFG_EXTICR2_EXTI5_PH ((uint16_t)0x0070)
- #define SYSCFG_EXTICR2_EXTI5_PI ((uint16_t)0x0080)
- #define SYSCFG_EXTICR2_EXTI5_PJ ((uint16_t)0x0090)
- #define SYSCFG_EXTICR2_EXTI5_PK ((uint16_t)0x00A0)
- #define SYSCFG_EXTICR2_EXTI6_PA ((uint16_t)0x0000)

EXTI6 configuration

- #define SYSCFG_EXTICR2_EXTI6_PB ((uint16_t)0x0100)
- #define SYSCFG_EXTICR2_EXTI6_PC ((uint16_t)0x0200)
- #define SYSCFG_EXTICR2_EXTI6_PD ((uint16_t)0x0300)
- #define SYSCFG_EXTICR2_EXTI6_PE ((uint16_t)0x0400)
- #define SYSCFG_EXTICR2_EXTI6_PF ((uint16_t)0x0500)
- #define SYSCFG_EXTICR2_EXTI6_PG ((uint16_t)0x0600)
- #define SYSCFG_EXTICR2_EXTI6_PH ((uint16_t)0x0700)
- #define SYSCFG_EXTICR2_EXTI6_PI ((uint16_t)0x0800)
- #define SYSCFG_EXTICR2_EXTI6_PJ ((uint16_t)0x0900)
- #define SYSCFG_EXTICR2_EXTI6_PK ((uint16_t)0x0A00)
- #define SYSCFG_EXTICR2_EXTI7_PA ((uint16_t)0x0000)

EXTI7 configuration

- #define SYSCFG_EXTICR2_EXTI7_PB ((uint16_t)0x1000)
- #define SYSCFG_EXTICR2_EXTI7_PC ((uint16_t)0x2000)
- #define SYSCFG_EXTICR2_EXTI7_PD ((uint16_t)0x3000)

- #define SYSCFG_EXTICR2_EXTI7_PE ((uint16_t)0x4000)
- #define SYSCFG_EXTICR2_EXTI7_PF ((uint16_t)0x5000)
- #define SYSCFG_EXTICR2_EXTI7_PG ((uint16_t)0x6000)
- #define SYSCFG_EXTICR2_EXTI7_PH ((uint16_t)0x7000)
- #define SYSCFG_EXTICR2_EXTI7_PI ((uint16_t)0x8000)
- #define SYSCFG_EXTICR2_EXTI7_PJ ((uint16_t)0x9000)
- #define SYSCFG_EXTICR2_EXTI7_PK ((uint16_t)0xA000)
- #define SYSCFG_EXTICR3_EXTI8 ((uint16_t)0x000F)
- #define SYSCFG_EXTICR3_EXTI9 ((uint16_t)0x00F0)
- #define SYSCFG_EXTICR3_EXTI10 ((uint16_t)0x0F00)
- #define SYSCFG_EXTICR3_EXTI11 ((uint16_t)0xF000)
- #define SYSCFG_EXTICR3_EXTI8_PA ((uint16_t)0x0000)

EXTI8 configuration

- #define SYSCFG_EXTICR3_EXTI8_PB ((uint16_t)0x0001)
- #define SYSCFG_EXTICR3_EXTI8_PC ((uint16_t)0x0002)
- #define SYSCFG_EXTICR3_EXTI8_PD ((uint16_t)0x0003)
- #define SYSCFG_EXTICR3_EXTI8_PE ((uint16_t)0x0004)
- #define SYSCFG_EXTICR3_EXTI8_PF ((uint16_t)0x0005)
- #define SYSCFG_EXTICR3_EXTI8_PG ((uint16_t)0x0006)
- #define SYSCFG_EXTICR3_EXTI8_PH ((uint16_t)0x0007)
- #define SYSCFG_EXTICR3_EXTI8_PI ((uint16_t)0x0008)
- #define SYSCFG_EXTICR3_EXTI8_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR3_EXTI9_PA ((uint16_t)0x0000)

EXTI9 configuration

- #define SYSCFG_EXTICR3_EXTI9_PB ((uint16_t)0x0010)
- #define SYSCFG_EXTICR3_EXTI9_PC ((uint16_t)0x0020)
- #define SYSCFG_EXTICR3_EXTI9_PD ((uint16_t)0x0030)
- #define SYSCFG_EXTICR3_EXTI9_PE ((uint16_t)0x0040)
- #define SYSCFG_EXTICR3_EXTI9_PF ((uint16_t)0x0050)
- #define SYSCFG_EXTICR3_EXTI9_PG ((uint16_t)0x0060)
- #define SYSCFG_EXTICR3_EXTI9_PH ((uint16_t)0x0070)
- #define SYSCFG_EXTICR3_EXTI9_PI ((uint16_t)0x0080)
- #define SYSCFG_EXTICR3_EXTI9_PJ ((uint16_t)0x0090)
- #define SYSCFG_EXTICR3_EXTI10_PA ((uint16_t)0x0000)

EXTI10 configuration

- #define SYSCFG_EXTICR3_EXTI10_PB ((uint16_t)0x0100)
- #define SYSCFG_EXTICR3_EXTI10_PC ((uint16_t)0x0200)
- #define SYSCFG_EXTICR3_EXTI10_PD ((uint16_t)0x0300)
- #define SYSCFG_EXTICR3_EXTI10_PE ((uint16_t)0x0400)
- #define SYSCFG_EXTICR3_EXTI10_PF ((uint16_t)0x0500)
- #define SYSCFG_EXTICR3_EXTI10_PG ((uint16_t)0x0600)
- #define SYSCFG_EXTICR3_EXTI10_PH ((uint16_t)0x0700)
- #define SYSCFG_EXTICR3_EXTI10_PI ((uint16_t)0x0800)
- #define SYSCFG_EXTICR3_EXTI10_PJ ((uint16_t)0x0900)
- #define SYSCFG_EXTICR3_EXTI11_PA ((uint16_t)0x0000)

EXTI11 configuration

- #define SYSCFG_EXTICR3_EXTI11_PB ((uint16_t)0x1000)
- #define SYSCFG_EXTICR3_EXTI11_PC ((uint16_t)0x2000)
- #define SYSCFG_EXTICR3_EXTI11_PD ((uint16_t)0x3000)
- #define SYSCFG_EXTICR3_EXTI11_PE ((uint16_t)0x4000)

- #define SYSCFG_EXTICR3_EXTI11_PF ((uint16_t)0x5000)
- #define SYSCFG_EXTICR3_EXTI11_PG ((uint16_t)0x6000)
- #define SYSCFG_EXTICR3_EXTI11_PH ((uint16_t)0x7000)
- #define SYSCFG_EXTICR3_EXTI11_PI ((uint16_t)0x8000)
- #define SYSCFG_EXTICR3_EXTI11_PJ ((uint16_t)0x9000)
- #define SYSCFG_EXTICR4_EXTI12 ((uint16_t)0x000F)
- #define SYSCFG_EXTICR4_EXTI13 ((uint16_t)0x00F0)
- #define SYSCFG_EXTICR4_EXTI14 ((uint16_t)0x0F00)
- #define SYSCFG_EXTICR4_EXTI15 ((uint16_t)0xF000)
- #define SYSCFG_EXTICR4_EXTI12_PA ((uint16_t)0x0000)

EXTI12 configuration

- #define SYSCFG_EXTICR4_EXTI12_PB ((uint16_t)0x0001)
- #define SYSCFG_EXTICR4_EXTI12_PC ((uint16_t)0x0002)
- #define SYSCFG_EXTICR4_EXTI12_PD ((uint16_t)0x0003)
- #define SYSCFG_EXTICR4_EXTI12_PE ((uint16_t)0x0004)
- #define SYSCFG_EXTICR4_EXTI12_PF ((uint16_t)0x0005)
- #define SYSCFG_EXTICR4_EXTI12_PG ((uint16_t)0x0006)
- #define SYSCFG_EXTICR4_EXTI12_PH ((uint16_t)0x0007)
- #define SYSCFG_EXTICR4_EXTI12_PI ((uint16_t)0x0008)
- #define SYSCFG_EXTICR4_EXTI12_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR4_EXTI13_PA ((uint16_t)0x0000)

EXTI13 configuration

- #define SYSCFG_EXTICR4_EXTI13_PB ((uint16_t)0x0010)
- #define SYSCFG_EXTICR4_EXTI13_PC ((uint16_t)0x0020)
- #define SYSCFG_EXTICR4_EXTI13_PD ((uint16_t)0x0030)
- #define SYSCFG_EXTICR4_EXTI13_PE ((uint16_t)0x0040)
- #define SYSCFG_EXTICR4_EXTI13_PF ((uint16_t)0x0050)
- #define SYSCFG_EXTICR4_EXTI13_PG ((uint16_t)0x0060)
- #define SYSCFG_EXTICR4_EXTI13_PH ((uint16_t)0x0070)
- #define SYSCFG_EXTICR4_EXTI13_PI ((uint16_t)0x0008)
- #define SYSCFG_EXTICR4_EXTI13_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR4_EXTI14_PA ((uint16_t)0x0000)

EXTI14 configuration

- #define SYSCFG_EXTICR4_EXTI14_PB ((uint16_t)0x0100)
- #define SYSCFG_EXTICR4_EXTI14_PC ((uint16_t)0x0200)
- #define SYSCFG_EXTICR4_EXTI14_PD ((uint16_t)0x0300)
- #define SYSCFG_EXTICR4_EXTI14_PE ((uint16_t)0x0400)
- #define SYSCFG_EXTICR4_EXTI14_PF ((uint16_t)0x0500)
- #define SYSCFG_EXTICR4_EXTI14_PG ((uint16_t)0x0600)
- #define SYSCFG_EXTICR4_EXTI14_PH ((uint16_t)0x0700)
- #define SYSCFG_EXTICR4_EXTI14_PI ((uint16_t)0x0800)
- #define SYSCFG_EXTICR4_EXTI14_PJ ((uint16_t)0x0900)
- #define SYSCFG_EXTICR4_EXTI15_PA ((uint16_t)0x0000)

EXTI15 configuration

- #define SYSCFG_EXTICR4_EXTI15_PB ((uint16_t)0x1000)
- #define SYSCFG_EXTICR4_EXTI15_PC ((uint16_t)0x2000)
- #define SYSCFG_EXTICR4_EXTI15_PD ((uint16_t)0x3000)
- #define SYSCFG_EXTICR4_EXTI15_PE ((uint16_t)0x4000)
- #define SYSCFG_EXTICR4_EXTI15_PF ((uint16_t)0x5000)
- #define SYSCFG_EXTICR4_EXTI15_PG ((uint16_t)0x6000)

- #define SYSCFG_EXTICR4_EXTI15_PH ((uint16_t)0x7000)
- #define SYSCFG_EXTICR4_EXTI15_PI ((uint16_t)0x8000)
- #define SYSCFG_EXTICR4_EXTI15_PJ ((uint16_t)0x9000)
- #define SYSCFG_CMPCR_CMP_PD ((uint32_t)0x00000001)
- #define SYSCFG_CMPCR_READY ((uint32_t)0x00000100)
- #define TIM_CR1_CEN ((uint16_t)0x0001)
- #define TIM_CR1_UDIS ((uint16_t)0x0002)
- #define TIM_CR1_URS ((uint16_t)0x0004)
- #define TIM_CR1_OPM ((uint16_t)0x0008)
- #define TIM_CR1_DIR ((uint16_t)0x0010)
- #define TIM_CR1_CMS ((uint16_t)0x0060)
- #define TIM_CR1_CMS_0 ((uint16_t)0x0020)
- #define TIM_CR1_CMS_1 ((uint16_t)0x0040)
- #define TIM_CR1_ARPE ((uint16_t)0x0080)
- #define TIM_CR1_CKD ((uint16_t)0x0300)
- #define TIM_CR1_CKD_0 ((uint16_t)0x0100)
- #define TIM_CR1_CKD_1 ((uint16_t)0x0200)
- #define TIM_CR2_CCPC ((uint16_t)0x0001)
- #define TIM_CR2_CCUS ((uint16_t)0x0004)
- #define TIM_CR2_CCDS ((uint16_t)0x0008)
- #define TIM_CR2_MMS ((uint16_t)0x0070)
- #define TIM_CR2_MMS_0 ((uint16_t)0x0010)
- #define TIM_CR2_MMS_1 ((uint16_t)0x0020)
- #define TIM_CR2_MMS_2 ((uint16_t)0x0040)
- #define TIM_CR2_TI1S ((uint16_t)0x0080)
- #define TIM_CR2_OIS1 ((uint16_t)0x0100)
- #define TIM_CR2_OIS1N ((uint16_t)0x0200)
- #define TIM_CR2_OIS2 ((uint16_t)0x0400)
- #define TIM_CR2_OIS2N ((uint16_t)0x0800)
- #define TIM_CR2_OIS3 ((uint16_t)0x1000)
- #define TIM_CR2_OIS3N ((uint16_t)0x2000)
- #define TIM_CR2_OIS4 ((uint16_t)0x4000)
- #define TIM_SMCR_SMS ((uint16_t)0x0007)
- #define TIM_SMCR_SMS_0 ((uint16_t)0x0001)
- #define TIM_SMCR_SMS_1 ((uint16_t)0x0002)
- #define TIM_SMCR_SMS_2 ((uint16_t)0x0004)
- #define TIM_SMCR_TS ((uint16_t)0x0070)
- #define TIM_SMCR_TS_0 ((uint16_t)0x0010)
- #define TIM_SMCR_TS_1 ((uint16_t)0x0020)
- #define TIM_SMCR_TS_2 ((uint16_t)0x0040)
- #define TIM_SMCR_MSM ((uint16_t)0x0080)
- #define TIM_SMCR_ETF ((uint16_t)0x0F00)
- #define TIM_SMCR_ETF_0 ((uint16_t)0x0100)
- #define TIM_SMCR_ETF_1 ((uint16_t)0x0200)
- #define TIM_SMCR_ETF_2 ((uint16_t)0x0400)
- #define TIM_SMCR_ETF_3 ((uint16_t)0x0800)
- #define TIM_SMCR_ETPS ((uint16_t)0x3000)
- #define TIM_SMCR_ETPS_0 ((uint16_t)0x1000)
- #define TIM_SMCR_ETPS_1 ((uint16_t)0x2000)
- #define TIM_SMCR_ECE ((uint16_t)0x4000)
- #define TIM_SMCR_ETP ((uint16_t)0x8000)
- #define TIM_DIER_UIE ((uint16_t)0x0001)
- #define TIM_DIER_CC1IE ((uint16_t)0x0002)
- #define TIM_DIER_CC2IE ((uint16_t)0x0004)
- #define TIM_DIER_CC3IE ((uint16_t)0x0008)

- #define **TIM_DIER_CC4IE** ((uint16_t)0x0010)
- #define **TIM_DIER_COMIE** ((uint16_t)0x0020)
- #define **TIM_DIER_TIE** ((uint16_t)0x0040)
- #define **TIM_DIER_BIE** ((uint16_t)0x0080)
- #define **TIM_DIER_UDE** ((uint16_t)0x0100)
- #define **TIM_DIER_CC1DE** ((uint16_t)0x0200)
- #define **TIM_DIER_CC2DE** ((uint16_t)0x0400)
- #define **TIM_DIER_CC3DE** ((uint16_t)0x0800)
- #define **TIM_DIER_CC4DE** ((uint16_t)0x1000)
- #define **TIM_DIER_COMDE** ((uint16_t)0x2000)
- #define **TIM_DIER_TDE** ((uint16_t)0x4000)
- #define **TIM_SR UIF** ((uint16_t)0x0001)
- #define **TIM_SR_CC1IF** ((uint16_t)0x0002)
- #define **TIM_SR_CC2IF** ((uint16_t)0x0004)
- #define **TIM_SR_CC3IF** ((uint16_t)0x0008)
- #define **TIM_SR_CC4IF** ((uint16_t)0x0010)
- #define **TIM_SR_COMIF** ((uint16_t)0x0020)
- #define **TIM_SR_TIF** ((uint16_t)0x0040)
- #define **TIM_SR_BIF** ((uint16_t)0x0080)
- #define **TIM_SR_CC1OF** ((uint16_t)0x0200)
- #define **TIM_SR_CC2OF** ((uint16_t)0x0400)
- #define **TIM_SR_CC3OF** ((uint16_t)0x0800)
- #define **TIM_SR_CC4OF** ((uint16_t)0x1000)
- #define **TIM_EGR_UG** ((uint8_t)0x01)
- #define **TIM_EGR_CC1G** ((uint8_t)0x02)
- #define **TIM_EGR_CC2G** ((uint8_t)0x04)
- #define **TIM_EGR_CC3G** ((uint8_t)0x08)
- #define **TIM_EGR_CC4G** ((uint8_t)0x10)
- #define **TIM_EGR_COMG** ((uint8_t)0x20)
- #define **TIM_EGR_TG** ((uint8_t)0x40)
- #define **TIM_EGR_BG** ((uint8_t)0x80)
- #define **TIM_CCMR1_CC1S** ((uint16_t)0x0003)
- #define **TIM_CCMR1_CC1S_0** ((uint16_t)0x0001)
- #define **TIM_CCMR1_CC1S_1** ((uint16_t)0x0002)
- #define **TIM_CCMR1_OC1FE** ((uint16_t)0x0004)
- #define **TIM_CCMR1_OC1PE** ((uint16_t)0x0008)
- #define **TIM_CCMR1_OC1M** ((uint16_t)0x0070)
- #define **TIM_CCMR1_OC1M_0** ((uint16_t)0x0010)
- #define **TIM_CCMR1_OC1M_1** ((uint16_t)0x0020)
- #define **TIM_CCMR1_OC1M_2** ((uint16_t)0x0040)
- #define **TIM_CCMR1_OC1CE** ((uint16_t)0x0080)
- #define **TIM_CCMR1_CC2S** ((uint16_t)0x0300)
- #define **TIM_CCMR1_CC2S_0** ((uint16_t)0x0100)
- #define **TIM_CCMR1_CC2S_1** ((uint16_t)0x0200)
- #define **TIM_CCMR1_OC2FE** ((uint16_t)0x0400)
- #define **TIM_CCMR1_OC2PE** ((uint16_t)0x0800)
- #define **TIM_CCMR1_OC2M** ((uint16_t)0x7000)
- #define **TIM_CCMR1_OC2M_0** ((uint16_t)0x1000)
- #define **TIM_CCMR1_OC2M_1** ((uint16_t)0x2000)
- #define **TIM_CCMR1_OC2M_2** ((uint16_t)0x4000)
- #define **TIM_CCMR1_OC2CE** ((uint16_t)0x8000)
- #define **TIM_CCMR1_IC1PSC** ((uint16_t)0x000C)
- #define **TIM_CCMR1_IC1PSC_0** ((uint16_t)0x0004)
- #define **TIM_CCMR1_IC1PSC_1** ((uint16_t)0x0008)
- #define **TIM_CCMR1_IC1F** ((uint16_t)0x00F0)

- #define TIM_CCMR1_IC1F_0 ((uint16_t)0x0010)
- #define TIM_CCMR1_IC1F_1 ((uint16_t)0x0020)
- #define TIM_CCMR1_IC1F_2 ((uint16_t)0x0040)
- #define TIM_CCMR1_IC1F_3 ((uint16_t)0x0080)
- #define TIM_CCMR1_IC2PSC ((uint16_t)0xC000)
- #define TIM_CCMR1_IC2PSC_0 ((uint16_t)0x0400)
- #define TIM_CCMR1_IC2PSC_1 ((uint16_t)0x0800)
- #define TIM_CCMR1_IC2F ((uint16_t)0xF000)
- #define TIM_CCMR1_IC2F_0 ((uint16_t)0x1000)
- #define TIM_CCMR1_IC2F_1 ((uint16_t)0x2000)
- #define TIM_CCMR1_IC2F_2 ((uint16_t)0x4000)
- #define TIM_CCMR1_IC2F_3 ((uint16_t)0x8000)
- #define TIM_CCMR2_CC3S ((uint16_t)0x0003)
- #define TIM_CCMR2_CC3S_0 ((uint16_t)0x0001)
- #define TIM_CCMR2_CC3S_1 ((uint16_t)0x0002)
- #define TIM_CCMR2_OC3FE ((uint16_t)0x0004)
- #define TIM_CCMR2_OC3PE ((uint16_t)0x0008)
- #define TIM_CCMR2_OC3M ((uint16_t)0x0070)
- #define TIM_CCMR2_OC3M_0 ((uint16_t)0x0010)
- #define TIM_CCMR2_OC3M_1 ((uint16_t)0x0020)
- #define TIM_CCMR2_OC3M_2 ((uint16_t)0x0040)
- #define TIM_CCMR2_OC3CE ((uint16_t)0x0080)
- #define TIM_CCMR2_CC4S ((uint16_t)0x0300)
- #define TIM_CCMR2_CC4S_0 ((uint16_t)0x0100)
- #define TIM_CCMR2_CC4S_1 ((uint16_t)0x0200)
- #define TIM_CCMR2_OC4FE ((uint16_t)0x0400)
- #define TIM_CCMR2_OC4PE ((uint16_t)0x0800)
- #define TIM_CCMR2_OC4M ((uint16_t)0x7000)
- #define TIM_CCMR2_OC4M_0 ((uint16_t)0x1000)
- #define TIM_CCMR2_OC4M_1 ((uint16_t)0x2000)
- #define TIM_CCMR2_OC4M_2 ((uint16_t)0x4000)
- #define TIM_CCMR2_OC4CE ((uint16_t)0x8000)
- #define TIM_CCMR2_IC3PSC ((uint16_t)0x000C)
- #define TIM_CCMR2_IC3PSC_0 ((uint16_t)0x0004)
- #define TIM_CCMR2_IC3PSC_1 ((uint16_t)0x0008)
- #define TIM_CCMR2_IC3F ((uint16_t)0x00F0)
- #define TIM_CCMR2_IC3F_0 ((uint16_t)0x0010)
- #define TIM_CCMR2_IC3F_1 ((uint16_t)0x0020)
- #define TIM_CCMR2_IC3F_2 ((uint16_t)0x0040)
- #define TIM_CCMR2_IC3F_3 ((uint16_t)0x0080)
- #define TIM_CCMR2_IC4PSC ((uint16_t)0xC000)
- #define TIM_CCMR2_IC4PSC_0 ((uint16_t)0x0400)
- #define TIM_CCMR2_IC4PSC_1 ((uint16_t)0x0800)
- #define TIM_CCMR2_IC4F ((uint16_t)0xF000)
- #define TIM_CCMR2_IC4F_0 ((uint16_t)0x1000)
- #define TIM_CCMR2_IC4F_1 ((uint16_t)0x2000)
- #define TIM_CCMR2_IC4F_2 ((uint16_t)0x4000)
- #define TIM_CCMR2_IC4F_3 ((uint16_t)0x8000)
- #define TIM_CCER_CC1E ((uint16_t)0x0001)
- #define TIM_CCER_CC1P ((uint16_t)0x0002)
- #define TIM_CCER_CC1NE ((uint16_t)0x0004)
- #define TIM_CCER_CC1NP ((uint16_t)0x0008)
- #define TIM_CCER_CC2E ((uint16_t)0x0010)
- #define TIM_CCER_CC2P ((uint16_t)0x0020)
- #define TIM_CCER_CC2NE ((uint16_t)0x0040)

- #define TIM_CCER_CC2NP ((uint16_t)0x0080)
- #define TIM_CCER_CC3E ((uint16_t)0x0100)
- #define TIM_CCER_CC3P ((uint16_t)0x0200)
- #define TIM_CCER_CC3NE ((uint16_t)0x0400)
- #define TIM_CCER_CC3NP ((uint16_t)0x0800)
- #define TIM_CCER_CC4E ((uint16_t)0x1000)
- #define TIM_CCER_CC4P ((uint16_t)0x2000)
- #define TIM_CCER_CC4NP ((uint16_t)0x8000)
- #define TIM_CNT_CNT ((uint16_t)0xFFFF)
- #define TIM_PSC_PSC ((uint16_t)0xFFFF)
- #define TIM_ARR_ARR ((uint16_t)0xFFFF)
- #define TIM_RCR REP ((uint8_t)0xFF)
- #define TIM_CCR1_CCR1 ((uint16_t)0xFFFF)
- #define TIM_CCR2_CCR2 ((uint16_t)0xFFFF)
- #define TIM_CCR3_CCR3 ((uint16_t)0xFFFF)
- #define TIM_CCR4_CCR4 ((uint16_t)0xFFFF)
- #define TIM_BDTR_DTG ((uint16_t)0x00FF)
- #define TIM_BDTR_DTG_0 ((uint16_t)0x0001)
- #define TIM_BDTR_DTG_1 ((uint16_t)0x0002)
- #define TIM_BDTR_DTG_2 ((uint16_t)0x0004)
- #define TIM_BDTR_DTG_3 ((uint16_t)0x0008)
- #define TIM_BDTR_DTG_4 ((uint16_t)0x0010)
- #define TIM_BDTR_DTG_5 ((uint16_t)0x0020)
- #define TIM_BDTR_DTG_6 ((uint16_t)0x0040)
- #define TIM_BDTR_DTG_7 ((uint16_t)0x0080)
- #define TIM_BDTR_LOCK ((uint16_t)0x0300)
- #define TIM_BDTR_LOCK_0 ((uint16_t)0x0100)
- #define TIM_BDTR_LOCK_1 ((uint16_t)0x0200)
- #define TIM_BDTR_OSSI ((uint16_t)0x0400)
- #define TIM_BDTR_OSSR ((uint16_t)0x0800)
- #define TIM_BDTR_BKE ((uint16_t)0x1000)
- #define TIM_BDTR_BKP ((uint16_t)0x2000)
- #define TIM_BDTR_AOE ((uint16_t)0x4000)
- #define TIM_BDTR_MOE ((uint16_t)0x8000)
- #define TIM_DCR_DBA ((uint16_t)0x001F)
- #define TIM_DCR_DBA_0 ((uint16_t)0x0001)
- #define TIM_DCR_DBA_1 ((uint16_t)0x0002)
- #define TIM_DCR_DBA_2 ((uint16_t)0x0004)
- #define TIM_DCR_DBA_3 ((uint16_t)0x0008)
- #define TIM_DCR_DBA_4 ((uint16_t)0x0010)
- #define TIM_DCR_DBL ((uint16_t)0x1F00)
- #define TIM_DCR_DBL_0 ((uint16_t)0x0100)
- #define TIM_DCR_DBL_1 ((uint16_t)0x0200)
- #define TIM_DCR_DBL_2 ((uint16_t)0x0400)
- #define TIM_DCR_DBL_3 ((uint16_t)0x0800)
- #define TIM_DCR_DBL_4 ((uint16_t)0x1000)
- #define TIM_DMAR_DMAB ((uint16_t)0xFFFF)
- #define TIM_OR_TI4_RMP ((uint16_t)0x00C0)
- #define TIM_OR_TI4_RMP_0 ((uint16_t)0x0040)
- #define TIM_OR_TI4_RMP_1 ((uint16_t)0x0080)
- #define TIM_OR_ITR1_RMP ((uint16_t)0x0C00)
- #define TIM_OR_ITR1_RMP_0 ((uint16_t)0x0400)
- #define TIM_OR_ITR1_RMP_1 ((uint16_t)0x0800)
- #define USART_SR_PE ((uint16_t)0x0001)
- #define USART_SR_FE ((uint16_t)0x0002)

- #define USART_SR_NE ((uint16_t)0x0004)
- #define USART_SR_ORE ((uint16_t)0x0008)
- #define USART_SR_IDLE ((uint16_t)0x0010)
- #define USART_SR_RXNE ((uint16_t)0x0020)
- #define USART_SR_TC ((uint16_t)0x0040)
- #define USART_SR_TXE ((uint16_t)0x0080)
- #define USART_SR_LBD ((uint16_t)0x0100)
- #define USART_SR_CTS ((uint16_t)0x0200)
- #define USART_DR_DR ((uint16_t)0x01FF)
- #define USART_BRR_DIV_Fraction ((uint16_t)0x000F)
- #define USART_BRR_DIV_Mantissa ((uint16_t)0xFFFF0)
- #define USART_CR1_SBK ((uint16_t)0x0001)
- #define USART_CR1_RWU ((uint16_t)0x0002)
- #define USART_CR1_RE ((uint16_t)0x0004)
- #define USART_CR1_TE ((uint16_t)0x0008)
- #define USART_CR1_IDLEIE ((uint16_t)0x0010)
- #define USART_CR1_RXNEIE ((uint16_t)0x0020)
- #define USART_CR1_TCIE ((uint16_t)0x0040)
- #define USART_CR1_TXEIE ((uint16_t)0x0080)
- #define USART_CR1_PEIE ((uint16_t)0x0100)
- #define USART_CR1_PS ((uint16_t)0x0200)
- #define USART_CR1_PCE ((uint16_t)0x0400)
- #define USART_CR1_WAKE ((uint16_t)0x0800)
- #define USART_CR1_M ((uint16_t)0x1000)
- #define USART_CR1_UE ((uint16_t)0x2000)
- #define USART_CR1_OVER8 ((uint16_t)0x8000)
- #define USART_CR2_ADD ((uint16_t)0x000F)
- #define USART_CR2_LBDL ((uint16_t)0x0020)
- #define USART_CR2_LBDIE ((uint16_t)0x0040)
- #define USART_CR2_LBCL ((uint16_t)0x0100)
- #define USART_CR2_CPHA ((uint16_t)0x0200)
- #define USART_CR2_CPOL ((uint16_t)0x0400)
- #define USART_CR2_CLKEN ((uint16_t)0x0800)
- #define USART_CR2_STOP ((uint16_t)0x3000)
- #define USART_CR2_STOP_0 ((uint16_t)0x1000)
- #define USART_CR2_STOP_1 ((uint16_t)0x2000)
- #define USART_CR2 LINEN ((uint16_t)0x4000)
- #define USART_CR3_EIE ((uint16_t)0x0001)
- #define USART_CR3_IREN ((uint16_t)0x0002)
- #define USART_CR3_IRLP ((uint16_t)0x0004)
- #define USART_CR3_HDSEL ((uint16_t)0x0008)
- #define USART_CR3_NACK ((uint16_t)0x0010)
- #define USART_CR3_SCEN ((uint16_t)0x0020)
- #define USART_CR3_DMAR ((uint16_t)0x0040)
- #define USART_CR3_DMAT ((uint16_t)0x0080)
- #define USART_CR3_RTSE ((uint16_t)0x0100)
- #define USART_CR3_CTSE ((uint16_t)0x0200)
- #define USART_CR3_CTSIE ((uint16_t)0x0400)
- #define USART_CR3_ONEBIT ((uint16_t)0x0800)
- #define USART_GTPR_PSC ((uint16_t)0x00FF)
- #define USART_GTPR_PSC_0 ((uint16_t)0x0001)
- #define USART_GTPR_PSC_1 ((uint16_t)0x0002)
- #define USART_GTPR_PSC_2 ((uint16_t)0x0004)
- #define USART_GTPR_PSC_3 ((uint16_t)0x0008)
- #define USART_GTPR_PSC_4 ((uint16_t)0x0010)

- #define USART_GTPR_PSC_5 ((uint16_t)0x0020)
- #define USART_GTPR_PSC_6 ((uint16_t)0x0040)
- #define USART_GTPR_PSC_7 ((uint16_t)0x0080)
- #define USART_GTPR_GT ((uint16_t)0xFF00)
- #define WWDG_CR_T ((uint8_t)0x7F)
- #define WWDG_CR_T_0 ((uint8_t)0x01)
- #define WWDG_CR_T_1 ((uint8_t)0x02)
- #define WWDG_CR_T_2 ((uint8_t)0x04)
- #define WWDG_CR_T_3 ((uint8_t)0x08)
- #define WWDG_CR_T_4 ((uint8_t)0x10)
- #define WWDG_CR_T_5 ((uint8_t)0x20)
- #define WWDG_CR_T_6 ((uint8_t)0x40)
- #define WWDG_CR_T0 WWDG_CR_T_0
- #define WWDG_CR_T1 WWDG_CR_T_1
- #define WWDG_CR_T2 WWDG_CR_T_2
- #define WWDG_CR_T3 WWDG_CR_T_3
- #define WWDG_CR_T4 WWDG_CR_T_4
- #define WWDG_CR_T5 WWDG_CR_T_5
- #define WWDG_CR_T6 WWDG_CR_T_6
- #define WWDG_CFR_WDGA ((uint8_t)0x80)
- #define WWDG_CFR_W ((uint16_t)0x007F)
- #define WWDG_CFR_W_0 ((uint16_t)0x0001)
- #define WWDG_CFR_W_1 ((uint16_t)0x0002)
- #define WWDG_CFR_W_2 ((uint16_t)0x0004)
- #define WWDG_CFR_W_3 ((uint16_t)0x0008)
- #define WWDG_CFR_W_4 ((uint16_t)0x0010)
- #define WWDG_CFR_W_5 ((uint16_t)0x0020)
- #define WWDG_CFR_W_6 ((uint16_t)0x0040)
- #define WWDG_CFR_W0 WWDG_CFR_W_0
- #define WWDG_CFR_W1 WWDG_CFR_W_1
- #define WWDG_CFR_W2 WWDG_CFR_W_2
- #define WWDG_CFR_W3 WWDG_CFR_W_3
- #define WWDG_CFR_W4 WWDG_CFR_W_4
- #define WWDG_CFR_W5 WWDG_CFR_W_5
- #define WWDG_CFR_W6 WWDG_CFR_W_6
- #define WWDG_CFR_WDGTB ((uint16_t)0x0180)
- #define WWDG_CFR_WDGTB_0 ((uint16_t)0x0080)
- #define WWDG_CFR_WDGTB_1 ((uint16_t)0x0100)
- #define WWDG_CFR_WDGTB0 WWDG_CFR_WDGTB_0
- #define WWDG_CFR_WDGTB1 WWDG_CFR_WDGTB_1
- #define WWDG_CFR_EWI ((uint16_t)0x0200)
- #define WWDG_SR_EWIF ((uint8_t)0x01)
- #define DBGMCU_IDCODE_DEV_ID ((uint32_t)0x00000FFF)
- #define DBGMCU_IDCODE_REV_ID ((uint32_t)0xFFFF0000)
- #define DBGMCU_CR_DBG_SLEEP ((uint32_t)0x00000001)
- #define DBGMCU_CR_DBG_STOP ((uint32_t)0x00000002)
- #define DBGMCU_CR_DBG_STANDBY ((uint32_t)0x00000004)
- #define DBGMCU_CR_TRACE_IOEN ((uint32_t)0x00000020)
- #define DBGMCU_CR_TRACE_MODE ((uint32_t)0x000000C0)
- #define DBGMCU_CR_TRACE_MODE_0 ((uint32_t)0x00000040)
- #define DBGMCU_CR_TRACE_MODE_1 ((uint32_t)0x00000080)
- #define DBGMCU_APB1_FZ_DBG_TIM2_STOP ((uint32_t)0x00000001)
- #define DBGMCU_APB1_FZ_DBG_TIM3_STOP ((uint32_t)0x00000002)
- #define DBGMCU_APB1_FZ_DBG_TIM4_STOP ((uint32_t)0x00000004)
- #define DBGMCU_APB1_FZ_DBG_TIM5_STOP ((uint32_t)0x00000008)

- #define **DBGMCU_APB1_FZ_DBG_TIM6_STOP** ((uint32_t)0x00000010)
- #define **DBGMCU_APB1_FZ_DBG_TIM7_STOP** ((uint32_t)0x00000020)
- #define **DBGMCU_APB1_FZ_DBG_TIM12_STOP** ((uint32_t)0x00000040)
- #define **DBGMCU_APB1_FZ_DBG_TIM13_STOP** ((uint32_t)0x00000080)
- #define **DBGMCU_APB1_FZ_DBG_TIM14_STOP** ((uint32_t)0x00000100)
- #define **DBGMCU_APB1_FZ_DBG_RTC_STOP** ((uint32_t)0x00000400)
- #define **DBGMCU_APB1_FZ_DBG_WWDG_STOP** ((uint32_t)0x00000800)
- #define **DBGMCU_APB1_FZ_DBG_IWDG_STOP** ((uint32_t)0x00001000)
- #define **DBGMCU_APB1_FZ_DBG_I2C1_SMBUS_TIMEOUT** ((uint32_t)0x00200000)
- #define **DBGMCU_APB1_FZ_DBG_I2C2_SMBUS_TIMEOUT** ((uint32_t)0x00400000)
- #define **DBGMCU_APB1_FZ_DBG_I2C3_SMBUS_TIMEOUT** ((uint32_t)0x00800000)
- #define **DBGMCU_APB1_FZ_DBG_CAN1_STOP** ((uint32_t)0x02000000)
- #define **DBGMCU_APB1_FZ_DBG_CAN2_STOP** ((uint32_t)0x04000000)
- #define **DBGMCU_APB1_FZ_DBG_IWDEG_STOP** **DBGMCU_APB1_FZ_DBG_IWDG_STOP**
- #define **DBGMCU_APB1_FZ_DBG_TIM1_STOP** ((uint32_t)0x00000001)
- #define **DBGMCU_APB1_FZ_DBG_TIM8_STOP** ((uint32_t)0x00000002)
- #define **DBGMCU_APB1_FZ_DBG_TIM9_STOP** ((uint32_t)0x00010000)
- #define **DBGMCU_APB1_FZ_DBG_TIM10_STOP** ((uint32_t)0x00020000)
- #define **DBGMCU_APB1_FZ_DBG_TIM11_STOP** ((uint32_t)0x00040000)
- #define **ETH_MACCR_WD** ((uint32_t)0x00800000) /* Watchdog disable */
- #define **ETH_MACCR_JD** ((uint32_t)0x00400000) /* Jabber disable */
- #define **ETH_MACCR_IFG** ((uint32_t)0x000E0000) /* Inter-frame gap */
- #define **ETH_MACCR_IFG_96Bit** ((uint32_t)0x00000000) /* Minimum IFG between frames during transmission is 96Bit */
- #define **ETH_MACCR_IFG_88Bit** ((uint32_t)0x00020000) /* Minimum IFG between frames during transmission is 88Bit */
- #define **ETH_MACCR_IFG_80Bit** ((uint32_t)0x00040000) /* Minimum IFG between frames during transmission is 80Bit */
- #define **ETH_MACCR_IFG_72Bit** ((uint32_t)0x00060000) /* Minimum IFG between frames during transmission is 72Bit */
- #define **ETH_MACCR_IFG_64Bit** ((uint32_t)0x00080000) /* Minimum IFG between frames during transmission is 64Bit */
- #define **ETH_MACCR_IFG_56Bit** ((uint32_t)0x000A0000) /* Minimum IFG between frames during transmission is 56Bit */
- #define **ETH_MACCR_IFG_48Bit** ((uint32_t)0x000C0000) /* Minimum IFG between frames during transmission is 48Bit */
- #define **ETH_MACCR_IFG_40Bit** ((uint32_t)0x000E0000) /* Minimum IFG between frames during transmission is 40Bit */
- #define **ETH_MACCR_CSD** ((uint32_t)0x00010000) /* Carrier sense disable (during transmission) */
- #define **ETH_MACCR_FES** ((uint32_t)0x00004000) /* Fast ethernet speed */
- #define **ETH_MACCR_ROD** ((uint32_t)0x00002000) /* Receive own disable */
- #define **ETH_MACCR_LM** ((uint32_t)0x00001000) /* loopback mode */
- #define **ETH_MACCR_DM** ((uint32_t)0x00000800) /* Duplex mode */
- #define **ETH_MACCR_IPCO** ((uint32_t)0x00000400) /* IP Checksum offload */
- #define **ETH_MACCR_RD** ((uint32_t)0x00000200) /* Retry disable */
- #define **ETH_MACCR_APCS** ((uint32_t)0x00000080) /* Automatic Pad/**CRC** stripping */
- #define **ETH_MACCR_BL**
- #define **ETH_MACCR_BL_10** ((uint32_t)0x00000000) /* k = min (n, 10) */
- #define **ETH_MACCR_BL_8** ((uint32_t)0x00000020) /* k = min (n, 8) */
- #define **ETH_MACCR_BL_4** ((uint32_t)0x00000040) /* k = min (n, 4) */
- #define **ETH_MACCR_BL_1** ((uint32_t)0x00000060) /* k = min (n, 1) */
- #define **ETH_MACCR_DC** ((uint32_t)0x00000010) /* Defferal check */
- #define **ETH_MACCR_TE** ((uint32_t)0x00000008) /* Transmitter enable */
- #define **ETH_MACCR_RE** ((uint32_t)0x00000004) /* Receiver enable */
- #define **ETH_MACFFR_RA** ((uint32_t)0x80000000) /* Receive all */

- #define **ETH_MACFFR_HPF** ((uint32_t)0x000000400) /* Hash or perfect filter */
- #define **ETH_MACFFR_SAF** ((uint32_t)0x000000200) /* Source address filter enable */
- #define **ETH_MACFFR_SAIF** ((uint32_t)0x000000100) /* SA inverse filtering */
- #define **ETH_MACFFR_PCF** ((uint32_t)0x000000C0) /* Pass control frames: 3 cases */
- #define **ETH_MACFFR_PCF_BlockAll** ((uint32_t)0x00000040) /* MAC filters all control frames from reaching the application */
- #define **ETH_MACFFR_PCF_ForwardAll** ((uint32_t)0x00000080) /* MAC forwards all control frames to application even if they fail the Address Filter */
- #define **ETH_MACFFR_PCF_ForwardPassedAddrFilter** ((uint32_t)0x000000C0) /* MAC forwards control frames that pass the Address Filter. */
- #define **ETH_MACFFR_BFD** ((uint32_t)0x00000020) /* Broadcast frame disable */
- #define **ETH_MACFFR_PAM** ((uint32_t)0x00000010) /* Pass all multicast */
- #define **ETH_MACFFR_DAIF** ((uint32_t)0x00000008) /* DA Inverse filtering */
- #define **ETH_MACFFR_HM** ((uint32_t)0x00000004) /* Hash multicast */
- #define **ETH_MACFFR_HU** ((uint32_t)0x00000002) /* Hash unicast */
- #define **ETH_MACFFR_PM** ((uint32_t)0x00000001) /* Promiscuous mode */
- #define **ETH_MACHTHR_HTH** ((uint32_t)0xFFFFFFFF) /* Hash table high */
- #define **ETH_MACHTLR_HTL** ((uint32_t)0xFFFFFFFF) /* Hash table low */
- #define **ETH_MACMIIAR_PA** ((uint32_t)0x0000F800) /* Physical layer address */
- #define **ETH_MACMIIAR_MR** ((uint32_t)0x000007C0) /* MII register in the selected PHY */
- #define **ETH_MACMIIAR_CR** ((uint32_t)0x0000001C) /* CR clock range: 6 cases */
- #define **ETH_MACMIIAR_CR_Div42** ((uint32_t)0x00000000) /* HCLK:60-100 MHz; MDC clock= HCLK/42 */
- #define **ETH_MACMIIAR_CR_Div62** ((uint32_t)0x00000004) /* HCLK:100-150 MHz; MDC clock= HCLK/62 */
- #define **ETH_MACMIIAR_CR_Div16** ((uint32_t)0x00000008) /* HCLK:20-35 MHz; MDC clock= HCLK/16 */
- #define **ETH_MACMIIAR_CR_Div26** ((uint32_t)0x0000000C) /* HCLK:35-60 MHz; MDC clock= HCLK/26 */
- #define **ETH_MACMIIAR_CR_Div102** ((uint32_t)0x00000010) /* HCLK:150-168 MHz; MDC clock= HCLK/102 */
- #define **ETH_MACMIIAR_MW** ((uint32_t)0x00000002) /* MII write */
- #define **ETH_MACMIIAR_MB** ((uint32_t)0x00000001) /* MII busy */
- #define **ETH_MACMIIDR_MD** ((uint32_t)0x0000FFFF) /* MII data: read/write data from/to PHY */
- #define **ETH_MACFCR_PT** ((uint32_t)0xFFFF0000) /* Pause time */
- #define **ETH_MACFCR_ZQPD** ((uint32_t)0x00000080) /* Zero-quanta pause disable */
- #define **ETH_MACFCR_PLT** ((uint32_t)0x00000030) /* Pause low threshold: 4 cases */
- #define **ETH_MACFCR_PLT_Minus4** ((uint32_t)0x00000000) /* Pause time minus 4 slot times */
- #define **ETH_MACFCR_PLT_Minus28** ((uint32_t)0x00000010) /* Pause time minus 28 slot times */
- #define **ETH_MACFCR_PLT_Minus144** ((uint32_t)0x00000020) /* Pause time minus 144 slot times */
- #define **ETH_MACFCR_PLT_Minus256** ((uint32_t)0x00000030) /* Pause time minus 256 slot times */
- #define **ETH_MACFCR_UPFD** ((uint32_t)0x00000008) /* Unicast pause frame detect */
- #define **ETH_MACFCR_RFCE** ((uint32_t)0x00000004) /* Receive flow control enable */
- #define **ETH_MACFCR_TFCE** ((uint32_t)0x00000002) /* Transmit flow control enable */
- #define **ETH_MACFCR_FCBBPA** ((uint32_t)0x00000001) /* Flow control busy/backpressure activate */
- #define **ETH_MACVLANTR_VLANTC** ((uint32_t)0x00010000) /* 12-bit VLAN tag comparison */
- #define **ETH_MACVLANTR_VLANTI** ((uint32_t)0x0000FFFF) /* VLAN tag identifier (for receive frames) */
- #define **ETH_MACRWUFFR_D** ((uint32_t)0xFFFFFFFF) /* Wake-up frame filter register data */
- #define **ETH_MACPMTCSR_WFFRPPR** ((uint32_t)0x80000000) /* Wake-Up Frame Filter Register Pointer Reset */
- #define **ETH_MACPMTCSR_GU** ((uint32_t)0x00000200) /* Global Unicast */
- #define **ETH_MACPMTCSR_WFR** ((uint32_t)0x00000040) /* Wake-Up Frame Received */
- #define **ETH_MACPMTCSR_MPR** ((uint32_t)0x00000020) /* Magic Packet Received */
- #define **ETH_MACPMTCSR_WFE** ((uint32_t)0x00000004) /* Wake-Up Frame Enable */
- #define **ETH_MACPMTCSR_MPE** ((uint32_t)0x00000002) /* Magic Packet Enable */
- #define **ETH_MACPMTCSR_PD** ((uint32_t)0x00000001) /* Power Down */

- #define **ETH_MACSR_TSTS** ((uint32_t)0x000000200) /* Time stamp trigger status */
- #define **ETH_MACSR_MMCTS** ((uint32_t)0x000000040) /* MMC transmit status */
- #define **ETH_MACSR_MMMCRS** ((uint32_t)0x000000020) /* MMC receive status */
- #define **ETH_MACSR_MMCS** ((uint32_t)0x000000010) /* MMC status */
- #define **ETH_MACSR_PMTS** ((uint32_t)0x000000008) /* PMT status */
- #define **ETH_MACIMR_TSTIM** ((uint32_t)0x000000200) /* Time stamp trigger interrupt mask */
- #define **ETH_MACIMR_PMTIM** ((uint32_t)0x000000008) /* PMT interrupt mask */
- #define **ETH_MACA0HR_MACA0H** ((uint32_t)0x0000FFFF) /* MAC address0 high */
- #define **ETH_MACA0LR_MACA0L** ((uint32_t)0xFFFFFFFF) /* MAC address0 low */
- #define **ETH_MACA1HR_AE** ((uint32_t)0x80000000) /* Address enable */
- #define **ETH_MACA1HR_SA** ((uint32_t)0x40000000) /* Source address */
- #define **ETH_MACA1HR_MBC** ((uint32_t)0x3F000000) /* Mask byte control: bits to mask for comparison of the MAC Address bytes */
- #define **ETH_MACA1HR_MBC_HBits15_8** ((uint32_t)0x20000000) /* Mask MAC Address high reg bits [15:8] */
- #define **ETH_MACA1HR_MBC_HBits7_0** ((uint32_t)0x10000000) /* Mask MAC Address high reg bits [7:0] */
- #define **ETH_MACA1HR_MBC_LBits31_24** ((uint32_t)0x08000000) /* Mask MAC Address low reg bits [31:24] */
- #define **ETH_MACA1HR_MBC_LBits23_16** ((uint32_t)0x04000000) /* Mask MAC Address low reg bits [23:16] */
- #define **ETH_MACA1HR_MBC_LBits15_8** ((uint32_t)0x02000000) /* Mask MAC Address low reg bits [15:8] */
- #define **ETH_MACA1HR_MBC_LBits7_0** ((uint32_t)0x01000000) /* Mask MAC Address low reg bits [7:0] */
- #define **ETH_MACA1HR_MACA1H** ((uint32_t)0x0000FFFF) /* MAC address1 high */
- #define **ETH_MACA1LR_MACA1L** ((uint32_t)0xFFFFFFFF) /* MAC address1 low */
- #define **ETH_MACA2HR_AE** ((uint32_t)0x80000000) /* Address enable */
- #define **ETH_MACA2HR_SA** ((uint32_t)0x40000000) /* Source address */
- #define **ETH_MACA2HR_MBC** ((uint32_t)0x3F000000) /* Mask byte control */
- #define **ETH_MACA2HR_MBC_HBits15_8** ((uint32_t)0x20000000) /* Mask MAC Address high reg bits [15:8] */
- #define **ETH_MACA2HR_MBC_HBits7_0** ((uint32_t)0x10000000) /* Mask MAC Address high reg bits [7:0] */
- #define **ETH_MACA2HR_MBC_LBits31_24** ((uint32_t)0x08000000) /* Mask MAC Address low reg bits [31:24] */
- #define **ETH_MACA2HR_MBC_LBits23_16** ((uint32_t)0x04000000) /* Mask MAC Address low reg bits [23:16] */
- #define **ETH_MACA2HR_MBC_LBits15_8** ((uint32_t)0x02000000) /* Mask MAC Address low reg bits [15:8] */
- #define **ETH_MACA2HR_MBC_LBits7_0** ((uint32_t)0x01000000) /* Mask MAC Address low reg bits [7:0] */
- #define **ETH_MACA2HR_MACA2H** ((uint32_t)0x0000FFFF) /* MAC address1 high */
- #define **ETH_MACA2LR_MACA2L** ((uint32_t)0xFFFFFFFF) /* MAC address2 low */
- #define **ETH_MACA3HR_AE** ((uint32_t)0x80000000) /* Address enable */
- #define **ETH_MACA3HR_SA** ((uint32_t)0x40000000) /* Source address */
- #define **ETH_MACA3HR_MBC** ((uint32_t)0x3F000000) /* Mask byte control */
- #define **ETH_MACA3HR_MBC_HBits15_8** ((uint32_t)0x20000000) /* Mask MAC Address high reg bits [15:8] */
- #define **ETH_MACA3HR_MBC_HBits7_0** ((uint32_t)0x10000000) /* Mask MAC Address high reg bits [7:0] */
- #define **ETH_MACA3HR_MBC_LBits31_24** ((uint32_t)0x08000000) /* Mask MAC Address low reg bits [31:24] */
- #define **ETH_MACA3HR_MBC_LBits23_16** ((uint32_t)0x04000000) /* Mask MAC Address low reg bits [23:16] */
- #define **ETH_MACA3HR_MBC_LBits15_8** ((uint32_t)0x02000000) /* Mask MAC Address low reg bits [15:8] */

- #define **ETH_MACA3HR_MBC_LBits7_0** ((uint32_t)0x01000000) /* Mask MAC Address low reg bits [70] */
- #define **ETH_MACA3HR_MACA3H** ((uint32_t)0x0000FFFF) /* MAC address3 high */
- #define **ETH_MACA3LR_MACA3L** ((uint32_t)0xFFFFFFF) /* MAC address3 low */
- #define **ETH_MMCCR_MCFHP** ((uint32_t)0x00000020) /* MMC counter Full-Half preset */
- #define **ETH_MMCCR_MCP** ((uint32_t)0x00000010) /* MMC counter preset */
- #define **ETH_MMCCR_MCF** ((uint32_t)0x00000008) /* MMC Counter Freeze */
- #define **ETH_MMCCR_ROR** ((uint32_t)0x00000004) /* **Reset** on Read */
- #define **ETH_MMCCR_CSR** ((uint32_t)0x00000002) /* Counter Stop Rollover */
- #define **ETH_MMCCR_CR** ((uint32_t)0x00000001) /* Counters **Reset** */
- #define **ETH_MMCRIR_RGUFS** ((uint32_t)0x00020000) /* **Set** when Rx good unicast frames counter reaches half the maximum value */
- #define **ETH_MMCRIR_RFAES** ((uint32_t)0x00000040) /* **Set** when Rx alignment error counter reaches half the maximum value */
- #define **ETH_MMCRIR_RFCES** ((uint32_t)0x00000020) /* **Set** when Rx crc error counter reaches half the maximum value */
- #define **ETH_MMCTIR_TGFS** ((uint32_t)0x00200000) /* **Set** when Tx good frame count counter reaches half the maximum value */
- #define **ETH_MMCTIR_TGFMSCS** ((uint32_t)0x00008000) /* **Set** when Tx good multi col counter reaches half the maximum value */
- #define **ETH_MMCTIR_TGFSCS** ((uint32_t)0x00004000) /* **Set** when Tx good single col counter reaches half the maximum value */
- #define **ETH_MMCRIMR_RGUFM** ((uint32_t)0x00020000) /* Mask the interrupt when Rx good unicast frames counter reaches half the maximum value */
- #define **ETH_MMCRIMR_RFAEM** ((uint32_t)0x00000040) /* Mask the interrupt when Rx alignment error counter reaches half the maximum value */
- #define **ETH_MMCRIMR_RFCEM** ((uint32_t)0x00000020) /* Mask the interrupt when Rx crc error counter reaches half the maximum value */
- #define **ETH_MMCTIMR_TGFM** ((uint32_t)0x00200000) /* Mask the interrupt when Tx good frame count counter reaches half the maximum value */
- #define **ETH_MMCTIMR_TGFMSCM** ((uint32_t)0x00008000) /* Mask the interrupt when Tx good multi col counter reaches half the maximum value */
- #define **ETH_MMCTIMR_TGFSCM** ((uint32_t)0x00004000) /* Mask the interrupt when Tx good single col counter reaches half the maximum value */
- #define **ETH_MMCTGFSCCR_TGFSCC** ((uint32_t)0xFFFFFFFF) /* Number of successfully transmitted frames after a single collision in Half-duplex mode. */
- #define **ETH_MMCTGFMSCCR_TGFMSCC** ((uint32_t)0xFFFFFFFF) /* Number of successfully transmitted frames after more than a single collision in Half-duplex mode. */
- #define **ETH_MMCTGFCCR_TGFC** ((uint32_t)0xFFFFFFFF) /* Number of good frames transmitted. */
- #define **ETH_MMCRFCECR_RFCEC** ((uint32_t)0xFFFFFFFF) /* Number of frames received with **CRC** error. */
- #define **ETH_MMCRFAECR_RFAEC** ((uint32_t)0xFFFFFFFF) /* Number of frames received with alignment (dribble) error */
- #define **ETH_MMCRGUFCR_RGUFC** ((uint32_t)0xFFFFFFFF) /* Number of good unicast frames received. */
- #define **ETH_PTPTSCR_TSCNT** ((uint32_t)0x00030000) /* Time stamp clock node type */
- #define **ETH_PTPTSSR_TSSMRME** ((uint32_t)0x00008000) /* Time stamp snapshot for message relevant to master enable */
- #define **ETH_PTPTSSR_TSSEME** ((uint32_t)0x00004000) /* Time stamp snapshot for event message enable */
- #define **ETH_PTPTSSR_TSSIPV4FE** ((uint32_t)0x00002000) /* Time stamp snapshot for IPv4 frames enable */
- #define **ETH_PTPTSSR_TSSIPV6FE** ((uint32_t)0x00001000) /* Time stamp snapshot for IPv6 frames enable */
- #define **ETH_PTPTSSR_TSSPTPOEFE** ((uint32_t)0x00000800) /* Time stamp snapshot for PTP over ethernet frames enable */

- #define **ETH_PTPTSSR_TSPTPPSV2E** ((uint32_t)0x000000400) /* Time stamp PTP packet snooping for version2 format enable */
- #define **ETH_PTPTSSR_TSSSR** ((uint32_t)0x000000200) /* Time stamp Sub-seconds rollover */
- #define **ETH_PTPTSSR_TSSARFE** ((uint32_t)0x000000100) /* Time stamp snapshot for all received frames enable */
- #define **ETH_PTPTSCR_TSARU** ((uint32_t)0x000000020) /* Addend register update */
- #define **ETH_PTPTSCR_TSITE** ((uint32_t)0x000000010) /* Time stamp interrupt trigger enable */
- #define **ETH_PTPTSCR_TSSTU** ((uint32_t)0x000000008) /* Time stamp update */
- #define **ETH_PTPTSCR_TSSTI** ((uint32_t)0x000000004) /* Time stamp initialize */
- #define **ETH_PTPTSCR_TSFCU** ((uint32_t)0x000000002) /* Time stamp fine or coarse update */
- #define **ETH_PTPTSCR_TSE** ((uint32_t)0x000000001) /* Time stamp enable */
- #define **ETH_PTPSSIR_STSSI** ((uint32_t)0x000000FF) /* System time Sub-second increment value */
- #define **ETH_PTPTSHR_STS** ((uint32_t)0xFFFFFFFF) /* System Time second */
- #define **ETH_PTPTSLR_STPNS** ((uint32_t)0x800000000) /* System Time Positive or negative time */
- #define **ETH_PTPTSLR_STSS** ((uint32_t)0x7FFFFFFF) /* System Time sub-seconds */
- #define **ETH_PTPTSHUR_TSUS** ((uint32_t)0xFFFFFFFF) /* Time stamp update seconds */
- #define **ETH_PTPTSLUR_TSUPNS** ((uint32_t)0x800000000) /* Time stamp update Positive or negative time */
- #define **ETH_PTPTSLUR_TSUSS** ((uint32_t)0x7FFFFFFF) /* Time stamp update sub-seconds */
- #define **ETH_PTPTSAR_TSA** ((uint32_t)0xFFFFFFFF) /* Time stamp addend */
- #define **ETH_PTPTTHR_TTSH** ((uint32_t)0xFFFFFFFF) /* Target time stamp high */
- #define **ETH_PTPTTLR_TTSL** ((uint32_t)0xFFFFFFFF) /* Target time stamp low */
- #define **ETH_PTPTSSR_TSTTR** ((uint32_t)0x000000020) /* Time stamp target time reached */
- #define **ETH_PTPTSSR_TSSO** ((uint32_t)0x000000010) /* Time stamp seconds overflow */
- #define **ETH_DMABMR_AAB** ((uint32_t)0x02000000) /* Address-Aligned beats */
- #define **ETH_DMABMR_FPM** ((uint32_t)0x01000000) /* 4xPBL mode */
- #define **ETH_DMABMR_USP** ((uint32_t)0x00800000) /* Use separate PBL */
- #define **ETH_DMABMR_RDP** ((uint32_t)0x007E0000) /* RxDMA PBL */
- #define **ETH_DMABMR_RDP_1Beat** ((uint32_t)0x000200000) /* maximum number of beats to be transferred in one RxDMA transaction is 1 */
- #define **ETH_DMABMR_RDP_2Beat** ((uint32_t)0x000400000) /* maximum number of beats to be transferred in one RxDMA transaction is 2 */
- #define **ETH_DMABMR_RDP_4Beat** ((uint32_t)0x000800000) /* maximum number of beats to be transferred in one RxDMA transaction is 4 */
- #define **ETH_DMABMR_RDP_8Beat** ((uint32_t)0x001000000) /* maximum number of beats to be transferred in one RxDMA transaction is 8 */
- #define **ETH_DMABMR_RDP_16Beat** ((uint32_t)0x002000000) /* maximum number of beats to be transferred in one RxDMA transaction is 16 */
- #define **ETH_DMABMR_RDP_32Beat** ((uint32_t)0x004000000) /* maximum number of beats to be transferred in one RxDMA transaction is 32 */
- #define **ETH_DMABMR_RDP_4xPBL_4Beat** ((uint32_t)0x010200000) /* maximum number of beats to be transferred in one RxDMA transaction is 4 */
- #define **ETH_DMABMR_RDP_4xPBL_8Beat** ((uint32_t)0x010400000) /* maximum number of beats to be transferred in one RxDMA transaction is 8 */
- #define **ETH_DMABMR_RDP_4xPBL_16Beat** ((uint32_t)0x010800000) /* maximum number of beats to be transferred in one RxDMA transaction is 16 */
- #define **ETH_DMABMR_RDP_4xPBL_32Beat** ((uint32_t)0x011000000) /* maximum number of beats to be transferred in one RxDMA transaction is 32 */
- #define **ETH_DMABMR_RDP_4xPBL_64Beat** ((uint32_t)0x012000000) /* maximum number of beats to be transferred in one RxDMA transaction is 64 */
- #define **ETH_DMABMR_RDP_4xPBL_128Beat** ((uint32_t)0x014000000) /* maximum number of beats to be transferred in one RxDMA transaction is 128 */
- #define **ETH_DMABMR_FB** ((uint32_t)0x00010000) /* Fixed Burst */
- #define **ETH_DMABMR RTPR** ((uint32_t)0x0000C000) /* Rx Tx priority ratio */
- #define **ETH_DMABMR RTPR_1_1** ((uint32_t)0x00000000) /* Rx Tx priority ratio */

- #define **ETH_DMABMR_RTPR_2_1** ((uint32_t)0x00004000) /* Rx Tx priority ratio */
- #define **ETH_DMABMR_RTPR_3_1** ((uint32_t)0x00008000) /* Rx Tx priority ratio */
- #define **ETH_DMABMR_RTPR_4_1** ((uint32_t)0x0000C000) /* Rx Tx priority ratio */
- #define **ETH_DMABMR_PBL** ((uint32_t)0x00003F00) /* Programmable burst length */
- #define **ETH_DMABMR_PBL_1Beat** ((uint32_t)0x00000100) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 1 */
- #define **ETH_DMABMR_PBL_2Beat** ((uint32_t)0x00000200) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 2 */
- #define **ETH_DMABMR_PBL_4Beat** ((uint32_t)0x00000400) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 4 */
- #define **ETH_DMABMR_PBL_8Beat** ((uint32_t)0x00000800) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 8 */
- #define **ETH_DMABMR_PBL_16Beat** ((uint32_t)0x00001000) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 16 */
- #define **ETH_DMABMR_PBL_32Beat** ((uint32_t)0x00002000) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 32 */
- #define **ETH_DMABMR_PBL_4xPBL_4Beat** ((uint32_t)0x01000100) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 4 */
- #define **ETH_DMABMR_PBL_4xPBL_8Beat** ((uint32_t)0x01000200) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 8 */
- #define **ETH_DMABMR_PBL_4xPBL_16Beat** ((uint32_t)0x01000400) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 16 */
- #define **ETH_DMABMR_PBL_4xPBL_32Beat** ((uint32_t)0x01000800) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 32 */
- #define **ETH_DMABMR_PBL_4xPBL_64Beat** ((uint32_t)0x01001000) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 64 */
- #define **ETH_DMABMR_PBL_4xPBL_128Beat** ((uint32_t)0x01002000) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 128 */
- #define **ETH_DMABMR_EDE** ((uint32_t)0x00000080) /* Enhanced Descriptor Enable */
- #define **ETH_DMABMR_DSL** ((uint32_t)0x0000007C) /* Descriptor Skip Length */
- #define **ETH_DMABMR_DA** ((uint32_t)0x00000002) /* DMA arbitration scheme */
- #define **ETH_DMABMR_SR** ((uint32_t)0x00000001) /* Software reset */
- #define **ETH_DMATPDR_TPD** ((uint32_t)0xFFFFFFFF) /* Transmit poll demand */
- #define **ETH_DMARPDR_RPD** ((uint32_t)0xFFFFFFFF) /* Receive poll demand */
- #define **ETH_DMARDLAR_SRL** ((uint32_t)0xFFFFFFFF) /* Start of receive list */
- #define **ETH_DMATDLAR_STL** ((uint32_t)0xFFFFFFFF) /* Start of transmit list */
- #define **ETH_DMASR_TSTS** ((uint32_t)0x20000000) /* Time-stamp trigger status */
- #define **ETH_DMASR_PMTS** ((uint32_t)0x10000000) /* PMT status */
- #define **ETH_DMASR_MMCS** ((uint32_t)0x08000000) /* MMC status */
- #define **ETH_DMASR_EBS** ((uint32_t)0x03800000) /* Error bits status */
- #define **ETH_DMASR_EBS_DescAccess** ((uint32_t)0x02000000) /* Error bits 0-data buffer, 1-desc. access */
- #define **ETH_DMASR_EBS_ReadTransf** ((uint32_t)0x01000000) /* Error bits 0-write trnsf, 1-read transfr */
- #define **ETH_DMASR_EBS_DataTransfTx** ((uint32_t)0x00800000) /* Error bits 0-Rx DMA, 1-Tx DMA */
- #define **ETH_DMASR_TPS** ((uint32_t)0x00700000) /* Transmit process state */
- #define **ETH_DMASR_TPS_Stopped** ((uint32_t)0x00000000) /* Stopped - Reset or Stop Tx Command issued */
- #define **ETH_DMASR_TPS_Fetching** ((uint32_t)0x00100000) /* Running - fetching the Tx descriptor */
- #define **ETH_DMASR_TPS_Waiting** ((uint32_t)0x00200000) /* Running - waiting for status */
- #define **ETH_DMASR_TPS_Reading** ((uint32_t)0x00300000) /* Running - reading the data from host memory */
- #define **ETH_DMASR_TPS_Suspended** ((uint32_t)0x00600000) /* Suspended - Tx Descriptor unavailable */
- #define **ETH_DMASR_TPS_Closing** ((uint32_t)0x00700000) /* Running - closing Rx descriptor */
- #define **ETH_DMASR_RPS** ((uint32_t)0x000E0000) /* Receive process state */

- #define **ETH_DMASR_RPS_Stopped** ((uint32_t)0x00000000) /* Stopped - **Reset** or Stop Rx Command issued */
- #define **ETH_DMASR_RPS_Fetching** ((uint32_t)0x00020000) /* Running - fetching the Rx descriptor */
- #define **ETH_DMASR_RPS_Waiting** ((uint32_t)0x00060000) /* Running - waiting for packet */
- #define **ETH_DMASR_RPS_Suspended** ((uint32_t)0x00080000) /* Suspended - Rx Descriptor unavailable */
- #define **ETH_DMASR_RPS_Closing** ((uint32_t)0x000A0000) /* Running - closing descriptor */
- #define **ETH_DMASR_RPS_Queueing** ((uint32_t)0x000E0000) /* Running - queuing the receive frame into host memory */
- #define **ETH_DMASR_NIS** ((uint32_t)0x00010000) /* Normal interrupt summary */
- #define **ETH_DMASR_AIS** ((uint32_t)0x00008000) /* Abnormal interrupt summary */
- #define **ETH_DMASR_ERS** ((uint32_t)0x00004000) /* Early receive status */
- #define **ETH_DMASR_FBES** ((uint32_t)0x00002000) /* Fatal bus error status */
- #define **ETH_DMASR_ETS** ((uint32_t)0x00000400) /* Early transmit status */
- #define **ETH_DMASR_RWTS** ((uint32_t)0x00000200) /* Receive watchdog timeout status */
- #define **ETH_DMASR_RPSS** ((uint32_t)0x00000100) /* Receive process stopped status */
- #define **ETH_DMASR_RBUS** ((uint32_t)0x00000080) /* Receive **buffer** unavailable status */
- #define **ETH_DMASR_RS** ((uint32_t)0x00000040) /* Receive status */
- #define **ETH_DMASR_TUS** ((uint32_t)0x00000020) /* Transmit underflow status */
- #define **ETH_DMASR_ROS** ((uint32_t)0x00000010) /* Receive overflow status */
- #define **ETH_DMASR_TJTS** ((uint32_t)0x00000008) /* Transmit jabber timeout status */
- #define **ETH_DMASR_TBUS** ((uint32_t)0x00000004) /* Transmit **buffer** unavailable status */
- #define **ETH_DMASR_TPSS** ((uint32_t)0x00000002) /* Transmit process stopped status */
- #define **ETH_DMASR_TS** ((uint32_t)0x00000001) /* Transmit status */
- #define **ETH_DMAOMR_DTCEFD** ((uint32_t)0x04000000) /* Disable Dropping of TCP/IP checksum error frames */
- #define **ETH_DMAOMR_RSF** ((uint32_t)0x02000000) /* Receive store and forward */
- #define **ETH_DMAOMR_DFRF** ((uint32_t)0x01000000) /* Disable flushing of received frames */
- #define **ETH_DMAOMR_TSF** ((uint32_t)0x00200000) /* Transmit store and forward */
- #define **ETH_DMAOMR_FTF** ((uint32_t)0x00100000) /* Flush transmit FIFO */
- #define **ETH_DMAOMR_TTC** ((uint32_t)0x0001C000) /* Transmit threshold control */
- #define **ETH_DMAOMR_TTC_64Bytes** ((uint32_t)0x00000000) /* threshold level of the MTL Transmit FIFO is 64 Bytes */
- #define **ETH_DMAOMR_TTC_128Bytes** ((uint32_t)0x00004000) /* threshold level of the MTL Transmit FIFO is 128 Bytes */
- #define **ETH_DMAOMR_TTC_192Bytes** ((uint32_t)0x00008000) /* threshold level of the MTL Transmit FIFO is 192 Bytes */
- #define **ETH_DMAOMR_TTC_256Bytes** ((uint32_t)0x0000C000) /* threshold level of the MTL Transmit FIFO is 256 Bytes */
- #define **ETH_DMAOMR_TTC_40Bytes** ((uint32_t)0x00010000) /* threshold level of the MTL Transmit FIFO is 40 Bytes */
- #define **ETH_DMAOMR_TTC_32Bytes** ((uint32_t)0x00014000) /* threshold level of the MTL Transmit FIFO is 32 Bytes */
- #define **ETH_DMAOMR_TTC_24Bytes** ((uint32_t)0x00018000) /* threshold level of the MTL Transmit FIFO is 24 Bytes */
- #define **ETH_DMAOMR_TTC_16Bytes** ((uint32_t)0x0001C000) /* threshold level of the MTL Transmit FIFO is 16 Bytes */
- #define **ETH_DMAOMR_ST** ((uint32_t)0x00002000) /* **Start**/stop transmission command */
- #define **ETH_DMAOMR_FEF** ((uint32_t)0x00000080) /* Forward error frames */
- #define **ETH_DMAOMR_FUGF** ((uint32_t)0x00000040) /* Forward undersized good frames */
- #define **ETH_DMAOMR_RTC** ((uint32_t)0x00000018) /* receive threshold control */
- #define **ETH_DMAOMR_RTC_64Bytes** ((uint32_t)0x00000000) /* threshold level of the MTL Receive FIFO is 64 Bytes */
- #define **ETH_DMAOMR_RTC_32Bytes** ((uint32_t)0x00000008) /* threshold level of the MTL Receive FIFO is 32 Bytes */

- #define **ETH_DMAOMR_RTC_96Bytes** ((uint32_t)0x00000010) /* threshold level of the MTL Receive FIFO is 96 Bytes */
- #define **ETH_DMAOMR_RTC_128Bytes** ((uint32_t)0x00000018) /* threshold level of the MTL Receive FIFO is 128 Bytes */
- #define **ETH_DMAOMR_OSF** ((uint32_t)0x00000004) /* operate on second frame */
- #define **ETH_DMAOMR_SR** ((uint32_t)0x00000002) /* Start/stop receive */
- #define **ETH_DMAIER_NISE** ((uint32_t)0x00010000) /* Normal interrupt summary enable */
- #define **ETH_DMAIER_AISE** ((uint32_t)0x00008000) /* Abnormal interrupt summary enable */
- #define **ETH_DMAIER_ERIE** ((uint32_t)0x00004000) /* Early receive interrupt enable */
- #define **ETH_DMAIER_FBEIE** ((uint32_t)0x00002000) /* Fatal bus error interrupt enable */
- #define **ETH_DMAIER_ETIE** ((uint32_t)0x00000400) /* Early transmit interrupt enable */
- #define **ETH_DMAIER_RWTIE** ((uint32_t)0x00000200) /* Receive watchdog timeout interrupt enable */
- #define **ETH_DMAIER_RPSIE** ((uint32_t)0x00000100) /* Receive process stopped interrupt enable */
- #define **ETH_DMAIER_RBUIE** ((uint32_t)0x00000080) /* Receive buffer unavailable interrupt enable */
- #define **ETH_DMAIER_RIE** ((uint32_t)0x00000040) /* Receive interrupt enable */
- #define **ETH_DMAIER_TUIE** ((uint32_t)0x00000020) /* Transmit Underflow interrupt enable */
- #define **ETH_DMAIER_ROIE** ((uint32_t)0x00000010) /* Receive Overflow interrupt enable */
- #define **ETH_DMAIER_TJTIE** ((uint32_t)0x00000008) /* Transmit jabber timeout interrupt enable */
- #define **ETH_DMAIER_TBUIE** ((uint32_t)0x00000004) /* Transmit buffer unavailable interrupt enable */
- #define **ETH_DMAIER_TPSIE** ((uint32_t)0x00000002) /* Transmit process stopped interrupt enable */
- #define **ETH_DMAIER_TIE** ((uint32_t)0x00000001) /* Transmit interrupt enable */
- #define **ETH_DMAMFBOCR_OFOC** ((uint32_t)0x10000000) /* Overflow bit for FIFO overflow counter */
- #define **ETH_DMAMFBOCR_MFA** ((uint32_t)0x0FFE0000) /* Number of frames missed by the application */
- #define **ETH_DMAMFBOCR_OMFC** ((uint32_t)0x00010000) /* Overflow bit for missed frame counter */
- #define **ETH_DMAMFBOCR_MFC** ((uint32_t)0x0000FFFF) /* Number of frames missed by the controller */
- #define **ETH_DMACHTDR_HTDAP** ((uint32_t)0xFFFFFFFF) /* Host transmit descriptor address pointer */
- #define **ETH_DMACHRDR_HRDAP** ((uint32_t)0xFFFFFFFF) /* Host receive descriptor address pointer */
- #define **ETH_DMACHTBAR_HTBAP** ((uint32_t)0xFFFFFFFF) /* Host transmit buffer address pointer */
- #define **ETH_DMACHRBAR_HRBAP** ((uint32_t)0xFFFFFFFF) /* Host receive buffer address pointer */

5.173.1 Detailed Description

5.173.2 Macro Definition Documentation

5.173.2.1 ADC_CCR_ADCPRE

```
#define ADC_CCR_ADCPRE ((uint32_t)0x00030000)
```

ADCPRE[1:0] bits (ADC prescaler)

5.173.2.2 ADC_CCR_ADCPRE_0

```
#define ADC_CCR_ADCPRE_0 ((uint32_t)0x00010000)
```

Bit 0

5.173.2.3 ADC_CCR_ADCPRE_1

```
#define ADC_CCR_ADCPRE_1 ((uint32_t)0x00020000)
```

Bit 1

5.173.2.4 ADC_CCR_DDS

```
#define ADC_CCR_DDS ((uint32_t)0x00002000)
```

DMA disable selection (Multi-ADC mode)

5.173.2.5 ADC_CCR_DELAY

```
#define ADC_CCR_DELAY ((uint32_t)0x00000F00)
```

DELAY[3:0] bits (Delay between 2 sampling phases)

5.173.2.6 ADC_CCR_DELAY_0

```
#define ADC_CCR_DELAY_0 ((uint32_t)0x00000100)
```

Bit 0

5.173.2.7 ADC_CCR_DELAY_1

```
#define ADC_CCR_DELAY_1 ((uint32_t)0x00000200)
```

Bit 1

5.173.2.8 ADC_CCR_DELAY_2

```
#define ADC_CCR_DELAY_2 ((uint32_t)0x00000400)
```

Bit 2

5.173.2.9 ADC_CCR_DELAY_3

```
#define ADC_CCR_DELAY_3 ((uint32_t)0x00000800)
```

Bit 3

5.173.2.10 ADC_CCR_DMA

```
#define ADC_CCR_DMA ((uint32_t)0x0000C000)
```

DMA[1:0] bits (Direct Memory Access mode for multimode)

5.173.2.11 ADC_CCR_DMA_0

```
#define ADC_CCR_DMA_0 ((uint32_t)0x00004000)
```

Bit 0

5.173.2.12 ADC_CCR_DMA_1

```
#define ADC_CCR_DMA_1 ((uint32_t)0x00008000)
```

Bit 1

5.173.2.13 ADC_CCR_MULTI

```
#define ADC_CCR_MULTI ((uint32_t)0x0000001F)
```

MULTI[4:0] bits (Multi-ADC mode selection)

5.173.2.14 ADC_CCR_MULTI_0

```
#define ADC_CCR_MULTI_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.15 ADC_CCR_MULTI_1

```
#define ADC_CCR_MULTI_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.16 ADC_CCR_MULTI_2

```
#define ADC_CCR_MULTI_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.17 ADC_CCR_MULTI_3

```
#define ADC_CCR_MULTI_3 ((uint32_t)0x00000008)
```

Bit 3

5.173.2.18 ADC_CCR_MULTI_4

```
#define ADC_CCR_MULTI_4 ((uint32_t)0x00000010)
```

Bit 4

5.173.2.19 ADC_CCR_TSVREFE

```
#define ADC_CCR_TSVREFE ((uint32_t)0x00800000)
```

Temperature Sensor and VREFINT Enable

5.173.2.20 ADC_CCR_VBATE

```
#define ADC_CCR_VBATE ((uint32_t)0x00400000)
```

VBAT Enable

5.173.2.21 ADC_CDR_DATA1

```
#define ADC_CDR_DATA1 ((uint32_t)0x0000FFFF)
```

1st data of a pair of regular conversions

5.173.2.22 ADC_CDR_DATA2

```
#define ADC_CDR_DATA2 ((uint32_t)0xFFFF0000)
```

2nd data of a pair of regular conversions

5.173.2.23 ADC_CR1_AWDCH

```
#define ADC_CR1_AWDCH ((uint32_t)0x0000001F)
```

AWDCH[4:0] bits (Analog watchdog channel select bits)

5.173.2.24 ADC_CR1_AWDCH_0

```
#define ADC_CR1_AWDCH_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.25 ADC_CR1_AWDCH_1

```
#define ADC_CR1_AWDCH_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.26 ADC_CR1_AWDCH_2

```
#define ADC_CR1_AWDCH_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.27 ADC_CR1_AWDCH_3

```
#define ADC_CR1_AWDCH_3 ((uint32_t)0x00000008)
```

Bit 3

5.173.2.28 ADC_CR1_AWDCH_4

```
#define ADC_CR1_AWDCH_4 ((uint32_t)0x00000010)
```

Bit 4

5.173.2.29 ADC_CR1_AWDEN

```
#define ADC_CR1_AWDEN ((uint32_t)0x00800000)
```

Analog watchdog enable on regular channels

5.173.2.30 ADC_CR1_AWDIE

```
#define ADC_CR1_AWDIE ((uint32_t)0x00000040)
```

Analog Watchdog interrupt enable

5.173.2.31 ADC_CR1_AWDSGL

```
#define ADC_CR1_AWDSGL ((uint32_t)0x00000200)
```

Enable the watchdog on a single channel in scan mode

5.173.2.32 ADC_CR1_DISCEN

```
#define ADC_CR1_DISCEN ((uint32_t)0x00000800)
```

Discontinuous mode on regular channels

5.173.2.33 ADC_CR1_DISCNUM

```
#define ADC_CR1_DISCNUM ((uint32_t)0x0000E000)
```

DISCNUM[2:0] bits (Discontinuous mode channel count)

5.173.2.34 ADC_CR1_DISCNUM_0

```
#define ADC_CR1_DISCNUM_0 ((uint32_t)0x00002000)
```

Bit 0

5.173.2.35 ADC_CR1_DISCNUM_1

```
#define ADC_CR1_DISCNUM_1 ((uint32_t)0x00004000)
```

Bit 1

5.173.2.36 ADC_CR1_DISCNUM_2

```
#define ADC_CR1_DISCNUM_2 ((uint32_t)0x00008000)
```

Bit 2

5.173.2.37 ADC_CR1_EOCIE

```
#define ADC_CR1_EOCIE ((uint32_t)0x00000020)
```

Interrupt enable for EOC

5.173.2.38 ADC_CR1_JAUTO

```
#define ADC_CR1_JAUTO ((uint32_t)0x00000400)
```

Automatic injected group conversion

5.173.2.39 ADC_CR1_JAWDEN

```
#define ADC_CR1_JAWDEN ((uint32_t)0x00400000)
```

Analog watchdog enable on injected channels

5.173.2.40 ADC_CR1_JDISCEN

```
#define ADC_CR1_JDISCEN ((uint32_t)0x00001000)
```

Discontinuous mode on injected channels

5.173.2.41 ADC_CR1_JEOCIE

```
#define ADC_CR1_JEOCIE ((uint32_t)0x00000080)
```

Interrupt enable for injected channels

5.173.2.42 ADC_CR1_OVRIE

```
#define ADC_CR1_OVRIE ((uint32_t)0x04000000)
```

overrun interrupt enable

5.173.2.43 ADC_CR1_RES

```
#define ADC_CR1_RES ((uint32_t)0x03000000)
```

RES[2:0] bits (Resolution)

5.173.2.44 ADC_CR1_RES_0

```
#define ADC_CR1_RES_0 ((uint32_t)0x01000000)
```

Bit 0

5.173.2.45 ADC_CR1_RES_1

```
#define ADC_CR1_RES_1 ((uint32_t)0x02000000)
```

Bit 1

5.173.2.46 ADC_CR1_SCAN

```
#define ADC_CR1_SCAN ((uint32_t)0x00000100)
```

Scan mode

5.173.2.47 ADC_CR2_ADON

```
#define ADC_CR2_ADON ((uint32_t)0x00000001)
```

A/D Converter ON / OFF

5.173.2.48 ADC_CR2_ALIGN

```
#define ADC_CR2_ALIGN ((uint32_t)0x00000800)
```

Data Alignment

5.173.2.49 ADC_CR2_CONT

```
#define ADC_CR2_CONT ((uint32_t)0x00000002)
```

Continuous Conversion

5.173.2.50 ADC_CR2_DDS

```
#define ADC_CR2_DDS ((uint32_t)0x00000200)
```

DMA disable selection (Single ADC)

5.173.2.51 ADC_CR2_DMA

```
#define ADC_CR2_DMA ((uint32_t)0x00000100)
```

Direct Memory access mode

5.173.2.52 ADC_CR2_EOCS

```
#define ADC_CR2_EOCS ((uint32_t)0x00000400)
```

End of conversion selection

5.173.2.53 ADC_CR2_EXTEN

```
#define ADC_CR2_EXTEN ((uint32_t)0x30000000)
```

EXTEN[1:0] bits (External Trigger Conversion mode for regular channelsp)

5.173.2.54 ADC_CR2_EXTEN_0

```
#define ADC_CR2_EXTEN_0 ((uint32_t)0x10000000)
```

Bit 0

5.173.2.55 ADC_CR2_EXTEN_1

```
#define ADC_CR2_EXTEN_1 ((uint32_t)0x20000000)
```

Bit 1

5.173.2.56 ADC_CR2_EXTSEL

```
#define ADC_CR2_EXTSEL ((uint32_t)0x0F000000)
```

EXTSEL[3:0] bits (External Event Select for regular group)

5.173.2.57 ADC_CR2_EXTSEL_0

```
#define ADC_CR2_EXTSEL_0 ((uint32_t)0x01000000)
```

Bit 0

5.173.2.58 ADC_CR2_EXTSEL_1

```
#define ADC_CR2_EXTSEL_1 ((uint32_t)0x02000000)
```

Bit 1

5.173.2.59 ADC_CR2_EXTSEL_2

```
#define ADC_CR2_EXTSEL_2 ((uint32_t)0x04000000)
```

Bit 2

5.173.2.60 ADC_CR2_EXTSEL_3

```
#define ADC_CR2_EXTSEL_3 ((uint32_t)0x08000000)
```

Bit 3

5.173.2.61 ADC_CR2_JEXTEN

```
#define ADC_CR2_JEXTEN ((uint32_t)0x00300000)
```

JEXTEN[1:0] bits (External Trigger Conversion mode for injected channelsp)

5.173.2.62 ADC_CR2_JEXTEN_0

```
#define ADC_CR2_JEXTEN_0 ((uint32_t)0x00100000)
```

Bit 0

5.173.2.63 ADC_CR2_JEXTEN_1

```
#define ADC_CR2_JEXTEN_1 ((uint32_t)0x00200000)
```

Bit 1

5.173.2.64 ADC_CR2_JEXTSEL

```
#define ADC_CR2_JEXTSEL ((uint32_t)0x000F0000)
```

JEXTSEL[3:0] bits (External event select for injected group)

5.173.2.65 ADC_CR2_JEXTSEL_0

```
#define ADC_CR2_JEXTSEL_0 ((uint32_t)0x00010000)
```

Bit 0

5.173.2.66 ADC_CR2_JEXTSEL_1

```
#define ADC_CR2_JEXTSEL_1 ((uint32_t)0x00020000)
```

Bit 1

5.173.2.67 ADC_CR2_JEXTSEL_2

```
#define ADC_CR2_JEXTSEL_2 ((uint32_t)0x00040000)
```

Bit 2

5.173.2.68 ADC_CR2_JEXTSEL_3

```
#define ADC_CR2_JEXTSEL_3 ((uint32_t)0x00080000)
```

Bit 3

5.173.2.69 ADC_CR2_JSWSTART

```
#define ADC_CR2_JSWSTART ((uint32_t)0x00400000)
```

Start Conversion of injected channels

5.173.2.70 ADC_CR2_SWSTART

```
#define ADC_CR2_SWSTART ((uint32_t)0x40000000)
```

Start Conversion of regular channels

5.173.2.71 ADC_CSR_AWD1

```
#define ADC_CSR_AWD1 ((uint32_t)0x00000001)
```

ADC1 Analog watchdog flag

5.173.2.72 ADC_CSR_AWD2

```
#define ADC_CSR_AWD2 ((uint32_t)0x00000100)
```

ADC2 Analog watchdog flag

5.173.2.73 ADC_CSR_AWD3

```
#define ADC_CSR_AWD3 ((uint32_t)0x00010000)
```

ADC3 Analog watchdog flag

5.173.2.74 ADC_CSR_EOC1

```
#define ADC_CSR_EOC1 ((uint32_t)0x00000002)
```

ADC1 End of conversion

5.173.2.75 ADC_CSR_EOC2

```
#define ADC_CSR_EOC2 ((uint32_t)0x00000200)
```

ADC2 End of conversion

5.173.2.76 ADC_CSR_EOC3

```
#define ADC_CSR_EOC3 ((uint32_t)0x00020000)
```

ADC3 End of conversion

5.173.2.77 ADC_CSR_JEOC1

```
#define ADC_CSR_JEOC1 ((uint32_t)0x00000004)
```

ADC1 Injected channel end of conversion

5.173.2.78 ADC_CSR_JEOC2

```
#define ADC_CSR_JEOC2 ((uint32_t)0x00000400)
```

ADC2 Injected channel end of conversion

5.173.2.79 ADC_CSR_JEOC3

```
#define ADC_CSR_JEOC3 ((uint32_t)0x00040000)
```

ADC3 Injected channel end of conversion

5.173.2.80 ADC_CSR_JSTRT1

```
#define ADC_CSR_JSTRT1 ((uint32_t)0x00000008)
```

ADC1 Injected channel Start flag

5.173.2.81 ADC_CSR_JSTRT2

```
#define ADC_CSR_JSTRT2 ((uint32_t)0x00000800)
```

ADC2 Injected channel Start flag

5.173.2.82 ADC_CSR_JSTRT3

```
#define ADC_CSR_JSTRT3 ((uint32_t)0x00080000)
```

ADC3 Injected channel Start flag

5.173.2.83 ADC_CSR_OVR1

```
#define ADC_CSR_OVR1 ((uint32_t)0x00000020)
```

ADC1 DMA overrun flag

5.173.2.84 ADC_CSR_OVR2

```
#define ADC_CSR_OVR2 ((uint32_t)0x00002000)
```

ADC2 DMA overrun flag

5.173.2.85 ADC_CSR_OVR3

```
#define ADC_CSR_OVR3 ((uint32_t)0x00200000)
```

ADC3 DMA overrun flag

5.173.2.86 ADC_CSR_STRT1

```
#define ADC_CSR_STRT1 ((uint32_t)0x00000010)
```

ADC1 Regular channel Start flag

5.173.2.87 ADC_CSR_STRT2

```
#define ADC_CSR_STRT2 ((uint32_t)0x00001000)
```

ADC2 Regular channel Start flag

5.173.2.88 ADC_CSR_STRT3

```
#define ADC_CSR_STRT3 ((uint32_t)0x00100000)
```

ADC3 Regular channel Start flag

5.173.2.89 ADC_DR_ADC2DATA

```
#define ADC_DR_ADC2DATA ((uint32_t)0xFFFF0000)
```

ADC2 data

5.173.2.90 ADC_DR_DATA

```
#define ADC_DR_DATA ((uint32_t)0x0000FFFF)
```

Regular data

5.173.2.91 ADC_HTR_HT

```
#define ADC_HTR_HT ((uint16_t)0x0FFF)
```

Analog watchdog high threshold

5.173.2.92 ADC_JDR1_JDATA

```
#define ADC_JDR1_JDATA ((uint16_t)0xFFFF)
```

Injected data

5.173.2.93 ADC_JDR2_JDATA

```
#define ADC_JDR2_JDATA ((uint16_t)0xFFFF)
```

Injected data

5.173.2.94 ADC_JDR3_JDATA

```
#define ADC_JDR3_JDATA ((uint16_t)0xFFFF)
```

Injected data

5.173.2.95 ADC_JDR4_JDATA

```
#define ADC_JDR4_JDATA ((uint16_t)0xFFFF)
```

Injected data

5.173.2.96 ADC_JOFR1_JOFFSET1

```
#define ADC_JOFR1_JOFFSET1 ((uint16_t)0x0FFF)
```

Data offset for injected channel 1

5.173.2.97 ADC_JOFR2_JOFFSET2

```
#define ADC_JOFR2_JOFFSET2 ((uint16_t)0x0FFF)
```

Data offset for injected channel 2

5.173.2.98 ADC_JOFR3_JOFFSET3

```
#define ADC_JOFR3_JOFFSET3 ((uint16_t)0x0FFF)
```

Data offset for injected channel 3

5.173.2.99 ADC_JOFR4_JOFFSET4

```
#define ADC_JOFR4_JOFFSET4 ((uint16_t)0x0FFF)
```

Data offset for injected channel 4

5.173.2.100 ADC_JSQR_JL

```
#define ADC_JSQR_JL ((uint32_t)0x00300000)
```

JL[1:0] bits (Injected Sequence length)

5.173.2.101 ADC_JSQR_JL_0

```
#define ADC_JSQR_JL_0 ((uint32_t)0x00100000)
```

Bit 0

5.173.2.102 ADC_JSQR_JL_1

```
#define ADC_JSQR_JL_1 ((uint32_t)0x00200000)
```

Bit 1

5.173.2.103 ADC_JSQR_JSQ1

```
#define ADC_JSQR_JSQ1 ((uint32_t)0x0000001F)
```

JSQ1[4:0] bits (1st conversion in injected sequence)

5.173.2.104 ADC_JSQR_JSQ1_0

```
#define ADC_JSQR_JSQ1_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.105 ADC_JSQR_JSQ1_1

```
#define ADC_JSQR_JSQ1_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.106 ADC_JSQR_JSQ1_2

```
#define ADC_JSQR_JSQ1_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.107 ADC_JSQR_JSQ1_3

```
#define ADC_JSQR_JSQ1_3 ((uint32_t)0x00000008)
```

Bit 3

5.173.2.108 ADC_JSQR_JSQ1_4

```
#define ADC_JSQR_JSQ1_4 ((uint32_t)0x00000010)
```

Bit 4

5.173.2.109 ADC_JSQR_JSQ2

```
#define ADC_JSQR_JSQ2 ((uint32_t)0x000003E0)
```

JSQ2[4:0] bits (2nd conversion in injected sequence)

5.173.2.110 ADC_JSQR_JSQ2_0

```
#define ADC_JSQR_JSQ2_0 ((uint32_t)0x00000020)
```

Bit 0

5.173.2.111 ADC_JSQR_JSQ2_1

```
#define ADC_JSQR_JSQ2_1 ((uint32_t)0x00000040)
```

Bit 1

5.173.2.112 ADC_JSQR_JSQ2_2

```
#define ADC_JSQR_JSQ2_2 ((uint32_t)0x00000080)
```

Bit 2

5.173.2.113 ADC_JSQR_JSQ2_3

```
#define ADC_JSQR_JSQ2_3 ((uint32_t)0x00000100)
```

Bit 3

5.173.2.114 ADC_JSQR_JSQ2_4

```
#define ADC_JSQR_JSQ2_4 ((uint32_t)0x00000200)
```

Bit 4

5.173.2.115 ADC_JSQR_JSQ3

```
#define ADC_JSQR_JSQ3 ((uint32_t)0x00007C00)
```

JSQ3[4:0] bits (3rd conversion in injected sequence)

5.173.2.116 ADC_JSQR_JSQ3_0

```
#define ADC_JSQR_JSQ3_0 ((uint32_t)0x00000400)
```

Bit 0

5.173.2.117 ADC_JSQR_JSQ3_1

```
#define ADC_JSQR_JSQ3_1 ((uint32_t)0x00000800)
```

Bit 1

5.173.2.118 ADC_JSQR_JSQ3_2

```
#define ADC_JSQR_JSQ3_2 ((uint32_t)0x00001000)
```

Bit 2

5.173.2.119 ADC_JSQR_JSQ3_3

```
#define ADC_JSQR_JSQ3_3 ((uint32_t)0x00002000)
```

Bit 3

5.173.2.120 ADC_JSQR_JSQ3_4

```
#define ADC_JSQR_JSQ3_4 ((uint32_t)0x00004000)
```

Bit 4

5.173.2.121 ADC_JSQR_JSQ4

```
#define ADC_JSQR_JSQ4 ((uint32_t)0x000F8000)
```

JSQ4[4:0] bits (4th conversion in injected sequence)

5.173.2.122 ADC_JSQR_JSQ4_0

```
#define ADC_JSQR_JSQ4_0 ((uint32_t)0x00008000)
```

Bit 0

5.173.2.123 ADC_JSQR_JSQ4_1

```
#define ADC_JSQR_JSQ4_1 ((uint32_t)0x00010000)
```

Bit 1

5.173.2.124 ADC_JSQR_JSQ4_2

```
#define ADC_JSQR_JSQ4_2 ((uint32_t)0x00020000)
```

Bit 2

5.173.2.125 ADC_JSQR_JSQ4_3

```
#define ADC_JSQR_JSQ4_3 ((uint32_t)0x00040000)
```

Bit 3

5.173.2.126 ADC_JSQR_JSQ4_4

```
#define ADC_JSQR_JSQ4_4 ((uint32_t)0x00080000)
```

Bit 4

5.173.2.127 ADC_LTR_LT

```
#define ADC_LTR_LT ((uint16_t)0xFFFF)
```

Analog watchdog low threshold

5.173.2.128 ADC_SMPR1_SMP10

```
#define ADC_SMPR1_SMP10 ((uint32_t)0x00000007)
```

SMP10[2:0] bits (Channel 10 Sample time selection)

5.173.2.129 ADC_SMPR1_SMP10_0

```
#define ADC_SMPR1_SMP10_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.130 ADC_SMPR1_SMP10_1

```
#define ADC_SMPR1_SMP10_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.131 ADC_SMPR1_SMP10_2

```
#define ADC_SMPR1_SMP10_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.132 ADC_SMPR1_SMP11

```
#define ADC_SMPR1_SMP11 ((uint32_t)0x00000038)
```

SMP11[2:0] bits (Channel 11 Sample time selection)

5.173.2.133 ADC_SMPR1_SMP11_0

```
#define ADC_SMPR1_SMP11_0 ((uint32_t)0x00000008)
```

Bit 0

5.173.2.134 ADC_SMPR1_SMP11_1

```
#define ADC_SMPR1_SMP11_1 ((uint32_t)0x00000010)
```

Bit 1

5.173.2.135 ADC_SMPR1_SMP11_2

```
#define ADC_SMPR1_SMP11_2 ((uint32_t)0x00000020)
```

Bit 2

5.173.2.136 ADC_SMPR1_SMP12

```
#define ADC_SMPR1_SMP12 ((uint32_t)0x000001C0)
```

SMP12[2:0] bits (Channel 12 Sample time selection)

5.173.2.137 ADC_SMPR1_SMP12_0

```
#define ADC_SMPR1_SMP12_0 ((uint32_t)0x00000040)
```

Bit 0

5.173.2.138 ADC_SMPR1_SMP12_1

```
#define ADC_SMPR1_SMP12_1 ((uint32_t)0x00000080)
```

Bit 1

5.173.2.139 ADC_SMPR1_SMP12_2

```
#define ADC_SMPR1_SMP12_2 ((uint32_t)0x00000100)
```

Bit 2

5.173.2.140 ADC_SMPR1_SMP13

```
#define ADC_SMPR1_SMP13 ((uint32_t)0x00000E00)
```

SMP13[2:0] bits (Channel 13 Sample time selection)

5.173.2.141 ADC_SMPR1_SMP13_0

```
#define ADC_SMPR1_SMP13_0 ((uint32_t)0x00000200)
```

Bit 0

5.173.2.142 ADC_SMPR1_SMP13_1

```
#define ADC_SMPR1_SMP13_1 ((uint32_t)0x00000400)
```

Bit 1

5.173.2.143 ADC_SMPR1_SMP13_2

```
#define ADC_SMPR1_SMP13_2 ((uint32_t)0x00000800)
```

Bit 2

5.173.2.144 ADC_SMPR1_SMP14

```
#define ADC_SMPR1_SMP14 ((uint32_t)0x00007000)
```

SMP14[2:0] bits (Channel 14 Sample time selection)

5.173.2.145 ADC_SMPR1_SMP14_0

```
#define ADC_SMPR1_SMP14_0 ((uint32_t)0x00001000)
```

Bit 0

5.173.2.146 ADC_SMPR1_SMP14_1

```
#define ADC_SMPR1_SMP14_1 ((uint32_t)0x00002000)
```

Bit 1

5.173.2.147 ADC_SMPR1_SMP14_2

```
#define ADC_SMPR1_SMP14_2 ((uint32_t)0x00004000)
```

Bit 2

5.173.2.148 ADC_SMPR1_SMP15

```
#define ADC_SMPR1_SMP15 ((uint32_t)0x00038000)
```

SMP15[2:0] bits (Channel 15 Sample time selection)

5.173.2.149 ADC_SMPR1_SMP15_0

```
#define ADC_SMPR1_SMP15_0 ((uint32_t)0x00008000)
```

Bit 0

5.173.2.150 ADC_SMPR1_SMP15_1

```
#define ADC_SMPR1_SMP15_1 ((uint32_t)0x00010000)
```

Bit 1

5.173.2.151 ADC_SMPR1_SMP15_2

```
#define ADC_SMPR1_SMP15_2 ((uint32_t)0x00020000)
```

Bit 2

5.173.2.152 ADC_SMPR1_SMP16

```
#define ADC_SMPR1_SMP16 ((uint32_t)0x001C0000)
```

SMP16[2:0] bits (Channel 16 Sample time selection)

5.173.2.153 ADC_SMPR1_SMP16_0

```
#define ADC_SMPR1_SMP16_0 ((uint32_t)0x00040000)
```

Bit 0

5.173.2.154 ADC_SMPR1_SMP16_1

```
#define ADC_SMPR1_SMP16_1 ((uint32_t)0x00080000)
```

Bit 1

5.173.2.155 ADC_SMPR1_SMP16_2

```
#define ADC_SMPR1_SMP16_2 ((uint32_t)0x00100000)
```

Bit 2

5.173.2.156 ADC_SMPR1_SMP17

```
#define ADC_SMPR1_SMP17 ((uint32_t)0x00E00000)
```

SMP17[2:0] bits (Channel 17 Sample time selection)

5.173.2.157 ADC_SMPR1_SMP17_0

```
#define ADC_SMPR1_SMP17_0 ((uint32_t)0x00200000)
```

Bit 0

5.173.2.158 ADC_SMPR1_SMP17_1

```
#define ADC_SMPR1_SMP17_1 ((uint32_t)0x00400000)
```

Bit 1

5.173.2.159 ADC_SMPR1_SMP17_2

```
#define ADC_SMPR1_SMP17_2 ((uint32_t)0x00800000)
```

Bit 2

5.173.2.160 ADC_SMPR1_SMP18

```
#define ADC_SMPR1_SMP18 ((uint32_t)0x07000000)
```

SMP18[2:0] bits (Channel 18 Sample time selection)

5.173.2.161 ADC_SMPR1_SMP18_0

```
#define ADC_SMPR1_SMP18_0 ((uint32_t)0x01000000)
```

Bit 0

5.173.2.162 ADC_SMPR1_SMP18_1

```
#define ADC_SMPR1_SMP18_1 ((uint32_t)0x02000000)
```

Bit 1

5.173.2.163 ADC_SMPR1_SMP18_2

```
#define ADC_SMPR1_SMP18_2 ((uint32_t)0x04000000)
```

Bit 2

5.173.2.164 ADC_SMPR2_SMP0

```
#define ADC_SMPR2_SMP0 ((uint32_t)0x00000007)
```

SMP0[2:0] bits (Channel 0 Sample time selection)

5.173.2.165 ADC_SMPR2_SMP0_0

```
#define ADC_SMPR2_SMP0_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.166 ADC_SMPR2_SMP0_1

```
#define ADC_SMPR2_SMP0_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.167 ADC_SMPR2_SMP0_2

```
#define ADC_SMPR2_SMP0_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.168 ADC_SMPR2_SMP1

```
#define ADC_SMPR2_SMP1 ((uint32_t)0x00000038)
```

SMP1[2:0] bits (Channel 1 Sample time selection)

5.173.2.169 ADC_SMPR2_SMP1_0

```
#define ADC_SMPR2_SMP1_0 ((uint32_t)0x00000008)
```

Bit 0

5.173.2.170 ADC_SMPR2_SMP1_1

```
#define ADC_SMPR2_SMP1_1 ((uint32_t)0x00000010)
```

Bit 1

5.173.2.171 ADC_SMPR2_SMP1_2

```
#define ADC_SMPR2_SMP1_2 ((uint32_t)0x00000020)
```

Bit 2

5.173.2.172 ADC_SMPR2_SMP2

```
#define ADC_SMPR2_SMP2 ((uint32_t)0x000001C0)
```

SMP2[2:0] bits (Channel 2 Sample time selection)

5.173.2.173 ADC_SMPR2_SMP2_0

```
#define ADC_SMPR2_SMP2_0 ((uint32_t)0x00000040)
```

Bit 0

5.173.2.174 ADC_SMPR2_SMP2_1

```
#define ADC_SMPR2_SMP2_1 ((uint32_t)0x00000080)
```

Bit 1

5.173.2.175 ADC_SMPR2_SMP2_2

```
#define ADC_SMPR2_SMP2_2 ((uint32_t)0x00000100)
```

Bit 2

5.173.2.176 ADC_SMPR2_SMP3

```
#define ADC_SMPR2_SMP3 ((uint32_t)0x00000E00)
```

SMP3[2:0] bits (Channel 3 Sample time selection)

5.173.2.177 ADC_SMPR2_SMP3_0

```
#define ADC_SMPR2_SMP3_0 ((uint32_t)0x00000200)
```

Bit 0

5.173.2.178 ADC_SMPR2_SMP3_1

```
#define ADC_SMPR2_SMP3_1 ((uint32_t)0x00000400)
```

Bit 1

5.173.2.179 ADC_SMPR2_SMP3_2

```
#define ADC_SMPR2_SMP3_2 ((uint32_t)0x00000800)
```

Bit 2

5.173.2.180 ADC_SMPR2_SMP4

```
#define ADC_SMPR2_SMP4 ((uint32_t)0x00007000)
```

SMP4[2:0] bits (Channel 4 Sample time selection)

5.173.2.181 ADC_SMPR2_SMP4_0

```
#define ADC_SMPR2_SMP4_0 ((uint32_t)0x00001000)
```

Bit 0

5.173.2.182 ADC_SMPR2_SMP4_1

```
#define ADC_SMPR2_SMP4_1 ((uint32_t)0x00002000)
```

Bit 1

5.173.2.183 ADC_SMPR2_SMP4_2

```
#define ADC_SMPR2_SMP4_2 ((uint32_t)0x00004000)
```

Bit 2

5.173.2.184 ADC_SMPR2_SMP5

```
#define ADC_SMPR2_SMP5 ((uint32_t)0x00038000)
```

SMP5[2:0] bits (Channel 5 Sample time selection)

5.173.2.185 ADC_SMPR2_SMP5_0

```
#define ADC_SMPR2_SMP5_0 ((uint32_t)0x00008000)
```

Bit 0

5.173.2.186 ADC_SMPR2_SMP5_1

```
#define ADC_SMPR2_SMP5_1 ((uint32_t)0x00010000)
```

Bit 1

5.173.2.187 ADC_SMPR2_SMP5_2

```
#define ADC_SMPR2_SMP5_2 ((uint32_t)0x00020000)
```

Bit 2

5.173.2.188 ADC_SMPR2_SMP6

```
#define ADC_SMPR2_SMP6 ((uint32_t)0x001C0000)
```

SMP6[2:0] bits (Channel 6 Sample time selection)

5.173.2.189 ADC_SMPR2_SMP6_0

```
#define ADC_SMPR2_SMP6_0 ((uint32_t)0x00040000)
```

Bit 0

5.173.2.190 ADC_SMPR2_SMP6_1

```
#define ADC_SMPR2_SMP6_1 ((uint32_t)0x00080000)
```

Bit 1

5.173.2.191 ADC_SMPR2_SMP6_2

```
#define ADC_SMPR2_SMP6_2 ((uint32_t)0x00100000)
```

Bit 2

5.173.2.192 ADC_SMPR2_SMP7

```
#define ADC_SMPR2_SMP7 ((uint32_t)0x00E00000)
```

SMP7[2:0] bits (Channel 7 Sample time selection)

5.173.2.193 ADC_SMPR2_SMP7_0

```
#define ADC_SMPR2_SMP7_0 ((uint32_t)0x00200000)
```

Bit 0

5.173.2.194 ADC_SMPR2_SMP7_1

```
#define ADC_SMPR2_SMP7_1 ((uint32_t)0x00400000)
```

Bit 1

5.173.2.195 ADC_SMPR2_SMP7_2

```
#define ADC_SMPR2_SMP7_2 ((uint32_t)0x00800000)
```

Bit 2

5.173.2.196 ADC_SMPR2_SMP8

```
#define ADC_SMPR2_SMP8 ((uint32_t)0x07000000)
```

SMP8[2:0] bits (Channel 8 Sample time selection)

5.173.2.197 ADC_SMPR2_SMP8_0

```
#define ADC_SMPR2_SMP8_0 ((uint32_t)0x01000000)
```

Bit 0

5.173.2.198 ADC_SMPR2_SMP8_1

```
#define ADC_SMPR2_SMP8_1 ((uint32_t)0x02000000)
```

Bit 1

5.173.2.199 ADC_SMPR2_SMP8_2

```
#define ADC_SMPR2_SMP8_2 ((uint32_t)0x04000000)
```

Bit 2

5.173.2.200 ADC_SMPR2_SMP9

```
#define ADC_SMPR2_SMP9 ((uint32_t)0x38000000)
```

SMP9[2:0] bits (Channel 9 Sample time selection)

5.173.2.201 ADC_SMPR2_SMP9_0

```
#define ADC_SMPR2_SMP9_0 ((uint32_t)0x08000000)
```

Bit 0

5.173.2.202 ADC_SMPR2_SMP9_1

```
#define ADC_SMPR2_SMP9_1 ((uint32_t)0x10000000)
```

Bit 1

5.173.2.203 ADC_SMPR2_SMP9_2

```
#define ADC_SMPR2_SMP9_2 ((uint32_t)0x20000000)
```

Bit 2

5.173.2.204 ADC_SQR1_L

```
#define ADC_SQR1_L ((uint32_t)0x00F00000)
```

L[3:0] bits (Regular channel sequence length)

5.173.2.205 ADC_SQR1_L_0

```
#define ADC_SQR1_L_0 ((uint32_t)0x00100000)
```

Bit 0

5.173.2.206 ADC_SQR1_L_1

```
#define ADC_SQR1_L_1 ((uint32_t)0x00200000)
```

Bit 1

5.173.2.207 ADC_SQR1_L_2

```
#define ADC_SQR1_L_2 ((uint32_t)0x00400000)
```

Bit 2

5.173.2.208 ADC_SQR1_L_3

```
#define ADC_SQR1_L_3 ((uint32_t)0x00800000)
```

Bit 3

5.173.2.209 ADC_SQR1_SQ13

```
#define ADC_SQR1_SQ13 ((uint32_t)0x0000001F)
```

SQ13[4:0] bits (13th conversion in regular sequence)

5.173.2.210 ADC_SQR1_SQ13_0

```
#define ADC_SQR1_SQ13_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.211 ADC_SQR1_SQ13_1

```
#define ADC_SQR1_SQ13_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.212 ADC_SQR1_SQ13_2

```
#define ADC_SQR1_SQ13_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.213 ADC_SQR1_SQ13_3

```
#define ADC_SQR1_SQ13_3 ((uint32_t)0x00000008)
```

Bit 3

5.173.2.214 ADC_SQR1_SQ13_4

```
#define ADC_SQR1_SQ13_4 ((uint32_t)0x00000010)
```

Bit 4

5.173.2.215 ADC_SQR1_SQ14

```
#define ADC_SQR1_SQ14 ((uint32_t)0x000003E0)
```

SQ14[4:0] bits (14th conversion in regular sequence)

5.173.2.216 ADC_SQR1_SQ14_0

```
#define ADC_SQR1_SQ14_0 ((uint32_t)0x00000020)
```

Bit 0

5.173.2.217 ADC_SQR1_SQ14_1

```
#define ADC_SQR1_SQ14_1 ((uint32_t)0x00000040)
```

Bit 1

5.173.2.218 ADC_SQR1_SQ14_2

```
#define ADC_SQR1_SQ14_2 ((uint32_t)0x00000080)
```

Bit 2

5.173.2.219 ADC_SQR1_SQ14_3

```
#define ADC_SQR1_SQ14_3 ((uint32_t)0x00000100)
```

Bit 3

5.173.2.220 ADC_SQR1_SQ14_4

```
#define ADC_SQR1_SQ14_4 ((uint32_t)0x00000200)
```

Bit 4

5.173.2.221 ADC_SQR1_SQ15

```
#define ADC_SQR1_SQ15 ((uint32_t)0x00007C00)
```

SQ15[4:0] bits (15th conversion in regular sequence)

5.173.2.222 ADC_SQR1_SQ15_0

```
#define ADC_SQR1_SQ15_0 ((uint32_t)0x00000400)
```

Bit 0

5.173.2.223 ADC_SQR1_SQ15_1

```
#define ADC_SQR1_SQ15_1 ((uint32_t)0x00000800)
```

Bit 1

5.173.2.224 ADC_SQR1_SQ15_2

```
#define ADC_SQR1_SQ15_2 ((uint32_t)0x00001000)
```

Bit 2

5.173.2.225 ADC_SQR1_SQ15_3

```
#define ADC_SQR1_SQ15_3 ((uint32_t)0x00002000)
```

Bit 3

5.173.2.226 ADC_SQR1_SQ15_4

```
#define ADC_SQR1_SQ15_4 ((uint32_t)0x00004000)
```

Bit 4

5.173.2.227 ADC_SQR1_SQ16

```
#define ADC_SQR1_SQ16 ((uint32_t)0x000F8000)
```

SQ16[4:0] bits (16th conversion in regular sequence)

5.173.2.228 ADC_SQR1_SQ16_0

```
#define ADC_SQR1_SQ16_0 ((uint32_t)0x00008000)
```

Bit 0

5.173.2.229 ADC_SQR1_SQ16_1

```
#define ADC_SQR1_SQ16_1 ((uint32_t)0x00010000)
```

Bit 1

5.173.2.230 ADC_SQR1_SQ16_2

```
#define ADC_SQR1_SQ16_2 ((uint32_t)0x00020000)
```

Bit 2

5.173.2.231 ADC_SQR1_SQ16_3

```
#define ADC_SQR1_SQ16_3 ((uint32_t)0x00040000)
```

Bit 3

5.173.2.232 ADC_SQR1_SQ16_4

```
#define ADC_SQR1_SQ16_4 ((uint32_t)0x00080000)
```

Bit 4

5.173.2.233 ADC_SQR2_SQ10

```
#define ADC_SQR2_SQ10 ((uint32_t)0x000F8000)
```

SQ10[4:0] bits (10th conversion in regular sequence)

5.173.2.234 ADC_SQR2_SQ10_0

```
#define ADC_SQR2_SQ10_0 ((uint32_t)0x00008000)
```

Bit 0

5.173.2.235 ADC_SQR2_SQ10_1

```
#define ADC_SQR2_SQ10_1 ((uint32_t)0x00010000)
```

Bit 1

5.173.2.236 ADC_SQR2_SQ10_2

```
#define ADC_SQR2_SQ10_2 ((uint32_t)0x00020000)
```

Bit 2

5.173.2.237 ADC_SQR2_SQ10_3

```
#define ADC_SQR2_SQ10_3 ((uint32_t)0x00040000)
```

Bit 3

5.173.2.238 ADC_SQR2_SQ10_4

```
#define ADC_SQR2_SQ10_4 ((uint32_t)0x00080000)
```

Bit 4

5.173.2.239 ADC_SQR2_SQ11

```
#define ADC_SQR2_SQ11 ((uint32_t)0x01F00000)
```

SQ11[4:0] bits (11th conversion in regular sequence)

5.173.2.240 ADC_SQR2_SQ11_0

```
#define ADC_SQR2_SQ11_0 ((uint32_t)0x00100000)
```

Bit 0

5.173.2.241 ADC_SQR2_SQ11_1

```
#define ADC_SQR2_SQ11_1 ((uint32_t)0x00200000)
```

Bit 1

5.173.2.242 ADC_SQR2_SQ11_2

```
#define ADC_SQR2_SQ11_2 ((uint32_t)0x00400000)
```

Bit 2

5.173.2.243 ADC_SQR2_SQ11_3

```
#define ADC_SQR2_SQ11_3 ((uint32_t)0x00800000)
```

Bit 3

5.173.2.244 ADC_SQR2_SQ11_4

```
#define ADC_SQR2_SQ11_4 ((uint32_t)0x01000000)
```

Bit 4

5.173.2.245 ADC_SQR2_SQ12

```
#define ADC_SQR2_SQ12 ((uint32_t)0x3E000000)
```

SQ12[4:0] bits (12th conversion in regular sequence)

5.173.2.246 ADC_SQR2_SQ12_0

```
#define ADC_SQR2_SQ12_0 ((uint32_t)0x02000000)
```

Bit 0

5.173.2.247 ADC_SQR2_SQ12_1

```
#define ADC_SQR2_SQ12_1 ((uint32_t)0x04000000)
```

Bit 1

5.173.2.248 ADC_SQR2_SQ12_2

```
#define ADC_SQR2_SQ12_2 ((uint32_t)0x08000000)
```

Bit 2

5.173.2.249 ADC_SQR2_SQ12_3

```
#define ADC_SQR2_SQ12_3 ((uint32_t)0x10000000)
```

Bit 3

5.173.2.250 ADC_SQR2_SQ12_4

```
#define ADC_SQR2_SQ12_4 ((uint32_t)0x20000000)
```

Bit 4

5.173.2.251 ADC_SQR2_SQ7

```
#define ADC_SQR2_SQ7 ((uint32_t)0x0000001F)
```

SQ7[4:0] bits (7th conversion in regular sequence)

5.173.2.252 ADC_SQR2_SQ7_0

```
#define ADC_SQR2_SQ7_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.253 ADC_SQR2_SQ7_1

```
#define ADC_SQR2_SQ7_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.254 ADC_SQR2_SQ7_2

```
#define ADC_SQR2_SQ7_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.255 ADC_SQR2_SQ7_3

```
#define ADC_SQR2_SQ7_3 ((uint32_t)0x00000008)
```

Bit 3

5.173.2.256 ADC_SQR2_SQ7_4

```
#define ADC_SQR2_SQ7_4 ((uint32_t)0x00000010)
```

Bit 4

5.173.2.257 ADC_SQR2_SQ8

```
#define ADC_SQR2_SQ8 ((uint32_t)0x000003E0)
```

SQ8[4:0] bits (8th conversion in regular sequence)

5.173.2.258 ADC_SQR2_SQ8_0

```
#define ADC_SQR2_SQ8_0 ((uint32_t)0x00000020)
```

Bit 0

5.173.2.259 ADC_SQR2_SQ8_1

```
#define ADC_SQR2_SQ8_1 ((uint32_t)0x00000040)
```

Bit 1

5.173.2.260 ADC_SQR2_SQ8_2

```
#define ADC_SQR2_SQ8_2 ((uint32_t)0x00000080)
```

Bit 2

5.173.2.261 ADC_SQR2_SQ8_3

```
#define ADC_SQR2_SQ8_3 ((uint32_t)0x00000100)
```

Bit 3

5.173.2.262 ADC_SQR2_SQ8_4

```
#define ADC_SQR2_SQ8_4 ((uint32_t)0x00000200)
```

Bit 4

5.173.2.263 ADC_SQR2_SQ9

```
#define ADC_SQR2_SQ9 ((uint32_t)0x00007C00)
```

SQ9[4:0] bits (9th conversion in regular sequence)

5.173.2.264 ADC_SQR2_SQ9_0

```
#define ADC_SQR2_SQ9_0 ((uint32_t)0x00000400)
```

Bit 0

5.173.2.265 ADC_SQR2_SQ9_1

```
#define ADC_SQR2_SQ9_1 ((uint32_t)0x00000800)
```

Bit 1

5.173.2.266 ADC_SQR2_SQ9_2

```
#define ADC_SQR2_SQ9_2 ((uint32_t)0x00001000)
```

Bit 2

5.173.2.267 ADC_SQR2_SQ9_3

```
#define ADC_SQR2_SQ9_3 ((uint32_t)0x00002000)
```

Bit 3

5.173.2.268 ADC_SQR2_SQ9_4

```
#define ADC_SQR2_SQ9_4 ((uint32_t)0x00004000)
```

Bit 4

5.173.2.269 ADC_SQR3_SQ1

```
#define ADC_SQR3_SQ1 ((uint32_t)0x0000001F)
```

SQ1[4:0] bits (1st conversion in regular sequence)

5.173.2.270 ADC_SQR3_SQ1_0

```
#define ADC_SQR3_SQ1_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.271 ADC_SQR3_SQ1_1

```
#define ADC_SQR3_SQ1_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.272 ADC_SQR3_SQ1_2

```
#define ADC_SQR3_SQ1_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.273 ADC_SQR3_SQ1_3

```
#define ADC_SQR3_SQ1_3 ((uint32_t)0x00000008)
```

Bit 3

5.173.2.274 ADC_SQR3_SQ1_4

```
#define ADC_SQR3_SQ1_4 ((uint32_t)0x00000010)
```

Bit 4

5.173.2.275 ADC_SQR3_SQ2

```
#define ADC_SQR3_SQ2 ((uint32_t)0x000003E0)
```

SQ2[4:0] bits (2nd conversion in regular sequence)

5.173.2.276 ADC_SQR3_SQ2_0

```
#define ADC_SQR3_SQ2_0 ((uint32_t)0x00000020)
```

Bit 0

5.173.2.277 ADC_SQR3_SQ2_1

```
#define ADC_SQR3_SQ2_1 ((uint32_t)0x00000040)
```

Bit 1

5.173.2.278 ADC_SQR3_SQ2_2

```
#define ADC_SQR3_SQ2_2 ((uint32_t)0x00000080)
```

Bit 2

5.173.2.279 ADC_SQR3_SQ2_3

```
#define ADC_SQR3_SQ2_3 ((uint32_t)0x00000100)
```

Bit 3

5.173.2.280 ADC_SQR3_SQ2_4

```
#define ADC_SQR3_SQ2_4 ((uint32_t)0x00000200)
```

Bit 4

5.173.2.281 ADC_SQR3_SQ3

```
#define ADC_SQR3_SQ3 ((uint32_t)0x00007C00)
```

SQ3[4:0] bits (3rd conversion in regular sequence)

5.173.2.282 ADC_SQR3_SQ3_0

```
#define ADC_SQR3_SQ3_0 ((uint32_t)0x00000400)
```

Bit 0

5.173.2.283 ADC_SQR3_SQ3_1

```
#define ADC_SQR3_SQ3_1 ((uint32_t)0x00000800)
```

Bit 1

5.173.2.284 ADC_SQR3_SQ3_2

```
#define ADC_SQR3_SQ3_2 ((uint32_t)0x00001000)
```

Bit 2

5.173.2.285 ADC_SQR3_SQ3_3

```
#define ADC_SQR3_SQ3_3 ((uint32_t)0x00002000)
```

Bit 3

5.173.2.286 ADC_SQR3_SQ3_4

```
#define ADC_SQR3_SQ3_4 ((uint32_t)0x00004000)
```

Bit 4

5.173.2.287 ADC_SQR3_SQ4

```
#define ADC_SQR3_SQ4 ((uint32_t)0x000F8000)
```

SQ4[4:0] bits (4th conversion in regular sequence)

5.173.2.288 ADC_SQR3_SQ4_0

```
#define ADC_SQR3_SQ4_0 ((uint32_t)0x00008000)
```

Bit 0

5.173.2.289 ADC_SQR3_SQ4_1

```
#define ADC_SQR3_SQ4_1 ((uint32_t)0x00010000)
```

Bit 1

5.173.2.290 ADC_SQR3_SQ4_2

```
#define ADC_SQR3_SQ4_2 ((uint32_t)0x00020000)
```

Bit 2

5.173.2.291 ADC_SQR3_SQ4_3

```
#define ADC_SQR3_SQ4_3 ((uint32_t)0x00040000)
```

Bit 3

5.173.2.292 ADC_SQR3_SQ4_4

```
#define ADC_SQR3_SQ4_4 ((uint32_t)0x00080000)
```

Bit 4

5.173.2.293 ADC_SQR3_SQ5

```
#define ADC_SQR3_SQ5 ((uint32_t)0x01F00000)
```

SQ5[4:0] bits (5th conversion in regular sequence)

5.173.2.294 ADC_SQR3_SQ5_0

```
#define ADC_SQR3_SQ5_0 ((uint32_t)0x00100000)
```

Bit 0

5.173.2.295 ADC_SQR3_SQ5_1

```
#define ADC_SQR3_SQ5_1 ((uint32_t)0x00200000)
```

Bit 1

5.173.2.296 ADC_SQR3_SQ5_2

```
#define ADC_SQR3_SQ5_2 ((uint32_t)0x00400000)
```

Bit 2

5.173.2.297 ADC_SQR3_SQ5_3

```
#define ADC_SQR3_SQ5_3 ((uint32_t)0x00800000)
```

Bit 3

5.173.2.298 ADC_SQR3_SQ5_4

```
#define ADC_SQR3_SQ5_4 ((uint32_t)0x01000000)
```

Bit 4

5.173.2.299 ADC_SQR3_SQ6

```
#define ADC_SQR3_SQ6 ((uint32_t)0x3E000000)
```

SQ6[4:0] bits (6th conversion in regular sequence)

5.173.2.300 ADC_SQR3_SQ6_0

```
#define ADC_SQR3_SQ6_0 ((uint32_t)0x02000000)
```

Bit 0

5.173.2.301 ADC_SQR3_SQ6_1

```
#define ADC_SQR3_SQ6_1 ((uint32_t)0x04000000)
```

Bit 1

5.173.2.302 ADC_SQR3_SQ6_2

```
#define ADC_SQR3_SQ6_2 ((uint32_t)0x08000000)
```

Bit 2

5.173.2.303 ADC_SQR3_SQ6_3

```
#define ADC_SQR3_SQ6_3 ((uint32_t)0x10000000)
```

Bit 3

5.173.2.304 ADC_SQR3_SQ6_4

```
#define ADC_SQR3_SQ6_4 ((uint32_t)0x20000000)
```

Bit 4

5.173.2.305 ADC_SR_AWD

```
#define ADC_SR_AWD ((uint8_t)0x01)
```

Analog watchdog flag

5.173.2.306 ADC_SR_EOC

```
#define ADC_SR_EOC ((uint8_t)0x02)
```

End of conversion

5.173.2.307 ADC_SR_JEOC

```
#define ADC_SR_JEOC ((uint8_t)0x04)
```

Injected channel end of conversion

5.173.2.308 ADC_SR_JSTRT

```
#define ADC_SR_JSTRT ((uint8_t)0x08)
```

Injected channel Start flag

5.173.2.309 ADC_SR_OVR

```
#define ADC_SR_OVR ((uint8_t)0x20)
```

Overrun flag

5.173.2.310 ADC_SR_STRT

```
#define ADC_SR_STRT ((uint8_t)0x10)
```

Regular channel Start flag

5.173.2.311 CAN_BTR_BRP

```
#define CAN_BTR_BRP ((uint32_t)0x000003FF)
```

Baud Rate Prescaler

5.173.2.312 CAN_BTR_LBKM

```
#define CAN_BTR_LBKM ((uint32_t)0x40000000)
```

Loop Back Mode (Debug)

5.173.2.313 CAN_BTR_SILM

```
#define CAN_BTR_SILM ((uint32_t)0x80000000)
```

Silent Mode Mailbox registers

5.173.2.314 CAN_BTR_SJW

```
#define CAN_BTR_SJW ((uint32_t)0x03000000)
```

Resynchronization Jump Width

5.173.2.315 CAN_BTR_TS1

```
#define CAN_BTR_TS1 ((uint32_t)0x000F0000)
```

Time Segment 1

5.173.2.316 CAN_BTR_TS2

```
#define CAN_BTR_TS2 ((uint32_t)0x00700000)
```

Time Segment 2

5.173.2.317 CAN_ESR_BOFF

```
#define CAN_ESR_BOFF ((uint32_t)0x00000004)
```

Bus-Off Flag

5.173.2.318 CAN_ESR_EPVF

```
#define CAN_ESR_EPVF ((uint32_t)0x00000002)
```

Error Passive Flag

5.173.2.319 CAN_ESR_EWGF

```
#define CAN_ESR_EWGF ((uint32_t)0x00000001)
```

Error Warning Flag

5.173.2.320 CAN_ESR_LEC

```
#define CAN_ESR_LEC ((uint32_t)0x00000070)
```

LEC[2:0] bits (Last Error Code)

5.173.2.321 CAN_ESR_LEC_0

```
#define CAN_ESR_LEC_0 ((uint32_t)0x00000010)
```

Bit 0

5.173.2.322 CAN_ESR_LEC_1

```
#define CAN_ESR_LEC_1 ((uint32_t)0x00000020)
```

Bit 1

5.173.2.323 CAN_ESR_LEC_2

```
#define CAN_ESR_LEC_2 ((uint32_t)0x00000040)
```

Bit 2

5.173.2.324 CAN_ESR_REC

```
#define CAN_ESR_REC ((uint32_t)0xFF000000)
```

Receive Error Counter

5.173.2.325 CAN_ESR_TEC

```
#define CAN_ESR_TEC ((uint32_t)0x00FF0000)
```

Least significant byte of the 9-bit Transmit Error Counter

5.173.2.326 CAN_F0R1_FB0

```
#define CAN_F0R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.327 CAN_F0R1_FB1

```
#define CAN_F0R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.328 CAN_F0R1_FB10

```
#define CAN_F0R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.329 CAN_F0R1_FB11

```
#define CAN_F0R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.330 CAN_F0R1_FB12

```
#define CAN_F0R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.331 CAN_F0R1_FB13

```
#define CAN_F0R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.332 CAN_F0R1_FB14

```
#define CAN_F0R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.333 CAN_F0R1_FB15

```
#define CAN_F0R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.334 CAN_F0R1_FB16

```
#define CAN_F0R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.335 CAN_F0R1_FB17

```
#define CAN_F0R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.336 CAN_F0R1_FB18

```
#define CAN_F0R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.337 CAN_F0R1_FB19

```
#define CAN_F0R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.338 CAN_F0R1_FB2

```
#define CAN_F0R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.339 CAN_F0R1_FB20

```
#define CAN_F0R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.340 CAN_F0R1_FB21

```
#define CAN_F0R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.341 CAN_F0R1_FB22

```
#define CAN_F0R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.342 CAN_F0R1_FB23

```
#define CAN_F0R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.343 CAN_F0R1_FB24

```
#define CAN_F0R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.344 CAN_F0R1_FB25

```
#define CAN_F0R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.345 CAN_F0R1_FB26

```
#define CAN_F0R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.346 CAN_F0R1_FB27

```
#define CAN_F0R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.347 CAN_F0R1_FB28

```
#define CAN_F0R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.348 CAN_F0R1_FB29

```
#define CAN_F0R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.349 CAN_F0R1_FB3

```
#define CAN_F0R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.350 CAN_F0R1_FB30

```
#define CAN_F0R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.351 CAN_F0R1_FB31

```
#define CAN_F0R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.352 CAN_F0R1_FB4

```
#define CAN_F0R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.353 CAN_F0R1_FB5

```
#define CAN_F0R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.354 CAN_F0R1_FB6

```
#define CAN_F0R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.355 CAN_F0R1_FB7

```
#define CAN_F0R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.356 CAN_F0R1_FB8

```
#define CAN_F0R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.357 CAN_F0R1_FB9

```
#define CAN_F0R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.358 CAN_F0R2_FB0

```
#define CAN_F0R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.359 CAN_F0R2_FB1

```
#define CAN_F0R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.360 CAN_F0R2_FB10

```
#define CAN_F0R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.361 CAN_F0R2_FB11

```
#define CAN_F0R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.362 CAN_F0R2_FB12

```
#define CAN_F0R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.363 CAN_F0R2_FB13

```
#define CAN_F0R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.364 CAN_F0R2_FB14

```
#define CAN_F0R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.365 CAN_F0R2_FB15

```
#define CAN_F0R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.366 CAN_F0R2_FB16

```
#define CAN_F0R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.367 CAN_F0R2_FB17

```
#define CAN_F0R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.368 CAN_F0R2_FB18

```
#define CAN_F0R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.369 CAN_F0R2_FB19

```
#define CAN_F0R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.370 CAN_F0R2_FB2

```
#define CAN_F0R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.371 CAN_F0R2_FB20

```
#define CAN_F0R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.372 CAN_F0R2_FB21

```
#define CAN_F0R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.373 CAN_F0R2_FB22

```
#define CAN_F0R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.374 CAN_F0R2_FB23

```
#define CAN_F0R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.375 CAN_F0R2_FB24

```
#define CAN_F0R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.376 CAN_F0R2_FB25

```
#define CAN_F0R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.377 CAN_F0R2_FB26

```
#define CAN_F0R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.378 CAN_F0R2_FB27

```
#define CAN_F0R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.379 CAN_F0R2_FB28

```
#define CAN_F0R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.380 CAN_F0R2_FB29

```
#define CAN_F0R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.381 CAN_F0R2_FB3

```
#define CAN_F0R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.382 CAN_F0R2_FB30

```
#define CAN_F0R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.383 CAN_F0R2_FB31

```
#define CAN_F0R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.384 CAN_F0R2_FB4

```
#define CAN_F0R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.385 CAN_F0R2_FB5

```
#define CAN_F0R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.386 CAN_F0R2_FB6

```
#define CAN_F0R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.387 CAN_F0R2_FB7

```
#define CAN_F0R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.388 CAN_F0R2_FB8

```
#define CAN_F0R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.389 CAN_F0R2_FB9

```
#define CAN_F0R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.390 CAN_F10R1_FB0

```
#define CAN_F10R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.391 CAN_F10R1_FB1

```
#define CAN_F10R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.392 CAN_F10R1_FB10

```
#define CAN_F10R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.393 CAN_F10R1_FB11

```
#define CAN_F10R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.394 CAN_F10R1_FB12

```
#define CAN_F10R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.395 CAN_F10R1_FB13

```
#define CAN_F10R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.396 CAN_F10R1_FB14

```
#define CAN_F10R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.397 CAN_F10R1_FB15

```
#define CAN_F10R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.398 CAN_F10R1_FB16

```
#define CAN_F10R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.399 CAN_F10R1_FB17

```
#define CAN_F10R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.400 CAN_F10R1_FB18

```
#define CAN_F10R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.401 CAN_F10R1_FB19

```
#define CAN_F10R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.402 CAN_F10R1_FB2

```
#define CAN_F10R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.403 CAN_F10R1_FB20

```
#define CAN_F10R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.404 CAN_F10R1_FB21

```
#define CAN_F10R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.405 CAN_F10R1_FB22

```
#define CAN_F10R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.406 CAN_F10R1_FB23

```
#define CAN_F10R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.407 CAN_F10R1_FB24

```
#define CAN_F10R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.408 CAN_F10R1_FB25

```
#define CAN_F10R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.409 CAN_F10R1_FB26

```
#define CAN_F10R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.410 CAN_F10R1_FB27

```
#define CAN_F10R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.411 CAN_F10R1_FB28

```
#define CAN_F10R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.412 CAN_F10R1_FB29

```
#define CAN_F10R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.413 CAN_F10R1_FB3

```
#define CAN_F10R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.414 CAN_F10R1_FB30

```
#define CAN_F10R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.415 CAN_F10R1_FB31

```
#define CAN_F10R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.416 CAN_F10R1_FB4

```
#define CAN_F10R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.417 CAN_F10R1_FB5

```
#define CAN_F10R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.418 CAN_F10R1_FB6

```
#define CAN_F10R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.419 CAN_F10R1_FB7

```
#define CAN_F10R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.420 CAN_F10R1_FB8

```
#define CAN_F10R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.421 CAN_F10R1_FB9

```
#define CAN_F10R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.422 CAN_F10R2_FB0

```
#define CAN_F10R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.423 CAN_F10R2_FB1

```
#define CAN_F10R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.424 CAN_F10R2_FB10

```
#define CAN_F10R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.425 CAN_F10R2_FB11

```
#define CAN_F10R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.426 CAN_F10R2_FB12

```
#define CAN_F10R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.427 CAN_F10R2_FB13

```
#define CAN_F10R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.428 CAN_F10R2_FB14

```
#define CAN_F10R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.429 CAN_F10R2_FB15

```
#define CAN_F10R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.430 CAN_F10R2_FB16

```
#define CAN_F10R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.431 CAN_F10R2_FB17

```
#define CAN_F10R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.432 CAN_F10R2_FB18

```
#define CAN_F10R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.433 CAN_F10R2_FB19

```
#define CAN_F10R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.434 CAN_F10R2_FB2

```
#define CAN_F10R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.435 CAN_F10R2_FB20

```
#define CAN_F10R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.436 CAN_F10R2_FB21

```
#define CAN_F10R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.437 CAN_F10R2_FB22

```
#define CAN_F10R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.438 CAN_F10R2_FB23

```
#define CAN_F10R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.439 CAN_F10R2_FB24

```
#define CAN_F10R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.440 CAN_F10R2_FB25

```
#define CAN_F10R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.441 CAN_F10R2_FB26

```
#define CAN_F10R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.442 CAN_F10R2_FB27

```
#define CAN_F10R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.443 CAN_F10R2_FB28

```
#define CAN_F10R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.444 CAN_F10R2_FB29

```
#define CAN_F10R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.445 CAN_F10R2_FB3

```
#define CAN_F10R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.446 CAN_F10R2_FB30

```
#define CAN_F10R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.447 CAN_F10R2_FB31

```
#define CAN_F10R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.448 CAN_F10R2_FB4

```
#define CAN_F10R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.449 CAN_F10R2_FB5

```
#define CAN_F10R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.450 CAN_F10R2_FB6

```
#define CAN_F10R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.451 CAN_F10R2_FB7

```
#define CAN_F10R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.452 CAN_F10R2_FB8

```
#define CAN_F10R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.453 CAN_F10R2_FB9

```
#define CAN_F10R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.454 CAN_F11R1_FB0

```
#define CAN_F11R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.455 CAN_F11R1_FB1

```
#define CAN_F11R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.456 CAN_F11R1_FB10

```
#define CAN_F11R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.457 CAN_F11R1_FB11

```
#define CAN_F11R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.458 CAN_F11R1_FB12

```
#define CAN_F11R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.459 CAN_F11R1_FB13

```
#define CAN_F11R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.460 CAN_F11R1_FB14

```
#define CAN_F11R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.461 CAN_F11R1_FB15

```
#define CAN_F11R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.462 CAN_F11R1_FB16

```
#define CAN_F11R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.463 CAN_F11R1_FB17

```
#define CAN_F11R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.464 CAN_F11R1_FB18

```
#define CAN_F11R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.465 CAN_F11R1_FB19

```
#define CAN_F11R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.466 CAN_F11R1_FB2

```
#define CAN_F11R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.467 CAN_F11R1_FB20

```
#define CAN_F11R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.468 CAN_F11R1_FB21

```
#define CAN_F11R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.469 CAN_F11R1_FB22

```
#define CAN_F11R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.470 CAN_F11R1_FB23

```
#define CAN_F11R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.471 CAN_F11R1_FB24

```
#define CAN_F11R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.472 CAN_F11R1_FB25

```
#define CAN_F11R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.473 CAN_F11R1_FB26

```
#define CAN_F11R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.474 CAN_F11R1_FB27

```
#define CAN_F11R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.475 CAN_F11R1_FB28

```
#define CAN_F11R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.476 CAN_F11R1_FB29

```
#define CAN_F11R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.477 CAN_F11R1_FB3

```
#define CAN_F11R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.478 CAN_F11R1_FB30

```
#define CAN_F11R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.479 CAN_F11R1_FB31

```
#define CAN_F11R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.480 CAN_F11R1_FB4

```
#define CAN_F11R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.481 CAN_F11R1_FB5

```
#define CAN_F11R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.482 CAN_F11R1_FB6

```
#define CAN_F11R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.483 CAN_F11R1_FB7

```
#define CAN_F11R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.484 CAN_F11R1_FB8

```
#define CAN_F11R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.485 CAN_F11R1_FB9

```
#define CAN_F11R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.486 CAN_F11R2_FB0

```
#define CAN_F11R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.487 CAN_F11R2_FB1

```
#define CAN_F11R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.488 CAN_F11R2_FB10

```
#define CAN_F11R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.489 CAN_F11R2_FB11

```
#define CAN_F11R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.490 CAN_F11R2_FB12

```
#define CAN_F11R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.491 CAN_F11R2_FB13

```
#define CAN_F11R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.492 CAN_F11R2_FB14

```
#define CAN_F11R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.493 CAN_F11R2_FB15

```
#define CAN_F11R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.494 CAN_F11R2_FB16

```
#define CAN_F11R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.495 CAN_F11R2_FB17

```
#define CAN_F11R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.496 CAN_F11R2_FB18

```
#define CAN_F11R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.497 CAN_F11R2_FB19

```
#define CAN_F11R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.498 CAN_F11R2_FB2

```
#define CAN_F11R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.499 CAN_F11R2_FB20

```
#define CAN_F11R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.500 CAN_F11R2_FB21

```
#define CAN_F11R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.501 CAN_F11R2_FB22

```
#define CAN_F11R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.502 CAN_F11R2_FB23

```
#define CAN_F11R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.503 CAN_F11R2_FB24

```
#define CAN_F11R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.504 CAN_F11R2_FB25

```
#define CAN_F11R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.505 CAN_F11R2_FB26

```
#define CAN_F11R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.506 CAN_F11R2_FB27

```
#define CAN_F11R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.507 CAN_F11R2_FB28

```
#define CAN_F11R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.508 CAN_F11R2_FB29

```
#define CAN_F11R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.509 CAN_F11R2_FB3

```
#define CAN_F11R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.510 CAN_F11R2_FB30

```
#define CAN_F11R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.511 CAN_F11R2_FB31

```
#define CAN_F11R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.512 CAN_F11R2_FB4

```
#define CAN_F11R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.513 CAN_F11R2_FB5

```
#define CAN_F11R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.514 CAN_F11R2_FB6

```
#define CAN_F11R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.515 CAN_F11R2_FB7

```
#define CAN_F11R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.516 CAN_F11R2_FB8

```
#define CAN_F11R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.517 CAN_F11R2_FB9

```
#define CAN_F11R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.518 CAN_F12R1_FB0

```
#define CAN_F12R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.519 CAN_F12R1_FB1

```
#define CAN_F12R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.520 CAN_F12R1_FB10

```
#define CAN_F12R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.521 CAN_F12R1_FB11

```
#define CAN_F12R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.522 CAN_F12R1_FB12

```
#define CAN_F12R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.523 CAN_F12R1_FB13

```
#define CAN_F12R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.524 CAN_F12R1_FB14

```
#define CAN_F12R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.525 CAN_F12R1_FB15

```
#define CAN_F12R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.526 CAN_F12R1_FB16

```
#define CAN_F12R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.527 CAN_F12R1_FB17

```
#define CAN_F12R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.528 CAN_F12R1_FB18

```
#define CAN_F12R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.529 CAN_F12R1_FB19

```
#define CAN_F12R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.530 CAN_F12R1_FB2

```
#define CAN_F12R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.531 CAN_F12R1_FB20

```
#define CAN_F12R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.532 CAN_F12R1_FB21

```
#define CAN_F12R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.533 CAN_F12R1_FB22

```
#define CAN_F12R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.534 CAN_F12R1_FB23

```
#define CAN_F12R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.535 CAN_F12R1_FB24

```
#define CAN_F12R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.536 CAN_F12R1_FB25

```
#define CAN_F12R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.537 CAN_F12R1_FB26

```
#define CAN_F12R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.538 CAN_F12R1_FB27

```
#define CAN_F12R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.539 CAN_F12R1_FB28

```
#define CAN_F12R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.540 CAN_F12R1_FB29

```
#define CAN_F12R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.541 CAN_F12R1_FB3

```
#define CAN_F12R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.542 CAN_F12R1_FB30

```
#define CAN_F12R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.543 CAN_F12R1_FB31

```
#define CAN_F12R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.544 CAN_F12R1_FB4

```
#define CAN_F12R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.545 CAN_F12R1_FB5

```
#define CAN_F12R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.546 CAN_F12R1_FB6

```
#define CAN_F12R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.547 CAN_F12R1_FB7

```
#define CAN_F12R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.548 CAN_F12R1_FB8

```
#define CAN_F12R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.549 CAN_F12R1_FB9

```
#define CAN_F12R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.550 CAN_F12R2_FB0

```
#define CAN_F12R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.551 CAN_F12R2_FB1

```
#define CAN_F12R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.552 CAN_F12R2_FB10

```
#define CAN_F12R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.553 CAN_F12R2_FB11

```
#define CAN_F12R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.554 CAN_F12R2_FB12

```
#define CAN_F12R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.555 CAN_F12R2_FB13

```
#define CAN_F12R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.556 CAN_F12R2_FB14

```
#define CAN_F12R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.557 CAN_F12R2_FB15

```
#define CAN_F12R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.558 CAN_F12R2_FB16

```
#define CAN_F12R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.559 CAN_F12R2_FB17

```
#define CAN_F12R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.560 CAN_F12R2_FB18

```
#define CAN_F12R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.561 CAN_F12R2_FB19

```
#define CAN_F12R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.562 CAN_F12R2_FB2

```
#define CAN_F12R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.563 CAN_F12R2_FB20

```
#define CAN_F12R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.564 CAN_F12R2_FB21

```
#define CAN_F12R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.565 CAN_F12R2_FB22

```
#define CAN_F12R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.566 CAN_F12R2_FB23

```
#define CAN_F12R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.567 CAN_F12R2_FB24

```
#define CAN_F12R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.568 CAN_F12R2_FB25

```
#define CAN_F12R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.569 CAN_F12R2_FB26

```
#define CAN_F12R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.570 CAN_F12R2_FB27

```
#define CAN_F12R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.571 CAN_F12R2_FB28

```
#define CAN_F12R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.572 CAN_F12R2_FB29

```
#define CAN_F12R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.573 CAN_F12R2_FB3

```
#define CAN_F12R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.574 CAN_F12R2_FB30

```
#define CAN_F12R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.575 CAN_F12R2_FB31

```
#define CAN_F12R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.576 CAN_F12R2_FB4

```
#define CAN_F12R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.577 CAN_F12R2_FB5

```
#define CAN_F12R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.578 CAN_F12R2_FB6

```
#define CAN_F12R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.579 CAN_F12R2_FB7

```
#define CAN_F12R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.580 CAN_F12R2_FB8

```
#define CAN_F12R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.581 CAN_F12R2_FB9

```
#define CAN_F12R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.582 CAN_F13R1_FB0

```
#define CAN_F13R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.583 CAN_F13R1_FB1

```
#define CAN_F13R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.584 CAN_F13R1_FB10

```
#define CAN_F13R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.585 CAN_F13R1_FB11

```
#define CAN_F13R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.586 CAN_F13R1_FB12

```
#define CAN_F13R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.587 CAN_F13R1_FB13

```
#define CAN_F13R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.588 CAN_F13R1_FB14

```
#define CAN_F13R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.589 CAN_F13R1_FB15

```
#define CAN_F13R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.590 CAN_F13R1_FB16

```
#define CAN_F13R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.591 CAN_F13R1_FB17

```
#define CAN_F13R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.592 CAN_F13R1_FB18

```
#define CAN_F13R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.593 CAN_F13R1_FB19

```
#define CAN_F13R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.594 CAN_F13R1_FB2

```
#define CAN_F13R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.595 CAN_F13R1_FB20

```
#define CAN_F13R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.596 CAN_F13R1_FB21

```
#define CAN_F13R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.597 CAN_F13R1_FB22

```
#define CAN_F13R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.598 CAN_F13R1_FB23

```
#define CAN_F13R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.599 CAN_F13R1_FB24

```
#define CAN_F13R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.600 CAN_F13R1_FB25

```
#define CAN_F13R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.601 CAN_F13R1_FB26

```
#define CAN_F13R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.602 CAN_F13R1_FB27

```
#define CAN_F13R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.603 CAN_F13R1_FB28

```
#define CAN_F13R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.604 CAN_F13R1_FB29

```
#define CAN_F13R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.605 CAN_F13R1_FB3

```
#define CAN_F13R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.606 CAN_F13R1_FB30

```
#define CAN_F13R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.607 CAN_F13R1_FB31

```
#define CAN_F13R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.608 CAN_F13R1_FB4

```
#define CAN_F13R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.609 CAN_F13R1_FB5

```
#define CAN_F13R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.610 CAN_F13R1_FB6

```
#define CAN_F13R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.611 CAN_F13R1_FB7

```
#define CAN_F13R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.612 CAN_F13R1_FB8

```
#define CAN_F13R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.613 CAN_F13R1_FB9

```
#define CAN_F13R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.614 CAN_F13R2_FB0

```
#define CAN_F13R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.615 CAN_F13R2_FB1

```
#define CAN_F13R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.616 CAN_F13R2_FB10

```
#define CAN_F13R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.617 CAN_F13R2_FB11

```
#define CAN_F13R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.618 CAN_F13R2_FB12

```
#define CAN_F13R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.619 CAN_F13R2_FB13

```
#define CAN_F13R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.620 CAN_F13R2_FB14

```
#define CAN_F13R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.621 CAN_F13R2_FB15

```
#define CAN_F13R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.622 CAN_F13R2_FB16

```
#define CAN_F13R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.623 CAN_F13R2_FB17

```
#define CAN_F13R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.624 CAN_F13R2_FB18

```
#define CAN_F13R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.625 CAN_F13R2_FB19

```
#define CAN_F13R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.626 CAN_F13R2_FB2

```
#define CAN_F13R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.627 CAN_F13R2_FB20

```
#define CAN_F13R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.628 CAN_F13R2_FB21

```
#define CAN_F13R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.629 CAN_F13R2_FB22

```
#define CAN_F13R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.630 CAN_F13R2_FB23

```
#define CAN_F13R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.631 CAN_F13R2_FB24

```
#define CAN_F13R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.632 CAN_F13R2_FB25

```
#define CAN_F13R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.633 CAN_F13R2_FB26

```
#define CAN_F13R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.634 CAN_F13R2_FB27

```
#define CAN_F13R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.635 CAN_F13R2_FB28

```
#define CAN_F13R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.636 CAN_F13R2_FB29

```
#define CAN_F13R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.637 CAN_F13R2_FB3

```
#define CAN_F13R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.638 CAN_F13R2_FB30

```
#define CAN_F13R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.639 CAN_F13R2_FB31

```
#define CAN_F13R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.640 CAN_F13R2_FB4

```
#define CAN_F13R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.641 CAN_F13R2_FB5

```
#define CAN_F13R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.642 CAN_F13R2_FB6

```
#define CAN_F13R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.643 CAN_F13R2_FB7

```
#define CAN_F13R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.644 CAN_F13R2_FB8

```
#define CAN_F13R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.645 CAN_F13R2_FB9

```
#define CAN_F13R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.646 CAN_F1R1_FB0

```
#define CAN_F1R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.647 CAN_F1R1_FB1

```
#define CAN_F1R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.648 CAN_F1R1_FB10

```
#define CAN_F1R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.649 CAN_F1R1_FB11

```
#define CAN_F1R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.650 CAN_F1R1_FB12

```
#define CAN_F1R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.651 CAN_F1R1_FB13

```
#define CAN_F1R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.652 CAN_F1R1_FB14

```
#define CAN_F1R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.653 CAN_F1R1_FB15

```
#define CAN_F1R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.654 CAN_F1R1_FB16

```
#define CAN_F1R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.655 CAN_F1R1_FB17

```
#define CAN_F1R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.656 CAN_F1R1_FB18

```
#define CAN_F1R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.657 CAN_F1R1_FB19

```
#define CAN_F1R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.658 CAN_F1R1_FB2

```
#define CAN_F1R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.659 CAN_F1R1_FB20

```
#define CAN_F1R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.660 CAN_F1R1_FB21

```
#define CAN_F1R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.661 CAN_F1R1_FB22

```
#define CAN_F1R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.662 CAN_F1R1_FB23

```
#define CAN_F1R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.663 CAN_F1R1_FB24

```
#define CAN_F1R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.664 CAN_F1R1_FB25

```
#define CAN_F1R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.665 CAN_F1R1_FB26

```
#define CAN_F1R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.666 CAN_F1R1_FB27

```
#define CAN_F1R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.667 CAN_F1R1_FB28

```
#define CAN_F1R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.668 CAN_F1R1_FB29

```
#define CAN_F1R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.669 CAN_F1R1_FB3

```
#define CAN_F1R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.670 CAN_F1R1_FB30

```
#define CAN_F1R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.671 CAN_F1R1_FB31

```
#define CAN_F1R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.672 CAN_F1R1_FB4

```
#define CAN_F1R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.673 CAN_F1R1_FB5

```
#define CAN_F1R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.674 CAN_F1R1_FB6

```
#define CAN_F1R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.675 CAN_F1R1_FB7

```
#define CAN_F1R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.676 CAN_F1R1_FB8

```
#define CAN_F1R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.677 CAN_F1R1_FB9

```
#define CAN_F1R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.678 CAN_F1R2_FB0

```
#define CAN_F1R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.679 CAN_F1R2_FB1

```
#define CAN_F1R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.680 CAN_F1R2_FB10

```
#define CAN_F1R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.681 CAN_F1R2_FB11

```
#define CAN_F1R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.682 CAN_F1R2_FB12

```
#define CAN_F1R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.683 CAN_F1R2_FB13

```
#define CAN_F1R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.684 CAN_F1R2_FB14

```
#define CAN_F1R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.685 CAN_F1R2_FB15

```
#define CAN_F1R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.686 CAN_F1R2_FB16

```
#define CAN_F1R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.687 CAN_F1R2_FB17

```
#define CAN_F1R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.688 CAN_F1R2_FB18

```
#define CAN_F1R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.689 CAN_F1R2_FB19

```
#define CAN_F1R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.690 CAN_F1R2_FB2

```
#define CAN_F1R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.691 CAN_F1R2_FB20

```
#define CAN_F1R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.692 CAN_F1R2_FB21

```
#define CAN_F1R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.693 CAN_F1R2_FB22

```
#define CAN_F1R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.694 CAN_F1R2_FB23

```
#define CAN_F1R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.695 CAN_F1R2_FB24

```
#define CAN_F1R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.696 CAN_F1R2_FB25

```
#define CAN_F1R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.697 CAN_F1R2_FB26

```
#define CAN_F1R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.698 CAN_F1R2_FB27

```
#define CAN_F1R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.699 CAN_F1R2_FB28

```
#define CAN_F1R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.700 CAN_F1R2_FB29

```
#define CAN_F1R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.701 CAN_F1R2_FB3

```
#define CAN_F1R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.702 CAN_F1R2_FB30

```
#define CAN_F1R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.703 CAN_F1R2_FB31

```
#define CAN_F1R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.704 CAN_F1R2_FB4

```
#define CAN_F1R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.705 CAN_F1R2_FB5

```
#define CAN_F1R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.706 CAN_F1R2_FB6

```
#define CAN_F1R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.707 CAN_F1R2_FB7

```
#define CAN_F1R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.708 CAN_F1R2_FB8

```
#define CAN_F1R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.709 CAN_F1R2_FB9

```
#define CAN_F1R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.710 CAN_F2R1_FB0

```
#define CAN_F2R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.711 CAN_F2R1_FB1

```
#define CAN_F2R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.712 CAN_F2R1_FB10

```
#define CAN_F2R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.713 CAN_F2R1_FB11

```
#define CAN_F2R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.714 CAN_F2R1_FB12

```
#define CAN_F2R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.715 CAN_F2R1_FB13

```
#define CAN_F2R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.716 CAN_F2R1_FB14

```
#define CAN_F2R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.717 CAN_F2R1_FB15

```
#define CAN_F2R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.718 CAN_F2R1_FB16

```
#define CAN_F2R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.719 CAN_F2R1_FB17

```
#define CAN_F2R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.720 CAN_F2R1_FB18

```
#define CAN_F2R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.721 CAN_F2R1_FB19

```
#define CAN_F2R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.722 CAN_F2R1_FB2

```
#define CAN_F2R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.723 CAN_F2R1_FB20

```
#define CAN_F2R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.724 CAN_F2R1_FB21

```
#define CAN_F2R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.725 CAN_F2R1_FB22

```
#define CAN_F2R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.726 CAN_F2R1_FB23

```
#define CAN_F2R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.727 CAN_F2R1_FB24

```
#define CAN_F2R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.728 CAN_F2R1_FB25

```
#define CAN_F2R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.729 CAN_F2R1_FB26

```
#define CAN_F2R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.730 CAN_F2R1_FB27

```
#define CAN_F2R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.731 CAN_F2R1_FB28

```
#define CAN_F2R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.732 CAN_F2R1_FB29

```
#define CAN_F2R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.733 CAN_F2R1_FB3

```
#define CAN_F2R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.734 CAN_F2R1_FB30

```
#define CAN_F2R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.735 CAN_F2R1_FB31

```
#define CAN_F2R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.736 CAN_F2R1_FB4

```
#define CAN_F2R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.737 CAN_F2R1_FB5

```
#define CAN_F2R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.738 CAN_F2R1_FB6

```
#define CAN_F2R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.739 CAN_F2R1_FB7

```
#define CAN_F2R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.740 CAN_F2R1_FB8

```
#define CAN_F2R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.741 CAN_F2R1_FB9

```
#define CAN_F2R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.742 CAN_F2R2_FB0

```
#define CAN_F2R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.743 CAN_F2R2_FB1

```
#define CAN_F2R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.744 CAN_F2R2_FB10

```
#define CAN_F2R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.745 CAN_F2R2_FB11

```
#define CAN_F2R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.746 CAN_F2R2_FB12

```
#define CAN_F2R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.747 CAN_F2R2_FB13

```
#define CAN_F2R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.748 CAN_F2R2_FB14

```
#define CAN_F2R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.749 CAN_F2R2_FB15

```
#define CAN_F2R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.750 CAN_F2R2_FB16

```
#define CAN_F2R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.751 CAN_F2R2_FB17

```
#define CAN_F2R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.752 CAN_F2R2_FB18

```
#define CAN_F2R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.753 CAN_F2R2_FB19

```
#define CAN_F2R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.754 CAN_F2R2_FB2

```
#define CAN_F2R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.755 CAN_F2R2_FB20

```
#define CAN_F2R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.756 CAN_F2R2_FB21

```
#define CAN_F2R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.757 CAN_F2R2_FB22

```
#define CAN_F2R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.758 CAN_F2R2_FB23

```
#define CAN_F2R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.759 CAN_F2R2_FB24

```
#define CAN_F2R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.760 CAN_F2R2_FB25

```
#define CAN_F2R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.761 CAN_F2R2_FB26

```
#define CAN_F2R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.762 CAN_F2R2_FB27

```
#define CAN_F2R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.763 CAN_F2R2_FB28

```
#define CAN_F2R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.764 CAN_F2R2_FB29

```
#define CAN_F2R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.765 CAN_F2R2_FB3

```
#define CAN_F2R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.766 CAN_F2R2_FB30

```
#define CAN_F2R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.767 CAN_F2R2_FB31

```
#define CAN_F2R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.768 CAN_F2R2_FB4

```
#define CAN_F2R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.769 CAN_F2R2_FB5

```
#define CAN_F2R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.770 CAN_F2R2_FB6

```
#define CAN_F2R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.771 CAN_F2R2_FB7

```
#define CAN_F2R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.772 CAN_F2R2_FB8

```
#define CAN_F2R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.773 CAN_F2R2_FB9

```
#define CAN_F2R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.774 CAN_F3R1_FB0

```
#define CAN_F3R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.775 CAN_F3R1_FB1

```
#define CAN_F3R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.776 CAN_F3R1_FB10

```
#define CAN_F3R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.777 CAN_F3R1_FB11

```
#define CAN_F3R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.778 CAN_F3R1_FB12

```
#define CAN_F3R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.779 CAN_F3R1_FB13

```
#define CAN_F3R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.780 CAN_F3R1_FB14

```
#define CAN_F3R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.781 CAN_F3R1_FB15

```
#define CAN_F3R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.782 CAN_F3R1_FB16

```
#define CAN_F3R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.783 CAN_F3R1_FB17

```
#define CAN_F3R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.784 CAN_F3R1_FB18

```
#define CAN_F3R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.785 CAN_F3R1_FB19

```
#define CAN_F3R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.786 CAN_F3R1_FB2

```
#define CAN_F3R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.787 CAN_F3R1_FB20

```
#define CAN_F3R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.788 CAN_F3R1_FB21

```
#define CAN_F3R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.789 CAN_F3R1_FB22

```
#define CAN_F3R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.790 CAN_F3R1_FB23

```
#define CAN_F3R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.791 CAN_F3R1_FB24

```
#define CAN_F3R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.792 CAN_F3R1_FB25

```
#define CAN_F3R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.793 CAN_F3R1_FB26

```
#define CAN_F3R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.794 CAN_F3R1_FB27

```
#define CAN_F3R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.795 CAN_F3R1_FB28

```
#define CAN_F3R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.796 CAN_F3R1_FB29

```
#define CAN_F3R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.797 CAN_F3R1_FB3

```
#define CAN_F3R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.798 CAN_F3R1_FB30

```
#define CAN_F3R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.799 CAN_F3R1_FB31

```
#define CAN_F3R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.800 CAN_F3R1_FB4

```
#define CAN_F3R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.801 CAN_F3R1_FB5

```
#define CAN_F3R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.802 CAN_F3R1_FB6

```
#define CAN_F3R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.803 CAN_F3R1_FB7

```
#define CAN_F3R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.804 CAN_F3R1_FB8

```
#define CAN_F3R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.805 CAN_F3R1_FB9

```
#define CAN_F3R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.806 CAN_F3R2_FB0

```
#define CAN_F3R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.807 CAN_F3R2_FB1

```
#define CAN_F3R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.808 CAN_F3R2_FB10

```
#define CAN_F3R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.809 CAN_F3R2_FB11

```
#define CAN_F3R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.810 CAN_F3R2_FB12

```
#define CAN_F3R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.811 CAN_F3R2_FB13

```
#define CAN_F3R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.812 CAN_F3R2_FB14

```
#define CAN_F3R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.813 CAN_F3R2_FB15

```
#define CAN_F3R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.814 CAN_F3R2_FB16

```
#define CAN_F3R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.815 CAN_F3R2_FB17

```
#define CAN_F3R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.816 CAN_F3R2_FB18

```
#define CAN_F3R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.817 CAN_F3R2_FB19

```
#define CAN_F3R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.818 CAN_F3R2_FB2

```
#define CAN_F3R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.819 CAN_F3R2_FB20

```
#define CAN_F3R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.820 CAN_F3R2_FB21

```
#define CAN_F3R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.821 CAN_F3R2_FB22

```
#define CAN_F3R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.822 CAN_F3R2_FB23

```
#define CAN_F3R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.823 CAN_F3R2_FB24

```
#define CAN_F3R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.824 CAN_F3R2_FB25

```
#define CAN_F3R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.825 CAN_F3R2_FB26

```
#define CAN_F3R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.826 CAN_F3R2_FB27

```
#define CAN_F3R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.827 CAN_F3R2_FB28

```
#define CAN_F3R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.828 CAN_F3R2_FB29

```
#define CAN_F3R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.829 CAN_F3R2_FB3

```
#define CAN_F3R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.830 CAN_F3R2_FB30

```
#define CAN_F3R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.831 CAN_F3R2_FB31

```
#define CAN_F3R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.832 CAN_F3R2_FB4

```
#define CAN_F3R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.833 CAN_F3R2_FB5

```
#define CAN_F3R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.834 CAN_F3R2_FB6

```
#define CAN_F3R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.835 CAN_F3R2_FB7

```
#define CAN_F3R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.836 CAN_F3R2_FB8

```
#define CAN_F3R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.837 CAN_F3R2_FB9

```
#define CAN_F3R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.838 CAN_F4R1_FB0

```
#define CAN_F4R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.839 CAN_F4R1_FB1

```
#define CAN_F4R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.840 CAN_F4R1_FB10

```
#define CAN_F4R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.841 CAN_F4R1_FB11

```
#define CAN_F4R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.842 CAN_F4R1_FB12

```
#define CAN_F4R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.843 CAN_F4R1_FB13

```
#define CAN_F4R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.844 CAN_F4R1_FB14

```
#define CAN_F4R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.845 CAN_F4R1_FB15

```
#define CAN_F4R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.846 CAN_F4R1_FB16

```
#define CAN_F4R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.847 CAN_F4R1_FB17

```
#define CAN_F4R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.848 CAN_F4R1_FB18

```
#define CAN_F4R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.849 CAN_F4R1_FB19

```
#define CAN_F4R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.850 CAN_F4R1_FB2

```
#define CAN_F4R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.851 CAN_F4R1_FB20

```
#define CAN_F4R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.852 CAN_F4R1_FB21

```
#define CAN_F4R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.853 CAN_F4R1_FB22

```
#define CAN_F4R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.854 CAN_F4R1_FB23

```
#define CAN_F4R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.855 CAN_F4R1_FB24

```
#define CAN_F4R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.856 CAN_F4R1_FB25

```
#define CAN_F4R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.857 CAN_F4R1_FB26

```
#define CAN_F4R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.858 CAN_F4R1_FB27

```
#define CAN_F4R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.859 CAN_F4R1_FB28

```
#define CAN_F4R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.860 CAN_F4R1_FB29

```
#define CAN_F4R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.861 CAN_F4R1_FB3

```
#define CAN_F4R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.862 CAN_F4R1_FB30

```
#define CAN_F4R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.863 CAN_F4R1_FB31

```
#define CAN_F4R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.864 CAN_F4R1_FB4

```
#define CAN_F4R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.865 CAN_F4R1_FB5

```
#define CAN_F4R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.866 CAN_F4R1_FB6

```
#define CAN_F4R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.867 CAN_F4R1_FB7

```
#define CAN_F4R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.868 CAN_F4R1_FB8

```
#define CAN_F4R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.869 CAN_F4R1_FB9

```
#define CAN_F4R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.870 CAN_F4R2_FB0

```
#define CAN_F4R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.871 CAN_F4R2_FB1

```
#define CAN_F4R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.872 CAN_F4R2_FB10

```
#define CAN_F4R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.873 CAN_F4R2_FB11

```
#define CAN_F4R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.874 CAN_F4R2_FB12

```
#define CAN_F4R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.875 CAN_F4R2_FB13

```
#define CAN_F4R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.876 CAN_F4R2_FB14

```
#define CAN_F4R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.877 CAN_F4R2_FB15

```
#define CAN_F4R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.878 CAN_F4R2_FB16

```
#define CAN_F4R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.879 CAN_F4R2_FB17

```
#define CAN_F4R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.880 CAN_F4R2_FB18

```
#define CAN_F4R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.881 CAN_F4R2_FB19

```
#define CAN_F4R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.882 CAN_F4R2_FB2

```
#define CAN_F4R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.883 CAN_F4R2_FB20

```
#define CAN_F4R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.884 CAN_F4R2_FB21

```
#define CAN_F4R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.885 CAN_F4R2_FB22

```
#define CAN_F4R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.886 CAN_F4R2_FB23

```
#define CAN_F4R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.887 CAN_F4R2_FB24

```
#define CAN_F4R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.888 CAN_F4R2_FB25

```
#define CAN_F4R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.889 CAN_F4R2_FB26

```
#define CAN_F4R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.890 CAN_F4R2_FB27

```
#define CAN_F4R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.891 CAN_F4R2_FB28

```
#define CAN_F4R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.892 CAN_F4R2_FB29

```
#define CAN_F4R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.893 CAN_F4R2_FB3

```
#define CAN_F4R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.894 CAN_F4R2_FB30

```
#define CAN_F4R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.895 CAN_F4R2_FB31

```
#define CAN_F4R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.896 CAN_F4R2_FB4

```
#define CAN_F4R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.897 CAN_F4R2_FB5

```
#define CAN_F4R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.898 CAN_F4R2_FB6

```
#define CAN_F4R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.899 CAN_F4R2_FB7

```
#define CAN_F4R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.900 CAN_F4R2_FB8

```
#define CAN_F4R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.901 CAN_F4R2_FB9

```
#define CAN_F4R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.902 CAN_F5R1_FB0

```
#define CAN_F5R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.903 CAN_F5R1_FB1

```
#define CAN_F5R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.904 CAN_F5R1_FB10

```
#define CAN_F5R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.905 CAN_F5R1_FB11

```
#define CAN_F5R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.906 CAN_F5R1_FB12

```
#define CAN_F5R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.907 CAN_F5R1_FB13

```
#define CAN_F5R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.908 CAN_F5R1_FB14

```
#define CAN_F5R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.909 CAN_F5R1_FB15

```
#define CAN_F5R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.910 CAN_F5R1_FB16

```
#define CAN_F5R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.911 CAN_F5R1_FB17

```
#define CAN_F5R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.912 CAN_F5R1_FB18

```
#define CAN_F5R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.913 CAN_F5R1_FB19

```
#define CAN_F5R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.914 CAN_F5R1_FB2

```
#define CAN_F5R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.915 CAN_F5R1_FB20

```
#define CAN_F5R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.916 CAN_F5R1_FB21

```
#define CAN_F5R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.917 CAN_F5R1_FB22

```
#define CAN_F5R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.918 CAN_F5R1_FB23

```
#define CAN_F5R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.919 CAN_F5R1_FB24

```
#define CAN_F5R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.920 CAN_F5R1_FB25

```
#define CAN_F5R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.921 CAN_F5R1_FB26

```
#define CAN_F5R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.922 CAN_F5R1_FB27

```
#define CAN_F5R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.923 CAN_F5R1_FB28

```
#define CAN_F5R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.924 CAN_F5R1_FB29

```
#define CAN_F5R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.925 CAN_F5R1_FB3

```
#define CAN_F5R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.926 CAN_F5R1_FB30

```
#define CAN_F5R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.927 CAN_F5R1_FB31

```
#define CAN_F5R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.928 CAN_F5R1_FB4

```
#define CAN_F5R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.929 CAN_F5R1_FB5

```
#define CAN_F5R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.930 CAN_F5R1_FB6

```
#define CAN_F5R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.931 CAN_F5R1_FB7

```
#define CAN_F5R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.932 CAN_F5R1_FB8

```
#define CAN_F5R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.933 CAN_F5R1_FB9

```
#define CAN_F5R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.934 CAN_F5R2_FB0

```
#define CAN_F5R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.935 CAN_F5R2_FB1

```
#define CAN_F5R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.936 CAN_F5R2_FB10

```
#define CAN_F5R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.937 CAN_F5R2_FB11

```
#define CAN_F5R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.938 CAN_F5R2_FB12

```
#define CAN_F5R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.939 CAN_F5R2_FB13

```
#define CAN_F5R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.940 CAN_F5R2_FB14

```
#define CAN_F5R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.941 CAN_F5R2_FB15

```
#define CAN_F5R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.942 CAN_F5R2_FB16

```
#define CAN_F5R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.943 CAN_F5R2_FB17

```
#define CAN_F5R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.944 CAN_F5R2_FB18

```
#define CAN_F5R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.945 CAN_F5R2_FB19

```
#define CAN_F5R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.946 CAN_F5R2_FB2

```
#define CAN_F5R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.947 CAN_F5R2_FB20

```
#define CAN_F5R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.948 CAN_F5R2_FB21

```
#define CAN_F5R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.949 CAN_F5R2_FB22

```
#define CAN_F5R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.950 CAN_F5R2_FB23

```
#define CAN_F5R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.951 CAN_F5R2_FB24

```
#define CAN_F5R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.952 CAN_F5R2_FB25

```
#define CAN_F5R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.953 CAN_F5R2_FB26

```
#define CAN_F5R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.954 CAN_F5R2_FB27

```
#define CAN_F5R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.955 CAN_F5R2_FB28

```
#define CAN_F5R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.956 CAN_F5R2_FB29

```
#define CAN_F5R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.957 CAN_F5R2_FB3

```
#define CAN_F5R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.958 CAN_F5R2_FB30

```
#define CAN_F5R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.959 CAN_F5R2_FB31

```
#define CAN_F5R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.960 CAN_F5R2_FB4

```
#define CAN_F5R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.961 CAN_F5R2_FB5

```
#define CAN_F5R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.962 CAN_F5R2_FB6

```
#define CAN_F5R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.963 CAN_F5R2_FB7

```
#define CAN_F5R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.964 CAN_F5R2_FB8

```
#define CAN_F5R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.965 CAN_F5R2_FB9

```
#define CAN_F5R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.966 CAN_F6R1_FB0

```
#define CAN_F6R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.967 CAN_F6R1_FB1

```
#define CAN_F6R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.968 CAN_F6R1_FB10

```
#define CAN_F6R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.969 CAN_F6R1_FB11

```
#define CAN_F6R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.970 CAN_F6R1_FB12

```
#define CAN_F6R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.971 CAN_F6R1_FB13

```
#define CAN_F6R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.972 CAN_F6R1_FB14

```
#define CAN_F6R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.973 CAN_F6R1_FB15

```
#define CAN_F6R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.974 CAN_F6R1_FB16

```
#define CAN_F6R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.975 CAN_F6R1_FB17

```
#define CAN_F6R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.976 CAN_F6R1_FB18

```
#define CAN_F6R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.977 CAN_F6R1_FB19

```
#define CAN_F6R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.978 CAN_F6R1_FB2

```
#define CAN_F6R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.979 CAN_F6R1_FB20

```
#define CAN_F6R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.980 CAN_F6R1_FB21

```
#define CAN_F6R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.981 CAN_F6R1_FB22

```
#define CAN_F6R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.982 CAN_F6R1_FB23

```
#define CAN_F6R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.983 CAN_F6R1_FB24

```
#define CAN_F6R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.984 CAN_F6R1_FB25

```
#define CAN_F6R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.985 CAN_F6R1_FB26

```
#define CAN_F6R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.986 CAN_F6R1_FB27

```
#define CAN_F6R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.987 CAN_F6R1_FB28

```
#define CAN_F6R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.988 CAN_F6R1_FB29

```
#define CAN_F6R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.989 CAN_F6R1_FB3

```
#define CAN_F6R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.990 CAN_F6R1_FB30

```
#define CAN_F6R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.991 CAN_F6R1_FB31

```
#define CAN_F6R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.992 CAN_F6R1_FB4

```
#define CAN_F6R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.993 CAN_F6R1_FB5

```
#define CAN_F6R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.994 CAN_F6R1_FB6

```
#define CAN_F6R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.995 CAN_F6R1_FB7

```
#define CAN_F6R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.996 CAN_F6R1_FB8

```
#define CAN_F6R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.997 CAN_F6R1_FB9

```
#define CAN_F6R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.998 CAN_F6R2_FB0

```
#define CAN_F6R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.999 CAN_F6R2_FB1

```
#define CAN_F6R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.1000 CAN_F6R2_FB10

```
#define CAN_F6R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.1001 CAN_F6R2_FB11

```
#define CAN_F6R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.1002 CAN_F6R2_FB12

```
#define CAN_F6R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.1003 CAN_F6R2_FB13

```
#define CAN_F6R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.1004 CAN_F6R2_FB14

```
#define CAN_F6R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.1005 CAN_F6R2_FB15

```
#define CAN_F6R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.1006 CAN_F6R2_FB16

```
#define CAN_F6R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.1007 CAN_F6R2_FB17

```
#define CAN_F6R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.1008 CAN_F6R2_FB18

```
#define CAN_F6R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.1009 CAN_F6R2_FB19

```
#define CAN_F6R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.1010 CAN_F6R2_FB2

```
#define CAN_F6R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.1011 CAN_F6R2_FB20

```
#define CAN_F6R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.1012 CAN_F6R2_FB21

```
#define CAN_F6R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.1013 CAN_F6R2_FB22

```
#define CAN_F6R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.1014 CAN_F6R2_FB23

```
#define CAN_F6R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.1015 CAN_F6R2_FB24

```
#define CAN_F6R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.1016 CAN_F6R2_FB25

```
#define CAN_F6R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.1017 CAN_F6R2_FB26

```
#define CAN_F6R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.1018 CAN_F6R2_FB27

```
#define CAN_F6R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.1019 CAN_F6R2_FB28

```
#define CAN_F6R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.1020 CAN_F6R2_FB29

```
#define CAN_F6R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.1021 CAN_F6R2_FB3

```
#define CAN_F6R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.1022 CAN_F6R2_FB30

```
#define CAN_F6R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.1023 CAN_F6R2_FB31

```
#define CAN_F6R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.1024 CAN_F6R2_FB4

```
#define CAN_F6R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.1025 CAN_F6R2_FB5

```
#define CAN_F6R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.1026 CAN_F6R2_FB6

```
#define CAN_F6R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.1027 CAN_F6R2_FB7

```
#define CAN_F6R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.1028 CAN_F6R2_FB8

```
#define CAN_F6R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.1029 CAN_F6R2_FB9

```
#define CAN_F6R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.1030 CAN_F7R1_FB0

```
#define CAN_F7R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.1031 CAN_F7R1_FB1

```
#define CAN_F7R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.1032 CAN_F7R1_FB10

```
#define CAN_F7R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.1033 CAN_F7R1_FB11

```
#define CAN_F7R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.1034 CAN_F7R1_FB12

```
#define CAN_F7R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.1035 CAN_F7R1_FB13

```
#define CAN_F7R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.1036 CAN_F7R1_FB14

```
#define CAN_F7R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.1037 CAN_F7R1_FB15

```
#define CAN_F7R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.1038 CAN_F7R1_FB16

```
#define CAN_F7R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.1039 CAN_F7R1_FB17

```
#define CAN_F7R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.1040 CAN_F7R1_FB18

```
#define CAN_F7R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.1041 CAN_F7R1_FB19

```
#define CAN_F7R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.1042 CAN_F7R1_FB2

```
#define CAN_F7R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.1043 CAN_F7R1_FB20

```
#define CAN_F7R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.1044 CAN_F7R1_FB21

```
#define CAN_F7R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.1045 CAN_F7R1_FB22

```
#define CAN_F7R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.1046 CAN_F7R1_FB23

```
#define CAN_F7R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.1047 CAN_F7R1_FB24

```
#define CAN_F7R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.1048 CAN_F7R1_FB25

```
#define CAN_F7R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.1049 CAN_F7R1_FB26

```
#define CAN_F7R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.1050 CAN_F7R1_FB27

```
#define CAN_F7R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.1051 CAN_F7R1_FB28

```
#define CAN_F7R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.1052 CAN_F7R1_FB29

```
#define CAN_F7R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.1053 CAN_F7R1_FB3

```
#define CAN_F7R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.1054 CAN_F7R1_FB30

```
#define CAN_F7R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.1055 CAN_F7R1_FB31

```
#define CAN_F7R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.1056 CAN_F7R1_FB4

```
#define CAN_F7R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.1057 CAN_F7R1_FB5

```
#define CAN_F7R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.1058 CAN_F7R1_FB6

```
#define CAN_F7R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.1059 CAN_F7R1_FB7

```
#define CAN_F7R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.1060 CAN_F7R1_FB8

```
#define CAN_F7R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.1061 CAN_F7R1_FB9

```
#define CAN_F7R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.1062 CAN_F7R2_FB0

```
#define CAN_F7R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.1063 CAN_F7R2_FB1

```
#define CAN_F7R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.1064 CAN_F7R2_FB10

```
#define CAN_F7R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.1065 CAN_F7R2_FB11

```
#define CAN_F7R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.1066 CAN_F7R2_FB12

```
#define CAN_F7R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.1067 CAN_F7R2_FB13

```
#define CAN_F7R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.1068 CAN_F7R2_FB14

```
#define CAN_F7R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.1069 CAN_F7R2_FB15

```
#define CAN_F7R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.1070 CAN_F7R2_FB16

```
#define CAN_F7R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.1071 CAN_F7R2_FB17

```
#define CAN_F7R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.1072 CAN_F7R2_FB18

```
#define CAN_F7R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.1073 CAN_F7R2_FB19

```
#define CAN_F7R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.1074 CAN_F7R2_FB2

```
#define CAN_F7R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.1075 CAN_F7R2_FB20

```
#define CAN_F7R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.1076 CAN_F7R2_FB21

```
#define CAN_F7R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.1077 CAN_F7R2_FB22

```
#define CAN_F7R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.1078 CAN_F7R2_FB23

```
#define CAN_F7R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.1079 CAN_F7R2_FB24

```
#define CAN_F7R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.1080 CAN_F7R2_FB25

```
#define CAN_F7R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.1081 CAN_F7R2_FB26

```
#define CAN_F7R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.1082 CAN_F7R2_FB27

```
#define CAN_F7R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.1083 CAN_F7R2_FB28

```
#define CAN_F7R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.1084 CAN_F7R2_FB29

```
#define CAN_F7R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.1085 CAN_F7R2_FB3

```
#define CAN_F7R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.1086 CAN_F7R2_FB30

```
#define CAN_F7R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.1087 CAN_F7R2_FB31

```
#define CAN_F7R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.1088 CAN_F7R2_FB4

```
#define CAN_F7R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.1089 CAN_F7R2_FB5

```
#define CAN_F7R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.1090 CAN_F7R2_FB6

```
#define CAN_F7R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.1091 CAN_F7R2_FB7

```
#define CAN_F7R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.1092 CAN_F7R2_FB8

```
#define CAN_F7R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.1093 CAN_F7R2_FB9

```
#define CAN_F7R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.1094 CAN_F8R1_FB0

```
#define CAN_F8R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.1095 CAN_F8R1_FB1

```
#define CAN_F8R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.1096 CAN_F8R1_FB10

```
#define CAN_F8R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.1097 CAN_F8R1_FB11

```
#define CAN_F8R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.1098 CAN_F8R1_FB12

```
#define CAN_F8R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.1099 CAN_F8R1_FB13

```
#define CAN_F8R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.1100 CAN_F8R1_FB14

```
#define CAN_F8R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.1101 CAN_F8R1_FB15

```
#define CAN_F8R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.1102 CAN_F8R1_FB16

```
#define CAN_F8R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.1103 CAN_F8R1_FB17

```
#define CAN_F8R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.1104 CAN_F8R1_FB18

```
#define CAN_F8R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.1105 CAN_F8R1_FB19

```
#define CAN_F8R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.1106 CAN_F8R1_FB2

```
#define CAN_F8R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.1107 CAN_F8R1_FB20

```
#define CAN_F8R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.1108 CAN_F8R1_FB21

```
#define CAN_F8R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.1109 CAN_F8R1_FB22

```
#define CAN_F8R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.1110 CAN_F8R1_FB23

```
#define CAN_F8R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.1111 CAN_F8R1_FB24

```
#define CAN_F8R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.1112 CAN_F8R1_FB25

```
#define CAN_F8R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.1113 CAN_F8R1_FB26

```
#define CAN_F8R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.1114 CAN_F8R1_FB27

```
#define CAN_F8R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.1115 CAN_F8R1_FB28

```
#define CAN_F8R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.1116 CAN_F8R1_FB29

```
#define CAN_F8R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.1117 CAN_F8R1_FB3

```
#define CAN_F8R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.1118 CAN_F8R1_FB30

```
#define CAN_F8R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.1119 CAN_F8R1_FB31

```
#define CAN_F8R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.1120 CAN_F8R1_FB4

```
#define CAN_F8R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.1121 CAN_F8R1_FB5

```
#define CAN_F8R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.1122 CAN_F8R1_FB6

```
#define CAN_F8R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.1123 CAN_F8R1_FB7

```
#define CAN_F8R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.1124 CAN_F8R1_FB8

```
#define CAN_F8R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.1125 CAN_F8R1_FB9

```
#define CAN_F8R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.1126 CAN_F8R2_FB0

```
#define CAN_F8R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.1127 CAN_F8R2_FB1

```
#define CAN_F8R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.1128 CAN_F8R2_FB10

```
#define CAN_F8R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.1129 CAN_F8R2_FB11

```
#define CAN_F8R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.1130 CAN_F8R2_FB12

```
#define CAN_F8R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.1131 CAN_F8R2_FB13

```
#define CAN_F8R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.1132 CAN_F8R2_FB14

```
#define CAN_F8R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.1133 CAN_F8R2_FB15

```
#define CAN_F8R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.1134 CAN_F8R2_FB16

```
#define CAN_F8R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.1135 CAN_F8R2_FB17

```
#define CAN_F8R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.1136 CAN_F8R2_FB18

```
#define CAN_F8R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.1137 CAN_F8R2_FB19

```
#define CAN_F8R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.1138 CAN_F8R2_FB2

```
#define CAN_F8R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.1139 CAN_F8R2_FB20

```
#define CAN_F8R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.1140 CAN_F8R2_FB21

```
#define CAN_F8R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.1141 CAN_F8R2_FB22

```
#define CAN_F8R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.1142 CAN_F8R2_FB23

```
#define CAN_F8R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.1143 CAN_F8R2_FB24

```
#define CAN_F8R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.1144 CAN_F8R2_FB25

```
#define CAN_F8R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.1145 CAN_F8R2_FB26

```
#define CAN_F8R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.1146 CAN_F8R2_FB27

```
#define CAN_F8R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.1147 CAN_F8R2_FB28

```
#define CAN_F8R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.1148 CAN_F8R2_FB29

```
#define CAN_F8R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.1149 CAN_F8R2_FB3

```
#define CAN_F8R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.1150 CAN_F8R2_FB30

```
#define CAN_F8R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.1151 CAN_F8R2_FB31

```
#define CAN_F8R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.1152 CAN_F8R2_FB4

```
#define CAN_F8R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.1153 CAN_F8R2_FB5

```
#define CAN_F8R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.1154 CAN_F8R2_FB6

```
#define CAN_F8R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.1155 CAN_F8R2_FB7

```
#define CAN_F8R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.1156 CAN_F8R2_FB8

```
#define CAN_F8R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.1157 CAN_F8R2_FB9

```
#define CAN_F8R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.1158 CAN_F9R1_FB0

```
#define CAN_F9R1_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.1159 CAN_F9R1_FB1

```
#define CAN_F9R1_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.1160 CAN_F9R1_FB10

```
#define CAN_F9R1_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.1161 CAN_F9R1_FB11

```
#define CAN_F9R1_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.1162 CAN_F9R1_FB12

```
#define CAN_F9R1_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.1163 CAN_F9R1_FB13

```
#define CAN_F9R1_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.1164 CAN_F9R1_FB14

```
#define CAN_F9R1_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.1165 CAN_F9R1_FB15

```
#define CAN_F9R1_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.1166 CAN_F9R1_FB16

```
#define CAN_F9R1_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.1167 CAN_F9R1_FB17

```
#define CAN_F9R1_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.1168 CAN_F9R1_FB18

```
#define CAN_F9R1_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.1169 CAN_F9R1_FB19

```
#define CAN_F9R1_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.1170 CAN_F9R1_FB2

```
#define CAN_F9R1_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.1171 CAN_F9R1_FB20

```
#define CAN_F9R1_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.1172 CAN_F9R1_FB21

```
#define CAN_F9R1_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.1173 CAN_F9R1_FB22

```
#define CAN_F9R1_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.1174 CAN_F9R1_FB23

```
#define CAN_F9R1_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.1175 CAN_F9R1_FB24

```
#define CAN_F9R1_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.1176 CAN_F9R1_FB25

```
#define CAN_F9R1_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.1177 CAN_F9R1_FB26

```
#define CAN_F9R1_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.1178 CAN_F9R1_FB27

```
#define CAN_F9R1_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.1179 CAN_F9R1_FB28

```
#define CAN_F9R1_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.1180 CAN_F9R1_FB29

```
#define CAN_F9R1_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.1181 CAN_F9R1_FB3

```
#define CAN_F9R1_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.1182 CAN_F9R1_FB30

```
#define CAN_F9R1_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.1183 CAN_F9R1_FB31

```
#define CAN_F9R1_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.1184 CAN_F9R1_FB4

```
#define CAN_F9R1_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.1185 CAN_F9R1_FB5

```
#define CAN_F9R1_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.1186 CAN_F9R1_FB6

```
#define CAN_F9R1_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.1187 CAN_F9R1_FB7

```
#define CAN_F9R1_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.1188 CAN_F9R1_FB8

```
#define CAN_F9R1_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.1189 CAN_F9R1_FB9

```
#define CAN_F9R1_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.1190 CAN_F9R2_FB0

```
#define CAN_F9R2_FB0 ((uint32_t)0x00000001)
```

Filter bit 0

5.173.2.1191 CAN_F9R2_FB1

```
#define CAN_F9R2_FB1 ((uint32_t)0x00000002)
```

Filter bit 1

5.173.2.1192 CAN_F9R2_FB10

```
#define CAN_F9R2_FB10 ((uint32_t)0x00000400)
```

Filter bit 10

5.173.2.1193 CAN_F9R2_FB11

```
#define CAN_F9R2_FB11 ((uint32_t)0x00000800)
```

Filter bit 11

5.173.2.1194 CAN_F9R2_FB12

```
#define CAN_F9R2_FB12 ((uint32_t)0x00001000)
```

Filter bit 12

5.173.2.1195 CAN_F9R2_FB13

```
#define CAN_F9R2_FB13 ((uint32_t)0x00002000)
```

Filter bit 13

5.173.2.1196 CAN_F9R2_FB14

```
#define CAN_F9R2_FB14 ((uint32_t)0x00004000)
```

Filter bit 14

5.173.2.1197 CAN_F9R2_FB15

```
#define CAN_F9R2_FB15 ((uint32_t)0x00008000)
```

Filter bit 15

5.173.2.1198 CAN_F9R2_FB16

```
#define CAN_F9R2_FB16 ((uint32_t)0x00010000)
```

Filter bit 16

5.173.2.1199 CAN_F9R2_FB17

```
#define CAN_F9R2_FB17 ((uint32_t)0x00020000)
```

Filter bit 17

5.173.2.1200 CAN_F9R2_FB18

```
#define CAN_F9R2_FB18 ((uint32_t)0x00040000)
```

Filter bit 18

5.173.2.1201 CAN_F9R2_FB19

```
#define CAN_F9R2_FB19 ((uint32_t)0x00080000)
```

Filter bit 19

5.173.2.1202 CAN_F9R2_FB2

```
#define CAN_F9R2_FB2 ((uint32_t)0x00000004)
```

Filter bit 2

5.173.2.1203 CAN_F9R2_FB20

```
#define CAN_F9R2_FB20 ((uint32_t)0x00100000)
```

Filter bit 20

5.173.2.1204 CAN_F9R2_FB21

```
#define CAN_F9R2_FB21 ((uint32_t)0x00200000)
```

Filter bit 21

5.173.2.1205 CAN_F9R2_FB22

```
#define CAN_F9R2_FB22 ((uint32_t)0x00400000)
```

Filter bit 22

5.173.2.1206 CAN_F9R2_FB23

```
#define CAN_F9R2_FB23 ((uint32_t)0x00800000)
```

Filter bit 23

5.173.2.1207 CAN_F9R2_FB24

```
#define CAN_F9R2_FB24 ((uint32_t)0x01000000)
```

Filter bit 24

5.173.2.1208 CAN_F9R2_FB25

```
#define CAN_F9R2_FB25 ((uint32_t)0x02000000)
```

Filter bit 25

5.173.2.1209 CAN_F9R2_FB26

```
#define CAN_F9R2_FB26 ((uint32_t)0x04000000)
```

Filter bit 26

5.173.2.1210 CAN_F9R2_FB27

```
#define CAN_F9R2_FB27 ((uint32_t)0x08000000)
```

Filter bit 27

5.173.2.1211 CAN_F9R2_FB28

```
#define CAN_F9R2_FB28 ((uint32_t)0x10000000)
```

Filter bit 28

5.173.2.1212 CAN_F9R2_FB29

```
#define CAN_F9R2_FB29 ((uint32_t)0x20000000)
```

Filter bit 29

5.173.2.1213 CAN_F9R2_FB3

```
#define CAN_F9R2_FB3 ((uint32_t)0x00000008)
```

Filter bit 3

5.173.2.1214 CAN_F9R2_FB30

```
#define CAN_F9R2_FB30 ((uint32_t)0x40000000)
```

Filter bit 30

5.173.2.1215 CAN_F9R2_FB31

```
#define CAN_F9R2_FB31 ((uint32_t)0x80000000)
```

Filter bit 31

5.173.2.1216 CAN_F9R2_FB4

```
#define CAN_F9R2_FB4 ((uint32_t)0x00000010)
```

Filter bit 4

5.173.2.1217 CAN_F9R2_FB5

```
#define CAN_F9R2_FB5 ((uint32_t)0x00000020)
```

Filter bit 5

5.173.2.1218 CAN_F9R2_FB6

```
#define CAN_F9R2_FB6 ((uint32_t)0x00000040)
```

Filter bit 6

5.173.2.1219 CAN_F9R2_FB7

```
#define CAN_F9R2_FB7 ((uint32_t)0x00000080)
```

Filter bit 7

5.173.2.1220 CAN_F9R2_FB8

```
#define CAN_F9R2_FB8 ((uint32_t)0x00000100)
```

Filter bit 8

5.173.2.1221 CAN_F9R2_FB9

```
#define CAN_F9R2_FB9 ((uint32_t)0x00000200)
```

Filter bit 9

5.173.2.1222 CAN_FA1R_FACT

```
#define CAN_FA1R_FACT ((uint16_t)0x3FFF)
```

Filter Active

5.173.2.1223 CAN_FA1R_FACT0

```
#define CAN_FA1R_FACT0 ((uint16_t)0x0001)
```

Filter 0 Active

5.173.2.1224 CAN_FA1R_FACT1

```
#define CAN_FA1R_FACT1 ((uint16_t)0x0002)
```

Filter 1 Active

5.173.2.1225 CAN_FA1R_FACT10

```
#define CAN_FA1R_FACT10 ((uint16_t)0x0400)
```

Filter 10 Active

5.173.2.1226 CAN_FA1R_FACT11

```
#define CAN_FA1R_FACT11 ((uint16_t)0x0800)
```

Filter 11 Active

5.173.2.1227 CAN_FA1R_FACT12

```
#define CAN_FA1R_FACT12 ((uint16_t)0x1000)
```

Filter 12 Active

5.173.2.1228 CAN_FA1R_FACT13

```
#define CAN_FA1R_FACT13 ((uint16_t)0x2000)
```

Filter 13 Active

5.173.2.1229 CAN_FA1R_FACT2

```
#define CAN_FA1R_FACT2 ((uint16_t)0x0004)
```

Filter 2 Active

5.173.2.1230 CAN_FA1R_FACT3

```
#define CAN_FA1R_FACT3 ((uint16_t)0x0008)
```

Filter 3 Active

5.173.2.1231 CAN_FA1R_FACT4

```
#define CAN_FA1R_FACT4 ((uint16_t)0x0010)
```

Filter 4 Active

5.173.2.1232 CAN_FA1R_FACT5

```
#define CAN_FA1R_FACT5 ((uint16_t)0x0020)
```

Filter 5 Active

5.173.2.1233 CAN_FA1R_FACT6

```
#define CAN_FA1R_FACT6 ((uint16_t)0x0040)
```

Filter 6 Active

5.173.2.1234 CAN_FA1R_FACT7

```
#define CAN_FA1R_FACT7 ((uint16_t)0x0080)
```

Filter 7 Active

5.173.2.1235 CAN_FA1R_FACT8

```
#define CAN_FA1R_FACT8 ((uint16_t)0x0100)
```

Filter 8 Active

5.173.2.1236 CAN_FA1R_FACT9

```
#define CAN_FA1R_FACT9 ((uint16_t)0x0200)
```

Filter 9 Active

5.173.2.1237 CAN_FFA1R_FFA

```
#define CAN_FFA1R_FFA ((uint16_t)0x3FFF)
```

Filter FIFO Assignment

5.173.2.1238 CAN_FFA1R_FFA0

```
#define CAN_FFA1R_FFA0 ((uint16_t)0x0001)
```

Filter FIFO Assignment for Filter 0

5.173.2.1239 CAN_FFA1R_FFA1

```
#define CAN_FFA1R_FFA1 ((uint16_t)0x0002)
```

Filter FIFO Assignment for Filter 1

5.173.2.1240 CAN_FFA1R_FFA10

```
#define CAN_FFA1R_FFA10 ((uint16_t)0x0400)
```

Filter FIFO Assignment for Filter 10

5.173.2.1241 CAN_FFA1R_FFA11

```
#define CAN_FFA1R_FFA11 ((uint16_t)0x0800)
```

Filter FIFO Assignment for Filter 11

5.173.2.1242 CAN_FFA1R_FFA12

```
#define CAN_FFA1R_FFA12 ((uint16_t)0x1000)
```

Filter FIFO Assignment for Filter 12

5.173.2.1243 CAN_FFA1R_FFA13

```
#define CAN_FFA1R_FFA13 ((uint16_t)0x2000)
```

Filter FIFO Assignment for Filter 13

5.173.2.1244 CAN_FFA1R_FFA2

```
#define CAN_FFA1R_FFA2 ((uint16_t)0x0004)
```

Filter FIFO Assignment for Filter 2

5.173.2.1245 CAN_FFA1R_FFA3

```
#define CAN_FFA1R_FFA3 ((uint16_t)0x0008)
```

Filter FIFO Assignment for Filter 3

5.173.2.1246 CAN_FFA1R_FFA4

```
#define CAN_FFA1R_FFA4 ((uint16_t)0x0010)
```

Filter FIFO Assignment for Filter 4

5.173.2.1247 CAN_FFA1R_FFA5

```
#define CAN_FFA1R_FFA5 ((uint16_t)0x0020)
```

Filter FIFO Assignment for Filter 5

5.173.2.1248 CAN_FFA1R_FFA6

```
#define CAN_FFA1R_FFA6 ((uint16_t)0x0040)
```

Filter FIFO Assignment for Filter 6

5.173.2.1249 CAN_FFA1R_FFA7

```
#define CAN_FFA1R_FFA7 ((uint16_t)0x0080)
```

Filter FIFO Assignment for Filter 7

5.173.2.1250 CAN_FFA1R_FFA8

```
#define CAN_FFA1R_FFA8 ((uint16_t)0x0100)
```

Filter FIFO Assignment for Filter 8

5.173.2.1251 CAN_FFA1R_FFA9

```
#define CAN_FFA1R_FFA9 ((uint16_t)0x0200)
```

Filter FIFO Assignment for Filter 9

5.173.2.1252 CAN_FM1R_FBM

```
#define CAN_FM1R_FBM ((uint16_t)0x3FFF)
```

Filter Mode

5.173.2.1253 CAN_FM1R_FBM0

```
#define CAN_FM1R_FBM0 ((uint16_t)0x0001)
```

Filter Init Mode bit 0

5.173.2.1254 CAN_FM1R_FBM1

```
#define CAN_FM1R_FBM1 ((uint16_t)0x0002)
```

Filter Init Mode bit 1

5.173.2.1255 CAN_FM1R_FBM10

```
#define CAN_FM1R_FBM10 ((uint16_t)0x0400)
```

Filter Init Mode bit 10

5.173.2.1256 CAN_FM1R_FBM11

```
#define CAN_FM1R_FBM11 ((uint16_t)0x0800)
```

Filter Init Mode bit 11

5.173.2.1257 CAN_FM1R_FBM12

```
#define CAN_FM1R_FBM12 ((uint16_t)0x1000)
```

Filter Init Mode bit 12

5.173.2.1258 CAN_FM1R_FBM13

```
#define CAN_FM1R_FBM13 ((uint16_t)0x2000)
```

Filter Init Mode bit 13

5.173.2.1259 CAN_FM1R_FBM2

```
#define CAN_FM1R_FBM2 ((uint16_t)0x0004)
```

Filter Init Mode bit 2

5.173.2.1260 CAN_FM1R_FBM3

```
#define CAN_FM1R_FBM3 ((uint16_t)0x0008)
```

Filter Init Mode bit 3

5.173.2.1261 CAN_FM1R_FBM4

```
#define CAN_FM1R_FBM4 ((uint16_t)0x0010)
```

Filter Init Mode bit 4

5.173.2.1262 CAN_FM1R_FBM5

```
#define CAN_FM1R_FBM5 ((uint16_t)0x0020)
```

Filter Init Mode bit 5

5.173.2.1263 CAN_FM1R_FBM6

```
#define CAN_FM1R_FBM6 ((uint16_t)0x0040)
```

Filter Init Mode bit 6

5.173.2.1264 CAN_FM1R_FBM7

```
#define CAN_FM1R_FBM7 ((uint16_t)0x0080)
```

Filter Init Mode bit 7

5.173.2.1265 CAN_FM1R_FBM8

```
#define CAN_FM1R_FBM8 ((uint16_t)0x0100)
```

Filter Init Mode bit 8

5.173.2.1266 CAN_FM1R_FBM9

```
#define CAN_FM1R_FBM9 ((uint16_t)0x0200)
```

Filter Init Mode bit 9

5.173.2.1267 CAN_FMR_INIT

```
#define CAN_FMR_INIT ((uint8_t)0x01)
```

Filter Init Mode

5.173.2.1268 CAN_FS1R_FSC

```
#define CAN_FS1R_FSC ((uint16_t)0x3FFF)
```

Filter Scale Configuration

5.173.2.1269 CAN_FS1R_FSC0

```
#define CAN_FS1R_FSC0 ((uint16_t)0x0001)
```

Filter Scale Configuration bit 0

5.173.2.1270 CAN_FS1R_FSC1

```
#define CAN_FS1R_FSC1 ((uint16_t)0x0002)
```

Filter Scale Configuration bit 1

5.173.2.1271 CAN_FS1R_FSC10

```
#define CAN_FS1R_FSC10 ((uint16_t)0x0400)
```

Filter Scale Configuration bit 10

5.173.2.1272 CAN_FS1R_FSC11

```
#define CAN_FS1R_FSC11 ((uint16_t)0x0800)
```

Filter Scale Configuration bit 11

5.173.2.1273 CAN_FS1R_FSC12

```
#define CAN_FS1R_FSC12 ((uint16_t)0x1000)
```

Filter Scale Configuration bit 12

5.173.2.1274 CAN_FS1R_FSC13

```
#define CAN_FS1R_FSC13 ((uint16_t)0x2000)
```

Filter Scale Configuration bit 13

5.173.2.1275 CAN_FS1R_FSC2

```
#define CAN_FS1R_FSC2 ((uint16_t)0x0004)
```

Filter Scale Configuration bit 2

5.173.2.1276 CAN_FS1R_FSC3

```
#define CAN_FS1R_FSC3 ((uint16_t)0x0008)
```

Filter Scale Configuration bit 3

5.173.2.1277 CAN_FS1R_FSC4

```
#define CAN_FS1R_FSC4 ((uint16_t)0x0010)
```

Filter Scale Configuration bit 4

5.173.2.1278 CAN_FS1R_FSC5

```
#define CAN_FS1R_FSC5 ((uint16_t)0x0020)
```

Filter Scale Configuration bit 5

5.173.2.1279 CAN_FS1R_FSC6

```
#define CAN_FS1R_FSC6 ((uint16_t)0x0040)
```

Filter Scale Configuration bit 6

5.173.2.1280 CAN_FS1R_FSC7

```
#define CAN_FS1R_FSC7 ((uint16_t)0x0080)
```

Filter Scale Configuration bit 7

5.173.2.1281 CAN_FS1R_FSC8

```
#define CAN_FS1R_FSC8 ((uint16_t)0x0100)
```

Filter Scale Configuration bit 8

5.173.2.1282 CAN_FS1R_FSC9

```
#define CAN_FS1R_FSC9 ((uint16_t)0x0200)
```

Filter Scale Configuration bit 9

5.173.2.1283 CAN_IER_BOFIE

```
#define CAN_IER_BOFIE ((uint32_t)0x00000400)
```

Bus-Off Interrupt Enable

5.173.2.1284 CAN_IER_EPVIE

```
#define CAN_IER_EPVIE ((uint32_t)0x00000200)
```

Error Passive Interrupt Enable

5.173.2.1285 CAN_IER_ERRIE

```
#define CAN_IER_ERRIE ((uint32_t)0x00008000)
```

Error Interrupt Enable

5.173.2.1286 CAN_IER_EWGIE

```
#define CAN_IER_EWGIE ((uint32_t)0x00000100)
```

Error Warning Interrupt Enable

5.173.2.1287 CAN_IER_FFIE0

```
#define CAN_IER_FFIE0 ((uint32_t)0x00000004)
```

FIFO Full Interrupt Enable

5.173.2.1288 CAN_IER_FFIE1

```
#define CAN_IER_FFIE1 ((uint32_t)0x00000020)
```

FIFO Full Interrupt Enable

5.173.2.1289 CAN_IER_FMPIE0

```
#define CAN_IER_FMPIE0 ((uint32_t)0x00000002)
```

FIFO Message Pending Interrupt Enable

5.173.2.1290 CAN_IER_FMPIE1

```
#define CAN_IER_FMPIE1 ((uint32_t)0x00000010)
```

FIFO Message Pending Interrupt Enable

5.173.2.1291 CAN_IER_FOVIE0

```
#define CAN_IER_FOVIE0 ((uint32_t)0x00000008)
```

FIFO Overrun Interrupt Enable

5.173.2.1292 CAN_IER_FOVIE1

```
#define CAN_IER_FOVIE1 ((uint32_t)0x00000040)
```

FIFO Overrun Interrupt Enable

5.173.2.1293 CAN_IER_LCIE

```
#define CAN_IER_LCIE ((uint32_t)0x00000800)
```

Last Error Code Interrupt Enable

5.173.2.1294 CAN_IER_SLKIE

```
#define CAN_IER_SLKIE ((uint32_t)0x00020000)
```

Sleep Interrupt Enable

5.173.2.1295 CAN_IER_TMEIE

```
#define CAN_IER_TMEIE ((uint32_t)0x00000001)
```

Transmit Mailbox Empty Interrupt Enable

5.173.2.1296 CAN_IER_WKUIE

```
#define CAN_IER_WKUIE ((uint32_t)0x00010000)
```

Wakeup Interrupt Enable

5.173.2.1297 CAN_MCR_ABOM

```
#define CAN_MCR_ABOM ((uint16_t)0x0040)
```

Automatic Bus-Off Management

5.173.2.1298 CAN_MCR_AWUM

```
#define CAN_MCR_AWUM ((uint16_t)0x0020)
```

Automatic Wakeup Mode

5.173.2.1299 CAN_MCR_INRQ

```
#define CAN_MCR_INRQ ((uint16_t)0x0001)
```

<CAN control and status registers Initialization Request

5.173.2.1300 CAN_MCR_NART

```
#define CAN_MCR_NART ((uint16_t)0x0010)
```

No Automatic Retransmission

5.173.2.1301 CAN_MCR_RESET

```
#define CAN_MCR_RESET ((uint16_t)0x8000)
```

bxCAN software master reset

5.173.2.1302 CAN_MCR_RFLM

```
#define CAN_MCR_RFLM ((uint16_t)0x0008)
```

Receive FIFO Locked Mode

5.173.2.1303 CAN_MCR_SLEEP

```
#define CAN_MCR_SLEEP ((uint16_t)0x0002)
```

Sleep Mode Request

5.173.2.1304 CAN_MCR_TTCM

```
#define CAN_MCR_TTCM ((uint16_t)0x0080)
```

Time Triggered Communication Mode

5.173.2.1305 CAN_MCR_TXFP

```
#define CAN_MCR_TXFP ((uint16_t)0x0004)
```

Transmit FIFO Priority

5.173.2.1306 CAN_MSR_ERRI

```
#define CAN_MSR_ERRI ((uint16_t)0x0004)
```

Error Interrupt

5.173.2.1307 CAN_MSR_INAK

```
#define CAN_MSR_INAK ((uint16_t)0x0001)
```

Initialization Acknowledge

5.173.2.1308 CAN_MSR_RX

```
#define CAN_MSR_RX ((uint16_t)0x0800)
```

CAN Rx Signal

5.173.2.1309 CAN_MSR_RXM

```
#define CAN_MSR_RXM ((uint16_t)0x0200)
```

Receive Mode

5.173.2.1310 CAN_MSR_SAMP

```
#define CAN_MSR_SAMP ((uint16_t)0x0400)
```

Last Sample Point

5.173.2.1311 CAN_MSR_SLAK

```
#define CAN_MSR_SLAK ((uint16_t)0x0002)
```

Sleep Acknowledge

5.173.2.1312 CAN_MSR_SLAKI

```
#define CAN_MSR_SLAKI ((uint16_t)0x0010)
```

Sleep Acknowledge Interrupt

5.173.2.1313 CAN_MSR_TXM

```
#define CAN_MSR_TXM ((uint16_t)0x0100)
```

Transmit Mode

5.173.2.1314 CAN_MSR_WKUI

```
#define CAN_MSR_WKUI ((uint16_t)0x0008)
```

Wakeup Interrupt

5.173.2.1315 CAN_RDH0R_DATA4

```
#define CAN_RDH0R_DATA4 ((uint32_t)0x000000FF)
```

Data byte 4

5.173.2.1316 CAN_RDH0R_DATA5

```
#define CAN_RDH0R_DATA5 ((uint32_t)0x0000FF00)
```

Data byte 5

5.173.2.1317 CAN_RDH0R_DATA6

```
#define CAN_RDH0R_DATA6 ((uint32_t)0x00FF0000)
```

Data byte 6

5.173.2.1318 CAN_RDH0R_DATA7

```
#define CAN_RDH0R_DATA7 ((uint32_t)0xFF000000)
```

Data byte 7

5.173.2.1319 CAN_RDH1R_DATA4

```
#define CAN_RDH1R_DATA4 ((uint32_t)0x000000FF)
```

Data byte 4

5.173.2.1320 CAN_RDH1R_DATA5

```
#define CAN_RDH1R_DATA5 ((uint32_t)0x0000FF00)
```

Data byte 5

5.173.2.1321 CAN_RDH1R_DATA6

```
#define CAN_RDH1R_DATA6 ((uint32_t)0x00FF0000)
```

Data byte 6

5.173.2.1322 CAN_RDH1R_DATA7

```
#define CAN_RDH1R_DATA7 ((uint32_t)0xFF000000)
```

Data byte 7 CAN filter registers

5.173.2.1323 CAN_RDL0R_DATA0

```
#define CAN_RDL0R_DATA0 ((uint32_t)0x000000FF)
```

Data byte 0

5.173.2.1324 CAN_RDL0R_DATA1

```
#define CAN_RDL0R_DATA1 ((uint32_t)0x0000FF00)
```

Data byte 1

5.173.2.1325 CAN_RDL0R_DATA2

```
#define CAN_RDL0R_DATA2 ((uint32_t)0x00FF0000)
```

Data byte 2

5.173.2.1326 CAN_RDL0R_DATA3

```
#define CAN_RDL0R_DATA3 ((uint32_t)0xFF000000)
```

Data byte 3

5.173.2.1327 CAN_RDL1R_DATA0

```
#define CAN_RDL1R_DATA0 ((uint32_t)0x000000FF)
```

Data byte 0

5.173.2.1328 CAN_RDL1R_DATA1

```
#define CAN_RDL1R_DATA1 ((uint32_t)0x0000FF00)
```

Data byte 1

5.173.2.1329 CAN_RDL1R_DATA2

```
#define CAN_RDL1R_DATA2 ((uint32_t)0x00FF0000)
```

Data byte 2

5.173.2.1330 CAN_RDL1R_DATA3

```
#define CAN_RDL1R_DATA3 ((uint32_t)0xFF000000)
```

Data byte 3

5.173.2.1331 CAN_RDT0R_DLC

```
#define CAN_RDT0R_DLC ((uint32_t)0x0000000F)
```

Data Length Code

5.173.2.1332 CAN_RDT0R_FMI

```
#define CAN_RDT0R_FMI ((uint32_t)0x0000FF00)
```

Filter Match Index

5.173.2.1333 CAN_RDT0R_TIME

```
#define CAN_RDT0R_TIME ((uint32_t)0xFFFF0000)
```

Message Time Stamp

5.173.2.1334 CAN_RDT1R_DLC

```
#define CAN_RDT1R_DLC ((uint32_t)0x0000000F)
```

Data Length Code

5.173.2.1335 CAN_RDT1R_FMI

```
#define CAN_RDT1R_FMI ((uint32_t)0x0000FF00)
```

Filter Match Index

5.173.2.1336 CAN_RDT1R_TIME

```
#define CAN_RDT1R_TIME ((uint32_t)0xFFFF0000)
```

Message Time Stamp

5.173.2.1337 CAN_RF0R_FMP0

```
#define CAN_RF0R_FMP0 ((uint8_t)0x03)
```

FIFO 0 Message Pending

5.173.2.1338 CAN_RF0R_FOVRO

```
#define CAN_RF0R_FOVRO ((uint8_t)0x10)
```

FIFO 0 Overrun

5.173.2.1339 CAN_RF0R_FULL0

```
#define CAN_RF0R_FULL0 ((uint8_t)0x08)
```

FIFO 0 Full

5.173.2.1340 CAN_RF0R_RFOM0

```
#define CAN_RF0R_RFOM0 ((uint8_t)0x20)
```

Release FIFO 0 Output Mailbox

5.173.2.1341 CAN_RF1R_FMP1

```
#define CAN_RF1R_FMP1 ((uint8_t)0x03)
```

FIFO 1 Message Pending

5.173.2.1342 CAN_RF1R_FOVR1

```
#define CAN_RF1R_FOVR1 ((uint8_t)0x10)
```

FIFO 1 Overrun

5.173.2.1343 CAN_RF1R_FULL1

```
#define CAN_RF1R_FULL1 ((uint8_t)0x08)
```

FIFO 1 Full

5.173.2.1344 CAN_RF1R_RFOM1

```
#define CAN_RF1R_RFOM1 ((uint8_t)0x20)
```

Release FIFO 1 Output Mailbox

5.173.2.1345 CAN_RI0R_EXID

```
#define CAN_RI0R_EXID ((uint32_t)0x001FFFF8)
```

Extended Identifier

5.173.2.1346 CAN_RI0R_IDE

```
#define CAN_RI0R_IDE ((uint32_t)0x00000004)
```

Identifier Extension

5.173.2.1347 CAN_RI0R_RTR

```
#define CAN_RI0R_RTR ((uint32_t)0x00000002)
```

Remote Transmission Request

5.173.2.1348 CAN_RI0R_STID

```
#define CAN_RI0R_STID ((uint32_t)0xFFE00000)
```

Standard Identifier or Extended Identifier

5.173.2.1349 CAN_RI1R_EXID

```
#define CAN_RI1R_EXID ((uint32_t)0x001FFFF8)
```

Extended identifier

5.173.2.1350 CAN_RI1R_IDE

```
#define CAN_RI1R_IDE ((uint32_t)0x00000004)
```

Identifier Extension

5.173.2.1351 CAN_RI1R_RTR

```
#define CAN_RI1R_RTR ((uint32_t)0x00000002)
```

Remote Transmission Request

5.173.2.1352 CAN_RI1R_STID

```
#define CAN_RI1R_STID ((uint32_t)0xFFE00000)
```

Standard Identifier or Extended Identifier

5.173.2.1353 CAN_TDH0R_DATA4

```
#define CAN_TDH0R_DATA4 ((uint32_t)0x000000FF)
```

Data byte 4

5.173.2.1354 CAN_TDH0R_DATA5

```
#define CAN_TDH0R_DATA5 ((uint32_t)0x0000FF00)
```

Data byte 5

5.173.2.1355 CAN_TDH0R_DATA6

```
#define CAN_TDH0R_DATA6 ((uint32_t)0x00FF0000)
```

Data byte 6

5.173.2.1356 CAN_TDH0R_DATA7

```
#define CAN_TDH0R_DATA7 ((uint32_t)0xFF000000)
```

Data byte 7

5.173.2.1357 CAN_TD1R_DATA4

```
#define CAN_TD1R_DATA4 ((uint32_t)0x000000FF)
```

Data byte 4

5.173.2.1358 CAN_TD1R_DATA5

```
#define CAN_TD1R_DATA5 ((uint32_t)0x0000FF00)
```

Data byte 5

5.173.2.1359 CAN_TD1R_DATA6

```
#define CAN_TD1R_DATA6 ((uint32_t)0x00FF0000)
```

Data byte 6

5.173.2.1360 CAN_TD1R_DATA7

```
#define CAN_TD1R_DATA7 ((uint32_t)0xFF000000)
```

Data byte 7

5.173.2.1361 CAN_TD2R_DATA4

```
#define CAN_TD2R_DATA4 ((uint32_t)0x000000FF)
```

Data byte 4

5.173.2.1362 CAN_TD2R_DATA5

```
#define CAN_TD2R_DATA5 ((uint32_t)0x0000FF00)
```

Data byte 5

5.173.2.1363 CAN_TD2R_DATA6

```
#define CAN_TD2R_DATA6 ((uint32_t)0x00FF0000)
```

Data byte 6

5.173.2.1364 CAN_TD2R_DATA7

```
#define CAN_TD2R_DATA7 ((uint32_t)0xFF000000)
```

Data byte 7

5.173.2.1365 CAN_TDL0R_DATA0

```
#define CAN_TDL0R_DATA0 ((uint32_t)0x000000FF)
```

Data byte 0

5.173.2.1366 CAN_TDL0R_DATA1

```
#define CAN_TDL0R_DATA1 ((uint32_t)0x0000FF00)
```

Data byte 1

5.173.2.1367 CAN_TDL0R_DATA2

```
#define CAN_TDL0R_DATA2 ((uint32_t)0x00FF0000)
```

Data byte 2

5.173.2.1368 CAN_TDL0R_DATA3

```
#define CAN_TDL0R_DATA3 ((uint32_t)0xFF000000)
```

Data byte 3

5.173.2.1369 CAN_TDL1R_DATA0

```
#define CAN_TDL1R_DATA0 ((uint32_t)0x000000FF)
```

Data byte 0

5.173.2.1370 CAN_TDL1R_DATA1

```
#define CAN_TDL1R_DATA1 ((uint32_t)0x0000FF00)
```

Data byte 1

5.173.2.1371 CAN_TDL1R_DATA2

```
#define CAN_TDL1R_DATA2 ((uint32_t)0x00FF0000)
```

Data byte 2

5.173.2.1372 CAN_TDL1R_DATA3

```
#define CAN_TDL1R_DATA3 ((uint32_t)0xFF000000)
```

Data byte 3

5.173.2.1373 CAN_TDL2R_DATA0

```
#define CAN_TDL2R_DATA0 ((uint32_t)0x000000FF)
```

Data byte 0

5.173.2.1374 CAN_TDL2R_DATA1

```
#define CAN_TDL2R_DATA1 ((uint32_t)0x0000FF00)
```

Data byte 1

5.173.2.1375 CAN_TDL2R_DATA2

```
#define CAN_TDL2R_DATA2 ((uint32_t)0x00FF0000)
```

Data byte 2

5.173.2.1376 CAN_TDL2R_DATA3

```
#define CAN_TDL2R_DATA3 ((uint32_t)0xFF000000)
```

Data byte 3

5.173.2.1377 CAN_TDT0R_DLC

```
#define CAN_TDT0R_DLC ((uint32_t)0x0000000F)
```

Data Length Code

5.173.2.1378 CAN_TDT0R_TGT

```
#define CAN_TDT0R_TGT ((uint32_t)0x000000100)
```

Transmit Global Time

5.173.2.1379 CAN_TDT0R_TIME

```
#define CAN_TDT0R_TIME ((uint32_t)0xFFFF0000)
```

Message Time Stamp

5.173.2.1380 CAN_TDT1R_DLC

```
#define CAN_TDT1R_DLC ((uint32_t)0x0000000F)
```

Data Length Code

5.173.2.1381 CAN_TDT1R_TGT

```
#define CAN_TDT1R_TGT ((uint32_t)0x00000100)
```

Transmit Global Time

5.173.2.1382 CAN_TDT1R_TIME

```
#define CAN_TDT1R_TIME ((uint32_t)0xFFFF0000)
```

Message Time Stamp

5.173.2.1383 CAN_TDT2R_DLC

```
#define CAN_TDT2R_DLC ((uint32_t)0x0000000F)
```

Data Length Code

5.173.2.1384 CAN_TDT2R_TGT

```
#define CAN_TDT2R_TGT ((uint32_t)0x00000100)
```

Transmit Global Time

5.173.2.1385 CAN_TDT2R_TIME

```
#define CAN_TDT2R_TIME ((uint32_t)0xFFFF0000)
```

Message Time Stamp

5.173.2.1386 CAN_TI0R_EXID

```
#define CAN_TI0R_EXID ((uint32_t)0x001FFFF8)
```

Extended Identifier

5.173.2.1387 CAN_TI0R_IDE

```
#define CAN_TI0R_IDE ((uint32_t)0x00000004)
```

Identifier Extension

5.173.2.1388 CAN_TI0R_RTR

```
#define CAN_TI0R_RTR ((uint32_t)0x00000002)
```

Remote Transmission Request

5.173.2.1389 CAN_TI0R_STID

```
#define CAN_TI0R_STID ((uint32_t)0xFFE00000)
```

Standard Identifier or Extended Identifier

5.173.2.1390 CAN_TI0R_TXRQ

```
#define CAN_TI0R_TXRQ ((uint32_t)0x00000001)
```

Transmit Mailbox Request

5.173.2.1391 CAN_TI1R_EXID

```
#define CAN_TI1R_EXID ((uint32_t)0x001FFFF8)
```

Extended Identifier

5.173.2.1392 CAN_TI1R_IDE

```
#define CAN_TI1R_IDE ((uint32_t)0x00000004)
```

Identifier Extension

5.173.2.1393 CAN_TI1R_RTR

```
#define CAN_TI1R_RTR ((uint32_t)0x00000002)
```

Remote Transmission Request

5.173.2.1394 CAN_TI1R_STID

```
#define CAN_TI1R_STID ((uint32_t)0xFFE00000)
```

Standard Identifier or Extended Identifier

5.173.2.1395 CAN_TI1R_TXRQ

```
#define CAN_TI1R_TXRQ ((uint32_t)0x00000001)
```

Transmit Mailbox Request

5.173.2.1396 CAN_TI2R_EXID

```
#define CAN_TI2R_EXID ((uint32_t)0x001FFFF8)
```

Extended identifier

5.173.2.1397 CAN_TI2R_IDE

```
#define CAN_TI2R_IDE ((uint32_t)0x00000004)
```

Identifier Extension

5.173.2.1398 CAN_TI2R_RTR

```
#define CAN_TI2R_RTR ((uint32_t)0x00000002)
```

Remote Transmission Request

5.173.2.1399 CAN_TI2R_STID

```
#define CAN_TI2R_STID ((uint32_t)0xFFE00000)
```

Standard Identifier or Extended Identifier

5.173.2.1400 CAN_TI2R_TXRQ

```
#define CAN_TI2R_TXRQ ((uint32_t)0x00000001)
```

Transmit Mailbox Request

5.173.2.1401 CAN_TSR_ABRQ0

```
#define CAN_TSR_ABRQ0 ((uint32_t)0x00000080)
```

Abort Request for Mailbox0

5.173.2.1402 CAN_TSR_ABRQ1

```
#define CAN_TSR_ABRQ1 ((uint32_t)0x00008000)
```

Abort Request for Mailbox 1

5.173.2.1403 CAN_TSR_ABRQ2

```
#define CAN_TSR_ABRQ2 ((uint32_t)0x00800000)
```

Abort Request for Mailbox 2

5.173.2.1404 CAN_TSR_ALST0

```
#define CAN_TSR_ALST0 ((uint32_t)0x00000004)
```

Arbitration Lost for Mailbox0

5.173.2.1405 CAN_TSR_ALST1

```
#define CAN_TSR_ALST1 ((uint32_t)0x00000400)
```

Arbitration Lost for Mailbox1

5.173.2.1406 CAN_TSR_ALST2

```
#define CAN_TSR_ALST2 ((uint32_t)0x00040000)
```

Arbitration Lost for mailbox 2

5.173.2.1407 CAN_TSR_CODE

```
#define CAN_TSR_CODE ((uint32_t)0x03000000)
```

Mailbox Code

5.173.2.1408 CAN_TSR_LOW

```
#define CAN_TSR_LOW ((uint32_t)0xE0000000)
```

LOW[2:0] bits

5.173.2.1409 CAN_TSR_LOW0

```
#define CAN_TSR_LOW0 ((uint32_t)0x20000000)
```

Lowest Priority Flag for Mailbox 0

5.173.2.1410 CAN_TSR_LOW1

```
#define CAN_TSR_LOW1 ((uint32_t)0x40000000)
```

Lowest Priority Flag for Mailbox 1

5.173.2.1411 CAN_TSR_LOW2

```
#define CAN_TSR_LOW2 ((uint32_t)0x80000000)
```

Lowest Priority Flag for Mailbox 2

5.173.2.1412 CAN_TSR_RQCP0

```
#define CAN_TSR_RQCP0 ((uint32_t)0x00000001)
```

Request Completed Mailbox0

5.173.2.1413 CAN_TSR_RQCP1

```
#define CAN_TSR_RQCP1 ((uint32_t)0x00000100)
```

Request Completed Mailbox1

5.173.2.1414 CAN_TSR_RQCP2

```
#define CAN_TSR_RQCP2 ((uint32_t)0x00010000)
```

Request Completed Mailbox2

5.173.2.1415 CAN_TSR_TERR0

```
#define CAN_TSR_TERR0 ((uint32_t)0x00000008)
```

Transmission Error of Mailbox0

5.173.2.1416 CAN_TSR_TERR1

```
#define CAN_TSR_TERR1 ((uint32_t)0x00000800)
```

Transmission Error of Mailbox1

5.173.2.1417 CAN_TSR_TERR2

```
#define CAN_TSR_TERR2 ((uint32_t)0x00080000)
```

Transmission Error of Mailbox 2

5.173.2.1418 CAN_TSR_TME

```
#define CAN_TSR_TME ((uint32_t)0x1C000000)
```

TME[2:0] bits

5.173.2.1419 CAN_TSR_TME0

```
#define CAN_TSR_TME0 ((uint32_t)0x04000000)
```

Transmit Mailbox 0 Empty

5.173.2.1420 CAN_TSR_TME1

```
#define CAN_TSR_TME1 ((uint32_t)0x08000000)
```

Transmit Mailbox 1 Empty

5.173.2.1421 CAN_TSR_TME2

```
#define CAN_TSR_TME2 ((uint32_t)0x10000000)
```

Transmit Mailbox 2 Empty

5.173.2.1422 CAN_TSR_TXOK0

```
#define CAN_TSR_TXOK0 ((uint32_t)0x00000002)
```

Transmission OK of Mailbox0

5.173.2.1423 CAN_TSR_TXOK1

```
#define CAN_TSR_TXOK1 ((uint32_t)0x00000200)
```

Transmission OK of Mailbox1

5.173.2.1424 CAN_TSR_TXOK2

```
#define CAN_TSR_TXOK2 ((uint32_t)0x00020000)
```

Transmission OK of Mailbox 2

5.173.2.1425 CRC_CR_RESET

```
#define CRC_CR_RESET ((uint8_t)0x01)
```

RESET bit

5.173.2.1426 CRC_DR_DR

```
#define CRC_DR_DR ((uint32_t)0xFFFFFFFF)
```

Data register bits

5.173.2.1427 CRC_IDR_IDR

```
#define CRC_IDR_IDR ((uint8_t)0xFF)
```

General-purpose 8-bit data register bits

5.173.2.1428 DAC_CR_BOFF1

```
#define DAC_CR_BOFF1 ((uint32_t)0x00000002)
```

DAC channel1 output buffer disable

5.173.2.1429 DAC_CR_BOFF2

```
#define DAC_CR_BOFF2 ((uint32_t)0x00020000)
```

DAC channel2 output buffer disable

5.173.2.1430 DAC_CR_DMAEN1

```
#define DAC_CR_DMAEN1 ((uint32_t)0x00001000)
```

DAC channel1 DMA enable

5.173.2.1431 DAC_CR_DMAEN2

```
#define DAC_CR_DMAEN2 ((uint32_t)0x10000000)
```

DAC channel2 DMA enabled

5.173.2.1432 DAC_CR_DMAUDRIE1

```
#define DAC_CR_DMAUDRIE1 ((uint32_t)0x00002000)
```

DAC channel1 DMA underrun interrupt enable

5.173.2.1433 DAC_CR_DMAUDRIE2

```
#define DAC_CR_DMAUDRIE2 ((uint32_t)0x20000000U)
```

DAC channel2 DMA underrun interrupt enable

5.173.2.1434 DAC_CR_EN1

```
#define DAC_CR_EN1 ((uint32_t)0x00000001)
```

DAC channel1 enable

5.173.2.1435 DAC_CR_EN2

```
#define DAC_CR_EN2 ((uint32_t)0x00010000)
```

DAC channel2 enable

5.173.2.1436 DAC_CR_MAMP1

```
#define DAC_CR_MAMP1 ((uint32_t)0x00000F00)
```

MAMP1 3:0

5.173.2.1437 DAC_CR_MAMP1_0

```
#define DAC_CR_MAMP1_0 ((uint32_t)0x00000100)
```

Bit 0

5.173.2.1438 DAC_CR_MAMP1_1

```
#define DAC_CR_MAMP1_1 ((uint32_t)0x00000200)
```

Bit 1

5.173.2.1439 DAC_CR_MAMP1_2

```
#define DAC_CR_MAMP1_2 ((uint32_t)0x00000400)
```

Bit 2

5.173.2.1440 DAC_CR_MAMP1_3

```
#define DAC_CR_MAMP1_3 ((uint32_t)0x00000800)
```

Bit 3

5.173.2.1441 DAC_CR_MAMP2

```
#define DAC_CR_MAMP2 ((uint32_t)0x0F000000)
```

MAMP2 3:0

5.173.2.1442 DAC_CR_MAMP2_0

```
#define DAC_CR_MAMP2_0 ((uint32_t)0x01000000)
```

Bit 0

5.173.2.1443 DAC_CR_MAMP2_1

```
#define DAC_CR_MAMP2_1 ((uint32_t)0x02000000)
```

Bit 1

5.173.2.1444 DAC_CR_MAMP2_2

```
#define DAC_CR_MAMP2_2 ((uint32_t)0x04000000)
```

Bit 2

5.173.2.1445 DAC_CR_MAMP2_3

```
#define DAC_CR_MAMP2_3 ((uint32_t)0x08000000)
```

Bit 3

5.173.2.1446 DAC_CR_TEN1

```
#define DAC_CR_TEN1 ((uint32_t)0x00000004)
```

DAC channel1 Trigger enable

5.173.2.1447 DAC_CR_TEN2

```
#define DAC_CR_TEN2 ((uint32_t)0x00040000)
```

DAC channel2 Trigger enable

5.173.2.1448 DAC_CR_TSEL1

```
#define DAC_CR_TSEL1 ((uint32_t)0x00000038)
```

TSEL1[2:0] (DAC channel1 Trigger selection)

5.173.2.1449 DAC_CR_TSEL1_0

```
#define DAC_CR_TSEL1_0 ((uint32_t)0x00000008)
```

Bit 0

5.173.2.1450 DAC_CR_TSEL1_1

```
#define DAC_CR_TSEL1_1 ((uint32_t)0x00000010)
```

Bit 1

5.173.2.1451 DAC_CR_TSEL1_2

```
#define DAC_CR_TSEL1_2 ((uint32_t)0x00000020)
```

Bit 2

5.173.2.1452 DAC_CR_TSEL2

```
#define DAC_CR_TSEL2 ((uint32_t)0x00380000)
```

TSEL2[2:0] (DAC channel2 Trigger selection)

5.173.2.1453 DAC_CR_TSEL2_0

```
#define DAC_CR_TSEL2_0 ((uint32_t)0x00080000)
```

Bit 0

5.173.2.1454 DAC_CR_TSEL2_1

```
#define DAC_CR_TSEL2_1 ((uint32_t)0x00100000)
```

Bit 1

5.173.2.1455 DAC_CR_TSEL2_2

```
#define DAC_CR_TSEL2_2 ((uint32_t)0x00200000)
```

Bit 2

5.173.2.1456 DAC_CR_WAVE1

```
#define DAC_CR_WAVE1 ((uint32_t)0x000000C0)
```

WAVE1 1:0

5.173.2.1457 DAC_CR_WAVE1_0

```
#define DAC_CR_WAVE1_0 ((uint32_t)0x00000040)
```

Bit 0

5.173.2.1458 DAC_CR_WAVE1_1

```
#define DAC_CR_WAVE1_1 ((uint32_t)0x00000080)
```

Bit 1

5.173.2.1459 DAC_CR_WAVE2

```
#define DAC_CR_WAVE2 ((uint32_t)0x00C00000)
```

WAVE2 1:0

5.173.2.1460 DAC_CR_WAVE2_0

```
#define DAC_CR_WAVE2_0 ((uint32_t)0x00400000)
```

Bit 0

5.173.2.1461 DAC_CR_WAVE2_1

```
#define DAC_CR_WAVE2_1 ((uint32_t)0x00800000)
```

Bit 1

5.173.2.1462 DAC_DHR12L1_DACC1DHR

```
#define DAC_DHR12L1_DACC1DHR ((uint16_t)0xFFFF0)
```

DAC channel1 12-bit Left aligned data

5.173.2.1463 DAC_DHR12L2_DACC2DHR

```
#define DAC_DHR12L2_DACC2DHR ((uint16_t)0xFFFF0)
```

DAC channel2 12-bit Left aligned data

5.173.2.1464 DAC_DHR12LD_DACC1DHR

```
#define DAC_DHR12LD_DACC1DHR ((uint32_t)0x0000FFFF0)
```

DAC channel1 12-bit Left aligned data

5.173.2.1465 DAC_DHR12LD_DACC2DHR

```
#define DAC_DHR12LD_DACC2DHR ((uint32_t)0xFFFF00000)
```

DAC channel2 12-bit Left aligned data

5.173.2.1466 DAC_DHR12R1_DACC1DHR

```
#define DAC_DHR12R1_DACC1DHR ((uint16_t)0x0FFF)
```

DAC channel1 12-bit Right aligned data

5.173.2.1467 DAC_DHR12R2_DACC2DHR

```
#define DAC_DHR12R2_DACC2DHR ((uint16_t)0x0FFF)
```

DAC channel2 12-bit Right aligned data

5.173.2.1468 DAC_DHR12RD_DACC1DHR

```
#define DAC_DHR12RD_DACC1DHR ((uint32_t)0x00000FFF)
```

DAC channel1 12-bit Right aligned data

5.173.2.1469 DAC_DHR12RD_DACC2DHR

```
#define DAC_DHR12RD_DACC2DHR ((uint32_t)0xFFFF0000)
```

DAC channel2 12-bit Right aligned data

5.173.2.1470 DAC_DHR8R1_DACC1DHR

```
#define DAC_DHR8R1_DACC1DHR ((uint8_t)0xFF)
```

DAC channel1 8-bit Right aligned data

5.173.2.1471 DAC_DHR8R2_DACC2DHR

```
#define DAC_DHR8R2_DACC2DHR ((uint8_t)0xFF)
```

DAC channel2 8-bit Right aligned data

5.173.2.1472 DAC_DHR8RD_DACC1DHR

```
#define DAC_DHR8RD_DACC1DHR ((uint16_t)0x00FF)
```

DAC channel1 8-bit Right aligned data

5.173.2.1473 DAC_DHR8RD_DACC2DHR

```
#define DAC_DHR8RD_DACC2DHR ((uint16_t)0xFF00)
```

DAC channel2 8-bit Right aligned data

5.173.2.1474 DAC_DOR1_DACC1DOR

```
#define DAC_DOR1_DACC1DOR ((uint16_t)0xFFFF)
```

DAC channel1 data output

5.173.2.1475 DAC_DOR2_DACC2DOR

```
#define DAC_DOR2_DACC2DOR ((uint16_t)0xFFFF)
```

DAC channel2 data output

5.173.2.1476 DAC_SR_DMAUDR1

```
#define DAC_SR_DMAUDR1 ((uint32_t)0x00002000)
```

DAC channel1 DMA underrun flag

5.173.2.1477 DAC_SR_DMAUDR2

```
#define DAC_SR_DMAUDR2 ((uint32_t)0x20000000)
```

DAC channel2 DMA underrun flag

5.173.2.1478 DAC_SWTRIGR_SWTRIG1

```
#define DAC_SWTRIGR_SWTRIG1 ((uint8_t)0x01)
```

DAC channel1 software trigger

5.173.2.1479 DAC_SWTRIGR_SWTRIG2

```
#define DAC_SWTRIGR_SWTRIG2 ((uint8_t)0x02)
```

DAC channel2 software trigger

5.173.2.1480 DBGMCU_CR_TRACE_MODE_0

```
#define DBGMCU_CR_TRACE_MODE_0 ((uint32_t)0x00000040)
```

Bit 0

5.173.2.1481 DBGMCU_CR_TRACE_MODE_1

```
#define DBGMCU_CR_TRACE_MODE_1 ((uint32_t)0x00000080)
```

Bit 1

5.173.2.1482 DFSDM_CHAWSCDR_AWFORD

```
#define DFSDM_CHAWSCDR_AWFORD ((uint32_t)0x00C00000)
```

AWFORD[1:0] Analog watchdog Sinc filter order on channel y

5.173.2.1483 DFSDM_CHAWSCDR_AWFORD_0

```
#define DFSDM_CHAWSCDR_AWFORD_0 ((uint32_t)0x00400000)
```

Analog watchdog Sinc filter order on channel y, Bit 0

5.173.2.1484 DFSDM_CHAWSCDR_AWFORD_1

```
#define DFSDM_CHAWSCDR_AWFORD_1 ((uint32_t)0x00800000)
```

Analog watchdog Sinc filter order on channel y, Bit 1

5.173.2.1485 DFSDM_CHAWSCDR_AWFOSR

```
#define DFSDM_CHAWSCDR_AWFOSR ((uint32_t)0x001F0000)
```

AWFOSR[4:0] Analog watchdog filter oversampling ratio on channel y

5.173.2.1486 DFSDM_CHAWSCDR_BKSCD

```
#define DFSDM_CHAWSCDR_BKSCD ((uint32_t)0x0000F000)
```

BKSCD[3:0] Break signal assignment for short circuit detector on channel y

5.173.2.1487 DFSDM_CHAWSCDR_SCDT

```
#define DFSDM_CHAWSCDR_SCDT ((uint32_t)0x000000FF)
```

SCDT[7:0] Short circuit detector threshold for channel y

5.173.2.1488 DFSDM_CHCGR1_CHEN

```
#define DFSDM_CHCGR1_CHEN ((uint32_t)0x00000080)
```

Channel y enable

5.173.2.1489 DFSDM_CHCGR1_CHINSEL

```
#define DFSDM_CHCGR1_CHINSEL ((uint32_t)0x00000100)
```

Serial inputs selection for channel y

5.173.2.1490 DFSDM_CHCGR1_CKABEN

```
#define DFSDM_CHCGR1_CKABEN ((uint32_t)0x00000040)
```

Clock absence detector enable on channel y

5.173.2.1491 DFSDM_CHCGR1_CKOUTDIV

```
#define DFSDM_CHCGR1_CKOUTDIV ((uint32_t)0x00FF0000)
```

CKOUTDIV[7:0] output serial clock divider

5.173.2.1492 DFSDM_CHCGR1_CKOUTSRC

```
#define DFSDM_CHCGR1_CKOUTSRC ((uint32_t)0x40000000)
```

Output serial clock source selection

5.173.2.1493 DFSDM_CHCFGR1_DATMPX

```
#define DFSDM_CHCFGR1_DATMPX ((uint32_t)0x00003000)
```

DATMPX[1:0] Input data multiplexer for channel y

5.173.2.1494 DFSDM_CHCFGR1_DATMPX_0

```
#define DFSDM_CHCFGR1_DATMPX_0 ((uint32_t)0x00001000)
```

Input data multiplexer for channel y, Bit 0

5.173.2.1495 DFSDM_CHCFGR1_DATMPX_1

```
#define DFSDM_CHCFGR1_DATMPX_1 ((uint32_t)0x00002000)
```

Input data multiplexer for channel y, Bit 1

5.173.2.1496 DFSDM_CHCFGR1_DATPACK

```
#define DFSDM_CHCFGR1_DATPACK ((uint32_t)0x0000C000)
```

DATPACK[1:0] Data packing mode

5.173.2.1497 DFSDM_CHCFGR1_DATPACK_0

```
#define DFSDM_CHCFGR1_DATPACK_0 ((uint32_t)0x00004000)
```

Data packing mode, Bit 0

5.173.2.1498 DFSDM_CHCFGR1_DATPACK_1

```
#define DFSDM_CHCFGR1_DATPACK_1 ((uint32_t)0x00008000)
```

Data packing mode, Bit 1

5.173.2.1499 DFSDM_CHCFGR1_DFSDMEN

```
#define DFSDM_CHCFGR1_DFSDMEN ((uint32_t)0x80000000)
```

Global enable for DFSDM interface

5.173.2.1500 DFSDM_CHCFGR1_SCDEN

```
#define DFSDM_CHCFGR1_SCDEN ((uint32_t)0x00000020)
```

Short circuit detector enable on channel y

5.173.2.1501 DFSDM_CHCFGR1_SITP

```
#define DFSDM_CHCFGR1_SITP ((uint32_t)0x00000003)
```

SITP[1:0] Serial interface type for channel y

5.173.2.1502 DFSDM_CHCFGR1_SITP_0

```
#define DFSDM_CHCFGR1_SITP_0 ((uint32_t)0x00000001)
```

Serial interface type for channel y, Bit 0

5.173.2.1503 DFSDM_CHCFGR1_SITP_1

```
#define DFSDM_CHCFGR1_SITP_1 ((uint32_t)0x00000002)
```

Serial interface type for channel y, Bit 1

5.173.2.1504 DFSDM_CHCFGR1_SPICKSEL

```
#define DFSDM_CHCFGR1_SPICKSEL ((uint32_t)0x0000000C)
```

SPICKSEL[1:0] SPI clock select for channel y

5.173.2.1505 DFSDM_CHCFGR1_SPICKSEL_0

```
#define DFSDM_CHCFGR1_SPICKSEL_0 ((uint32_t)0x00000004)
```

SPI clock select for channel y, Bit 0

5.173.2.1506 DFSDM_CHCFGR1_SPICKSEL_1

```
#define DFSDM_CHCFGR1_SPICKSEL_1 ((uint32_t)0x00000008)
```

SPI clock select for channel y, Bit 1

5.173.2.1507 DFSDM_CHCFGR2_DTRBS

```
#define DFSDM_CHCFGR2_DTRBS ((uint32_t)0x000000F8)
```

DTRBS[4:0] Data right bit-shift for channel y

5.173.2.1508 DFSDM_CHCFGR2_OFFSET

```
#define DFSDM_CHCFGR2_OFFSET ((uint32_t)0xFFFFFFF00)
```

OFFSET[23:0] 24-bit calibration offset for channel y

5.173.2.1509 DFSDM_CHDATINR_INDAT0

```
#define DFSDM_CHDATINR_INDAT0 ((uint32_t)0x0000FFFF)
```

INDAT0[31:16] Input data for channel y or channel (y+1)

5.173.2.1510 DFSDM_CHDATINR_INDAT1

```
#define DFSDM_CHDATINR_INDAT1 ((uint32_t)0xFFFF0000)
```

INDAT0[15:0] Input data for channel y

5.173.2.1511 DFSDM_CHWDATR_WDATA

```
#define DFSDM_CHWDATR_WDATA ((uint32_t)0x0000FFFF)
```

WDATA[15:0] Input channel y watchdog data

5.173.2.1512 DFSDM_FLTAWCFR_CLRAWHTF

```
#define DFSDM_FLTAWCFR_CLRAWHTF ((uint32_t)0x00000F00)
```

CLRAWHTF[15:8] Clear the Analog watchdog high threshold flag

5.173.2.1513 DFSDM_FLTAWCFR_CLRAWLTF

```
#define DFSDM_FLTAWCFR_CLRAWLTF ((uint32_t)0x0000000F)
```

CLRAWLTF[7:0] Clear the Analog watchdog low threshold flag

5.173.2.1514 DFSDM_FLTAWHTR_AWHT

```
#define DFSDM_FLTAWHTR_AWHT ((uint32_t)0xFFFFFFF0)
```

AWHT[23:0] Analog watchdog high threshold

5.173.2.1515 DFSDM_FLTAWHTR_BKAWH

```
#define DFSDM_FLTAWHTR_BKAWH ((uint32_t)0x0000000F)
```

BKAWH[3:0] Break signal assignment to analog watchdog high threshold event

5.173.2.1516 DFSDM_FLTAWLTR_AWL

```
#define DFSDM_FLTAWLTR_AWL ((uint32_t)0xFFFFFFF0)
```

AWLT[23:0] Analog watchdog low threshold

5.173.2.1517 DFSDM_FLTAWLTR_BKAWL

```
#define DFSDM_FLTAWLTR_BKAWL ((uint32_t)0x0000000F)
```

BKAWL[3:0] Break signal assignment to analog watchdog low threshold event

5.173.2.1518 DFSDM_FLTAWSR_AWHTF

```
#define DFSDM_FLTAWSR_AWHTF ((uint32_t)0x00000F00)
```

AWHTF[15:8] Analog watchdog high threshold error on given channels

5.173.2.1519 DFSDM_FLTAWSR_AWLTF

```
#define DFSDM_FLTAWSR_AWLTF ((uint32_t)0x0000000F)
```

AWLTF[7:0] Analog watchdog low threshold error on given channels

5.173.2.1520 DFSDM_FLTCNVTIMR_CNVCNT

```
#define DFSDM_FLTCNVTIMR_CNVCNT ((uint32_t)0xFFFFFFFF0)
```

CNVCNT[27:0]: 28-bit timer counting conversion time

5.173.2.1521 DFSDM_FLTCR1_AWFSEL

```
#define DFSDM_FLTCR1_AWFSEL ((uint32_t)0x40000000)
```

Analog watchdog fast mode select

5.173.2.1522 DFSDM_FLTCR1_DFEN

```
#define DFSDM_FLTCR1_DFEN ((uint32_t)0x00000001)
```

DFSDM enable

5.173.2.1523 DFSDM_FLTCR1_FAST

```
#define DFSDM_FLTCR1_FAST ((uint32_t)0x20000000)
```

Fast conversion mode selection

5.173.2.1524 DFSDM_FLTCR1_JDMAEN

```
#define DFSDM_FLTCR1_JDMAEN ((uint32_t)0x00000020)
```

DMA channel enabled to read data for the injected channel group

5.173.2.1525 DFSDM_FLTCR1_JEXTEN

```
#define DFSDM_FLTCR1_JEXTEN ((uint32_t)0x00006000)
```

JEXTEN[1:0] Trigger enable and trigger edge selection for injected conversions

5.173.2.1526 DFSDM_FLTCR1_JEXTEN_0

```
#define DFSDM_FLTCR1_JEXTEN_0 ((uint32_t)0x00002000)
```

Trigger enable and trigger edge selection for injected conversions, Bit 0

5.173.2.1527 DFSDM_FLTCR1_JEXTEN_1

```
#define DFSDM_FLTCR1_JEXTEN_1 ((uint32_t)0x00004000)
```

Trigger enable and trigger edge selection for injected conversions, Bit 1

5.173.2.1528 DFSDM_FLTCR1_JEXTSEL

```
#define DFSDM_FLTCR1_JEXTSEL ((uint32_t)0x00000700)
```

JEXTSEL[2:0]Trigger signal selection for launching injected conversions

5.173.2.1529 DFSDM_FLTCR1_JEXTSEL_0

```
#define DFSDM_FLTCR1_JEXTSEL_0 ((uint32_t)0x00000100)
```

Trigger signal selection for launching injected conversions, Bit 0

5.173.2.1530 DFSDM_FLTCR1_JEXTSEL_1

```
#define DFSDM_FLTCR1_JEXTSEL_1 ((uint32_t)0x00000200)
```

Trigger signal selection for launching injected conversions, Bit 1

5.173.2.1531 DFSDM_FLTCR1_JEXTSEL_2

```
#define DFSDM_FLTCR1_JEXTSEL_2 ((uint32_t)0x00000400)
```

Trigger signal selection for launching injected conversions, Bit 2

5.173.2.1532 DFSDM_FLTCR1_JSCAN

```
#define DFSDM_FLTCR1_JSCAN ((uint32_t)0x00000010)
```

Scanning conversion in continuous mode selection for injected conversions

5.173.2.1533 DFSDM_FLTCR1_JSWSTART

```
#define DFSDM_FLTCR1_JSWSTART ((uint32_t)0x00000002)
```

Start the conversion of the injected group of channels

5.173.2.1534 DFSDM_FLTCR1_JSYNC

```
#define DFSDM_FLTCR1_JSYNC ((uint32_t)0x00000008)
```

Launch an injected conversion synchronously with DFSDMx JSWSTART trigger

5.173.2.1535 DFSDM_FLTCR1_RCH

```
#define DFSDM_FLTCR1_RCH ((uint32_t)0x07000000)
```

RCH[2:0] Regular channel selection

5.173.2.1536 DFSDM_FLTCR1_RCONT

```
#define DFSDM_FLTCR1_RCONT ((uint32_t)0x00040000)
```

Continuous mode selection for regular conversions

5.173.2.1537 DFSDM_FLTCR1_RDMAEN

```
#define DFSDM_FLTCR1_RDMAEN ((uint32_t)0x00200000)
```

DMA channel enabled to read data for the regular conversion

5.173.2.1538 DFSDM_FLTCR1_RSWSTART

```
#define DFSDM_FLTCR1_RSWSTART ((uint32_t)0x00020000)
```

Software start of a conversion on the regular channel

5.173.2.1539 DFSDM_FLTCR1_RSYNC

```
#define DFSDM_FLTCR1_RSYNC ((uint32_t)0x00080000)
```

Launch regular conversion synchronously with DFSDMx

5.173.2.1540 DFSDM_FLTCR2_AWDCH

```
#define DFSDM_FLTCR2_AWDCH ((uint32_t)0x000F0000)
```

AWDCH[7:0] Analog watchdog channel selection

5.173.2.1541 DFSDM_FLTCR2_AWDIE

```
#define DFSDM_FLTCR2_AWDIE ((uint32_t)0x00000010)
```

Analog watchdog interrupt enable

5.173.2.1542 DFSDM_FLTCR2_CKABIE

```
#define DFSDM_FLTCR2_CKABIE ((uint32_t)0x00000040)
```

Clock absence interrupt enable

5.173.2.1543 DFSDM_FLTCR2_EXCH

```
#define DFSDM_FLTCR2_EXCH ((uint32_t)0x00000F00)
```

EXCH[7:0] Extreme detector channel selection

5.173.2.1544 DFSDM_FLTCR2_JEOCIE

```
#define DFSDM_FLTCR2_JEOCIE ((uint32_t)0x00000001)
```

Injected end of conversion interrupt enable

5.173.2.1545 DFSDM_FLTCR2_JOVRIE

```
#define DFSDM_FLTCR2_JOVRIE ((uint32_t)0x00000004)
```

Injected data overrun interrupt enable

5.173.2.1546 DFSDM_FLTCR2_REOCIE

```
#define DFSDM_FLTCR2_REOCIE ((uint32_t)0x00000002)
```

Regular end of conversion interrupt enable

5.173.2.1547 DFSDM_FLTCR2_ROVRIE

```
#define DFSDM_FLTCR2_ROVRIE ((uint32_t)0x00000008)
```

Regular data overrun interrupt enable

5.173.2.1548 DFSDM_FLTCR2_SCDIE

```
#define DFSDM_FLTCR2_SCDIE ((uint32_t)0x00000020)
```

Short circuit detector interrupt enable

5.173.2.1549 DFSDM_FLTEXMAX_EXMAX

```
#define DFSDM_FLTEXMAX_EXMAX ((uint32_t)0xFFFFFFF00)
```

EXMAX[23:0] Extreme detector maximum value

5.173.2.1550 DFSDM_FLTEXMAX_EXMAXCH

```
#define DFSDM_FLTEXMAX_EXMAXCH ((uint32_t)0x00000007)
```

EXMAXCH[2:0] Extreme detector maximum data channel

5.173.2.1551 DFSDM_FLTEXMIN_EXMIN

```
#define DFSDM_FLTEXMIN_EXMIN ((uint32_t)0xFFFFFFF00)
```

EXMIN[23:0] Extreme detector minimum value

5.173.2.1552 DFSDM_FLTEXMIN_EXMINCH

```
#define DFSDM_FLTEXMIN_EXMINCH ((uint32_t)0x00000007)
```

EXMINCH[2:0] Extreme detector minimum data channel

5.173.2.1553 DFSDM_FLTFCR_FORD

```
#define DFSDM_FLTFCR_FORD ((uint32_t)0xE0000000)
```

FORD[2:0] Sinc filter order

5.173.2.1554 DFSDM_FLTFCR_FORD_0

```
#define DFSDM_FLTFCR_FORD_0 ((uint32_t)0x20000000)
```

Sinc filter order, Bit 0

5.173.2.1555 DFSDM_FLTFCR_FORD_1

```
#define DFSDM_FLTFCR_FORD_1 ((uint32_t)0x40000000)
```

Sinc filter order, Bit 1

5.173.2.1556 DFSDM_FLTFCR_FORD_2

```
#define DFSDM_FLTFCR_FORD_2 ((uint32_t)0x80000000)
```

Sinc filter order, Bit 2

5.173.2.1557 DFSDM_FLTFCR_FOSR

```
#define DFSDM_FLTFCR_FOSR ((uint32_t)0x03FF0000)
```

FOSR[9:0] Sinc filter oversampling ratio (decimation rate)

5.173.2.1558 DFSDM_FLTFCR_IOSR

```
#define DFSDM_FLTFCR_IOSR ((uint32_t)0x000000FF)
```

IOSR[7:0] Integrator oversampling ratio (averaging length)

5.173.2.1559 DFSDM_FLTICR_CLRCKABF

```
#define DFSDM_FLTICR_CLRCKABF ((uint32_t)0x000F0000)
```

CLRCKABF[7:0] Clear the clock absence flag

5.173.2.1560 DFSDM_FLTICR_CLRJOVRF

```
#define DFSDM_FLTICR_CLRJOVRF ((uint32_t)0x00000004)
```

Clear the injected conversion overrun flag

5.173.2.1561 DFSDM_FLTICR_CLRROVRF

```
#define DFSDM_FLTICR_CLRROVRF ((uint32_t)0x00000008)
```

Clear the regular conversion overrun flag

5.173.2.1562 DFSDM_FLTICR_CLRSCTSDF

```
#define DFSDM_FLTICR_CLRSCTSDF ((uint32_t)0x0F000000)
```

CLRSCTSDF[7:0] Clear the short circuit detector flag

5.173.2.1563 DFSDM_FLTISR_AWDF

```
#define DFSDM_FLTISR_AWDF ((uint32_t)0x00000010)
```

Analog watchdog

5.173.2.1564 DFSDM_FLTISR_CKABF

```
#define DFSDM_FLTISR_CKABF ((uint32_t)0x000F0000)
```

CKABF[7:0] Clock absence flag

5.173.2.1565 DFSDM_FLTISR_JCIP

```
#define DFSDM_FLTISR_JCIP ((uint32_t)0x00002000)
```

Injected conversion in progress status

5.173.2.1566 DFSDM_FLTISR_JEOCF

```
#define DFSDM_FLTISR_JEOCF ((uint32_t)0x00000001)
```

End of injected conversion flag

5.173.2.1567 DFSDM_FLTISR_JOVRF

```
#define DFSDM_FLTISR_JOVRF ((uint32_t)0x00000004)
```

Injected conversion overrun flag

5.173.2.1568 DFSDM_FLTISR_RCIP

```
#define DFSDM_FLTISR_RCIP ((uint32_t)0x00004000)
```

Regular conversion in progress status

5.173.2.1569 DFSDM_FLTISR_REOCF

```
#define DFSDM_FLTISR_REOCF ((uint32_t)0x00000002)
```

End of regular conversion flag

5.173.2.1570 DFSDM_FLTISR_ROVRF

```
#define DFSDM_FLTISR_ROVRF ((uint32_t)0x00000008)
```

Regular conversion overrun flag

5.173.2.1571 DFSDM_FLTISR_SCDF

```
#define DFSDM_FLTISR_SCDF ((uint32_t)0x0F000000)
```

SCDF[7:0] Short circuit detector flag

5.173.2.1572 DFSDM_FLTJCHGR_JCHG

```
#define DFSDM_FLTJCHGR_JCHG ((uint32_t)0x000000FF)
```

JCHG[7:0] Injected channel group selection

5.173.2.1573 DFSDM_FLTJDATAR_JDATA

```
#define DFSDM_FLTJDATAR_JDATA ((uint32_t)0xFFFFFFF00)
```

JDATA[23:0] Injected group conversion data

5.173.2.1574 DFSDM_FLTJDATAR_JDATACH

```
#define DFSDM_FLTJDATAR_JDATACH ((uint32_t)0x00000007)
```

JDATACH[2:0] Injected channel most recently converted

5.173.2.1575 DFSDM_FLTRDATAR_RDATA

```
#define DFSDM_FLTRDATAR_RDATA ((uint32_t)0xFFFFFFF00)
```

RDATA[23:0] Regular channel conversion data

5.173.2.1576 DFSDM_FLTRDATAR_RDATACH

```
#define DFSDM_FLTRDATAR_RDATACH ((uint32_t)0x00000007)
```

RDATACH[2:0] Regular channel most recently converted

5.173.2.1577 DFSDM_FLTRDATAR_RPEND

```
#define DFSDM_FLTRDATAR_RPEND ((uint32_t)0x00000010)
```

RPEND Regular channel pending data

5.173.2.1578 DMA2D_AMTCR_DT

```
#define DMA2D_AMTCR_DT ((uint32_t)0x0000FF00)
```

Dead Time

5.173.2.1579 DMA2D_AMTCR_EN

```
#define DMA2D_AMTCR_EN ((uint32_t)0x00000001)
```

Enable

5.173.2.1580 DMA2D_BGCMAR_MA

```
#define DMA2D_BGCMAR_MA ((uint32_t)0xFFFFFFFF)
```

Memory Address

5.173.2.1581 DMA2D_BGCOLR_BLUE

```
#define DMA2D_BGCOLR_BLUE ((uint32_t)0x000000FF)
```

Blue Value

5.173.2.1582 DMA2D_BGCOLR_GREEN

```
#define DMA2D_BGCOLR_GREEN ((uint32_t)0x0000FF00)
```

Green Value

5.173.2.1583 DMA2D_BGCOLR_RED

```
#define DMA2D_BGCOLR_RED ((uint32_t)0x00FF0000)
```

Red Value

5.173.2.1584 DMA2D_BGMAR_MA

```
#define DMA2D_BGMAR_MA ((uint32_t)0xFFFFFFFF)
```

Memory Address

5.173.2.1585 DMA2D_BGOR_LO

```
#define DMA2D_BGOR_LO ((uint32_t)0x00003FFF)
```

Line Offset

5.173.2.1586 DMA2D_BGPFCCR_ALPHA

```
#define DMA2D_BGPFCCR_ALPHA ((uint32_t)0xFF000000)
```

Alpha value

5.173.2.1587 DMA2D_BGPFCCR_AM

```
#define DMA2D_BGPFCCR_AM ((uint32_t)0x00030000)
```

Alpha mode AM[1:0]

5.173.2.1588 DMA2D_BGPFCCR_AM_0

```
#define DMA2D_BGPFCCR_AM_0 ((uint32_t)0x00010000)
```

Alpha mode AM bit 0

5.173.2.1589 DMA2D_BGPFCCR_AM_1

```
#define DMA2D_BGPFCCR_AM_1 ((uint32_t)0x00020000)
```

Alpha mode AM bit 1

5.173.2.1590 DMA2D_BGPFCCR_CCM

```
#define DMA2D_BGPFCCR_CCM ((uint32_t)0x00000010)
```

CLUT Color mode

5.173.2.1591 DMA2D_BGPFCCR_CM

```
#define DMA2D_BGPFCCR_CM ((uint32_t)0x0000000F)
```

Input color mode CM[3:0]

5.173.2.1592 DMA2D_BGPFCCR_CM_0

```
#define DMA2D_BGPFCCR_CM_0 ((uint32_t)0x00000001)
```

Input color mode CM bit 0

5.173.2.1593 DMA2D_BGPFCCR_CM_1

```
#define DMA2D_BGPFCCR_CM_1 ((uint32_t)0x00000002)
```

Input color mode CM bit 1

5.173.2.1594 DMA2D_BGPFCCR_CM_2

```
#define DMA2D_BGPFCCR_CM_2 ((uint32_t)0x00000004)
```

Input color mode CM bit 2

5.173.2.1595 DMA2D_BGPFCCR_CS

```
#define DMA2D_BGPFCCR_CS ((uint32_t)0x0000FF00)
```

CLUT size

5.173.2.1596 DMA2D_BGPFCCR_START

```
#define DMA2D_BGPFCCR_START ((uint32_t)0x00000020)
```

Start

5.173.2.1597 DMA2D_CR_ABORT

```
#define DMA2D_CR_ABORT ((uint32_t)0x00000004)
```

Abort transfer

5.173.2.1598 DMA2D_CR_CAEIE

```
#define DMA2D_CR_CAEIE ((uint32_t)0x00000800)
```

CLUT Access Error Interrupt Enable

5.173.2.1599 DMA2D_CR_CEIE

```
#define DMA2D_CR_CEIE ((uint32_t)0x00002000)
```

Configuration Error Interrupt Enable

5.173.2.1600 DMA2D_CR_CTCIE

```
#define DMA2D_CR_CTCIE ((uint32_t)0x00001000)
```

CLUT Transfer Complete Interrupt Enable

5.173.2.1601 DMA2D_CR_MODE

```
#define DMA2D_CR_MODE ((uint32_t)0x00030000)
```

DMA2D Mode

5.173.2.1602 DMA2D_CR_START

```
#define DMA2D_CR_START ((uint32_t)0x00000001)
```

Start transfer

5.173.2.1603 DMA2D_CR_SUSP

```
#define DMA2D_CR_SUSP ((uint32_t)0x00000002)
```

Suspend transfer

5.173.2.1604 DMA2D_CR_TCIE

```
#define DMA2D_CR_TCIE ((uint32_t)0x00000200)
```

Transfer Complete Interrupt Enable

5.173.2.1605 DMA2D_CR_TEIE

```
#define DMA2D_CR_TEIE ((uint32_t)0x00000100)
```

Transfer Error Interrupt Enable

5.173.2.1606 DMA2D_CR_TWIE

```
#define DMA2D_CR_TWIE ((uint32_t)0x00000400)
```

Transfer Watermark Interrupt Enable

5.173.2.1607 DMA2D_FGCMAR_MA

```
#define DMA2D_FGCMAR_MA ((uint32_t)0xFFFFFFFF)
```

Memory Address

5.173.2.1608 DMA2D_FGCOLR_BLUE

```
#define DMA2D_FGCOLR_BLUE ((uint32_t)0x000000FF)
```

Blue Value

5.173.2.1609 DMA2D_FGCOLR_GREEN

```
#define DMA2D_FGCOLR_GREEN ((uint32_t)0x0000FF00)
```

Green Value

5.173.2.1610 DMA2D_FGCOLR_RED

```
#define DMA2D_FGCOLR_RED ((uint32_t)0x00FF0000)
```

Red Value

5.173.2.1611 DMA2D_FGMAR_MA

```
#define DMA2D_FGMAR_MA ((uint32_t)0xFFFFFFFF)
```

Memory Address

5.173.2.1612 DMA2D_FGOR_LO

```
#define DMA2D_FGOR_LO ((uint32_t)0x00003FFF)
```

Line Offset

5.173.2.1613 DMA2D_FGPFCCR_ALPHA

```
#define DMA2D_FGPFCCR_ALPHA ((uint32_t)0xFF000000)
```

Alpha value

5.173.2.1614 DMA2D_FGPFCCR_AM

```
#define DMA2D_FGPFCCR_AM ((uint32_t)0x00030000)
```

Alpha mode AM[1:0]

5.173.2.1615 DMA2D_FGPFCCR_AM_0

```
#define DMA2D_FGPFCCR_AM_0 ((uint32_t)0x00010000)
```

Alpha mode AM bit 0

5.173.2.1616 DMA2D_FGPFCCR_AM_1

```
#define DMA2D_FGPFCCR_AM_1 ((uint32_t)0x00020000)
```

Alpha mode AM bit 1

5.173.2.1617 DMA2D_FGPFCCR_CCM

```
#define DMA2D_FGPFCCR_CCM ((uint32_t)0x00000010)
```

CLUT Color mode

5.173.2.1618 DMA2D_FGPFCCR_CM

```
#define DMA2D_FGPFCCR_CM ((uint32_t)0x0000000F)
```

Input color mode CM[3:0]

5.173.2.1619 DMA2D_FGPFCCR_CM_0

```
#define DMA2D_FGPFCCR_CM_0 ((uint32_t)0x00000001)
```

Input color mode CM bit 0

5.173.2.1620 DMA2D_FGPFCCR_CM_1

```
#define DMA2D_FGPFCCR_CM_1 ((uint32_t)0x00000002)
```

Input color mode CM bit 1

5.173.2.1621 DMA2D_FGPFCCR_CM_2

```
#define DMA2D_FGPFCCR_CM_2 ((uint32_t)0x00000004)
```

Input color mode CM bit 2

5.173.2.1622 DMA2D_FGPFCCR_CM_3 [1/2]

```
#define DMA2D_FGPFCCR_CM_3 ((uint32_t)0x00000008)
```

Input color mode CM bit 3

5.173.2.1623 DMA2D_FGPFCCR_CM_3 [2/2]

```
#define DMA2D_FGPFCCR_CM_3 ((uint32_t)0x00000008)
```

Input color mode CM bit 3

5.173.2.1624 DMA2D_FGPFCCR_CS

```
#define DMA2D_FGPFCCR_CS ((uint32_t)0x0000FF00)
```

CLUT size

5.173.2.1625 DMA2D_FGPFCCR_START

```
#define DMA2D_FGPFCCR_START ((uint32_t)0x00000020)
```

Start

5.173.2.1626 DMA2D_IFCR_CAECAF

```
#define DMA2D_IFCR_CAECAF ((uint32_t)0x00000008)
```

Clears CLUT Access Error Interrupt Flag

5.173.2.1627 DMA2D_IFCR_CCEIF

```
#define DMA2D_IFCR_CCEIF ((uint32_t)0x00000020)
```

Clears Configuration Error Interrupt Flag

5.173.2.1628 DMA2D_IFCR_CCTCIF

```
#define DMA2D_IFCR_CCTCIF ((uint32_t)0x00000010)
```

Clears CLUT Transfer Complete Interrupt Flag

5.173.2.1629 DMA2D_IFCR_CTCIF

```
#define DMA2D_IFCR_CTCIF ((uint32_t)0x00000002)
```

Clears Transfer Complete Interrupt Flag

5.173.2.1630 DMA2D_IFCR_CTEIF

```
#define DMA2D_IFCR_CTEIF ((uint32_t)0x00000001)
```

Clears Transfer Error Interrupt Flag

5.173.2.1631 DMA2D_IFCR_CTWIF

```
#define DMA2D_IFCR_CTWIF ((uint32_t)0x00000004)
```

Clears Transfer Watermark Interrupt Flag

5.173.2.1632 DMA2D_IFSR_CCAEIF

```
#define DMA2D_IFSR_CCAEIF DMA2D_IFCR_CAECIF
```

Clears CLUT Access Error Interrupt Flag

5.173.2.1633 DMA2D_IFSR_CCEIF

```
#define DMA2D_IFSR_CCEIF DMA2D_IFCR_CCEIF
```

Clears Configuration Error Interrupt Flag

5.173.2.1634 DMA2D_IFSR_CCTCIF

```
#define DMA2D_IFSR_CCTCIF DMA2D_IFCR_CCTCIF
```

Clears CLUT Transfer Complete Interrupt Flag

5.173.2.1635 DMA2D_IFSR_CTCIF

```
#define DMA2D_IFSR_CTCIF DMA2D_IFCR_CTCIF
```

Clears Transfer Complete Interrupt Flag

5.173.2.1636 DMA2D_IFSR_CTEIF

```
#define DMA2D_IFSR_CTEIF DMA2D_IFCR_CTEIF
```

Clears Transfer Error Interrupt Flag

5.173.2.1637 DMA2D_IFSR_CTWIF

```
#define DMA2D_IFSR_CTWIF DMA2D_IFCR_CTWIF
```

Clears Transfer Watermark Interrupt Flag

5.173.2.1638 DMA2D_ISR_CAEIF

```
#define DMA2D_ISR_CAEIF ((uint32_t)0x00000008)
```

CLUT Access Error Interrupt Flag

5.173.2.1639 DMA2D_ISR_CEIF

```
#define DMA2D_ISR_CEIF ((uint32_t)0x00000020)
```

Configuration Error Interrupt Flag

5.173.2.1640 DMA2D_ISR_CTCIF

```
#define DMA2D_ISR_CTCIF ((uint32_t)0x00000010)
```

CLUT Transfer Complete Interrupt Flag

5.173.2.1641 DMA2D_ISR_TCIF

```
#define DMA2D_ISR_TCIF ((uint32_t)0x00000002)
```

Transfer Complete Interrupt Flag

5.173.2.1642 DMA2D_ISR_TEIF

```
#define DMA2D_ISR_TEIF ((uint32_t)0x00000001)
```

Transfer Error Interrupt Flag

5.173.2.1643 DMA2D_ISR_TWIF

```
#define DMA2D_ISR_TWIF ((uint32_t)0x00000004)
```

Transfer Watermark Interrupt Flag

5.173.2.1644 DMA2D_LWR_LW

```
#define DMA2D_LWR_LW ((uint32_t)0x0000FFFF)
```

Line Watermark

5.173.2.1645 DMA2D_NLR_NL

```
#define DMA2D_NLR_NL ((uint32_t)0x0000FFFF)
```

Number of Lines

5.173.2.1646 DMA2D_NLR_PL

```
#define DMA2D_NLR_PL ((uint32_t)0x3FFF0000)
```

Pixel per Lines

5.173.2.1647 DMA2D_OCOLR_ALPHA_1

```
#define DMA2D_OCOLR_ALPHA_1 ((uint32_t)0xFF000000)
```

Alpha Channel Value Mode_RGB565

5.173.2.1648 DMA2D_OCOLR_ALPHA_3

```
#define DMA2D_OCOLR_ALPHA_3 ((uint32_t)0x00008000)
```

Alpha Channel Value Mode_ARGB4444

5.173.2.1649 DMA2D_OCOLR_ALPHA_4

```
#define DMA2D_OCOLR_ALPHA_4 ((uint32_t)0x0000F000)
```

Alpha Channel Value

5.173.2.1650 DMA2D_OCOLR_BLUE_1

```
#define DMA2D_OCOLR_BLUE_1 ((uint32_t)0x000000FF)
```

<Mode_ARGB8888/RGB888 BLUE Value

5.173.2.1651 DMA2D_OCOLR_BLUE_2

```
#define DMA2D_OCOLR_BLUE_2 ((uint32_t)0x0000001F)
```

BLUE Value

5.173.2.1652 DMA2D_OCOLR_BLUE_3

```
#define DMA2D_OCOLR_BLUE_3 ((uint32_t)0x00000001F)
```

BLUE Value

5.173.2.1653 DMA2D_OCOLR_BLUE_4

```
#define DMA2D_OCOLR_BLUE_4 ((uint32_t)0x00000000F)
```

BLUE Value

5.173.2.1654 DMA2D_OCOLR_GREEN_1

```
#define DMA2D_OCOLR_GREEN_1 ((uint32_t)0x0000FF00)
```

GREEN Value

5.173.2.1655 DMA2D_OCOLR_GREEN_2

```
#define DMA2D_OCOLR_GREEN_2 ((uint32_t)0x000007E0)
```

GREEN Value

5.173.2.1656 DMA2D_OCOLR_GREEN_3

```
#define DMA2D_OCOLR_GREEN_3 ((uint32_t)0x000003E0)
```

GREEN Value

5.173.2.1657 DMA2D_OCOLR_GREEN_4

```
#define DMA2D_OCOLR_GREEN_4 ((uint32_t)0x000000F0)
```

GREEN Value

5.173.2.1658 DMA2D_OCOLR_RED_1

```
#define DMA2D_OCOLR_RED_1 ((uint32_t)0x00FF0000)
```

Red Value

5.173.2.1659 DMA2D_OCOLR_RED_2

```
#define DMA2D_OCOLR_RED_2 ((uint32_t)0x0000F800)
```

Red Value Mode_ARGB1555

5.173.2.1660 DMA2D_OCOLR_RED_3

```
#define DMA2D_OCOLR_RED_3 ((uint32_t)0x00007C00)
```

Red Value

5.173.2.1661 DMA2D_OCOLR_RED_4

```
#define DMA2D_OCOLR_RED_4 ((uint32_t)0x00000F00)
```

Red Value

5.173.2.1662 DMA2D_OMAR_MA

```
#define DMA2D_OMAR_MA ((uint32_t)0xFFFFFFFF)
```

Memory Address

5.173.2.1663 DMA2D_OOR_LO

```
#define DMA2D_OOR_LO ((uint32_t)0x00003FFF)
```

Line Offset

5.173.2.1664 DMA2D_OPFCCR_CM

```
#define DMA2D_OPFCCR_CM ((uint32_t)0x00000007)
```

Color mode CM[2:0]

5.173.2.1665 DMA2D_OPFCCR_CM_0

```
#define DMA2D_OPFCCR_CM_0 ((uint32_t)0x00000001)
```

Color mode CM bit 0

5.173.2.1666 DMA2D_OPFCCR_CM_1

```
#define DMA2D_OPFCCR_CM_1 ((uint32_t)0x00000002)
```

Color mode CM bit 1

5.173.2.1667 DMA2D_OPFCCR_CM_2

```
#define DMA2D_OPFCCR_CM_2 ((uint32_t)0x00000004)
```

Color mode CM bit 2

5.173.2.1668 ETH_MACCR_BL

```
#define ETH_MACCR_BL
```

Value:

```
          ((uint32_t)0x00000060) /* Back-off limit: random
integer number (r) of slot time delays before rescheduling
a transmission attempt during retries after a
collision: 0 =< r <2^k */
```

5.173.2.1669 EXTI_EMR_MR0

```
#define EXTI_EMR_MR0 ((uint32_t)0x00000001)
```

Event Mask on line 0

5.173.2.1670 EXTI_EMR_MR1

```
#define EXTI_EMR_MR1 ((uint32_t)0x00000002)
```

Event Mask on line 1

5.173.2.1671 EXTI_EMR_MR10

```
#define EXTI_EMR_MR10 ((uint32_t)0x00000400)
```

Event Mask on line 10

5.173.2.1672 EXTI_EMR_MR11

```
#define EXTI_EMR_MR11 ((uint32_t)0x00000800)
```

Event Mask on line 11

5.173.2.1673 EXTI_EMR_MR12

```
#define EXTI_EMR_MR12 ((uint32_t)0x00001000)
```

Event Mask on line 12

5.173.2.1674 EXTI_EMR_MR13

```
#define EXTI_EMR_MR13 ((uint32_t)0x00002000)
```

Event Mask on line 13

5.173.2.1675 EXTI_EMR_MR14

```
#define EXTI_EMR_MR14 ((uint32_t)0x00004000)
```

Event Mask on line 14

5.173.2.1676 EXTI_EMR_MR15

```
#define EXTI_EMR_MR15 ((uint32_t)0x00008000)
```

Event Mask on line 15

5.173.2.1677 EXTI_EMR_MR16

```
#define EXTI_EMR_MR16 ((uint32_t)0x00010000)
```

Event Mask on line 16

5.173.2.1678 EXTI_EMR_MR17

```
#define EXTI_EMR_MR17 ((uint32_t)0x00020000)
```

Event Mask on line 17

5.173.2.1679 EXTI_EMR_MR18

```
#define EXTI_EMR_MR18 ((uint32_t)0x00040000)
```

Event Mask on line 18

5.173.2.1680 EXTI_EMR_MR19

```
#define EXTI_EMR_MR19 ((uint32_t)0x00080000)
```

Event Mask on line 19

5.173.2.1681 EXTI_EMR_MR2

```
#define EXTI_EMR_MR2 ((uint32_t)0x00000004)
```

Event Mask on line 2

5.173.2.1682 EXTI_EMR_MR23

```
#define EXTI_EMR_MR23 ((uint32_t)0x00800000)
```

Event Mask on line 19

5.173.2.1683 EXTI_EMR_MR3

```
#define EXTI_EMR_MR3 ((uint32_t)0x00000008)
```

Event Mask on line 3

5.173.2.1684 EXTI_EMR_MR4

```
#define EXTI_EMR_MR4 ((uint32_t)0x00000010)
```

Event Mask on line 4

5.173.2.1685 EXTI_EMR_MR5

```
#define EXTI_EMR_MR5 ((uint32_t)0x00000020)
```

Event Mask on line 5

5.173.2.1686 EXTI_EMR_MR6

```
#define EXTI_EMR_MR6 ((uint32_t)0x00000040)
```

Event Mask on line 6

5.173.2.1687 EXTI_EMR_MR7

```
#define EXTI_EMR_MR7 ((uint32_t)0x00000080)
```

Event Mask on line 7

5.173.2.1688 EXTI_EMR_MR8

```
#define EXTI_EMR_MR8 ((uint32_t)0x00000100)
```

Event Mask on line 8

5.173.2.1689 EXTI_EMR_MR9

```
#define EXTI_EMR_MR9 ((uint32_t)0x00000200)
```

Event Mask on line 9

5.173.2.1690 EXTI_FTSR_TR0

```
#define EXTI_FTSR_TR0 ((uint32_t)0x00000001)
```

Falling trigger event configuration bit of line 0

5.173.2.1691 EXTI_FTSR_TR1

```
#define EXTI_FTSR_TR1 ((uint32_t)0x00000002)
```

Falling trigger event configuration bit of line 1

5.173.2.1692 EXTI_FTSR_TR10

```
#define EXTI_FTSR_TR10 ((uint32_t)0x00000400)
```

Falling trigger event configuration bit of line 10

5.173.2.1693 EXTI_FTSR_TR11

```
#define EXTI_FTSR_TR11 ((uint32_t)0x00000800)
```

Falling trigger event configuration bit of line 11

5.173.2.1694 EXTI_FTSR_TR12

```
#define EXTI_FTSR_TR12 ((uint32_t)0x00001000)
```

Falling trigger event configuration bit of line 12

5.173.2.1695 EXTI_FTSR_TR13

```
#define EXTI_FTSR_TR13 ((uint32_t)0x00002000)
```

Falling trigger event configuration bit of line 13

5.173.2.1696 EXTI_FTSR_TR14

```
#define EXTI_FTSR_TR14 ((uint32_t)0x00004000)
```

Falling trigger event configuration bit of line 14

5.173.2.1697 EXTI_FTSR_TR15

```
#define EXTI_FTSR_TR15 ((uint32_t)0x00008000)
```

Falling trigger event configuration bit of line 15

5.173.2.1698 EXTI_FTSR_TR16

```
#define EXTI_FTSR_TR16 ((uint32_t)0x00010000)
```

Falling trigger event configuration bit of line 16

5.173.2.1699 EXTI_FTSR_TR17

```
#define EXTI_FTSR_TR17 ((uint32_t)0x00020000)
```

Falling trigger event configuration bit of line 17

5.173.2.1700 EXTI_FTSR_TR18

```
#define EXTI_FTSR_TR18 ((uint32_t)0x00040000)
```

Falling trigger event configuration bit of line 18

5.173.2.1701 EXTI_FTSR_TR19

```
#define EXTI_FTSR_TR19 ((uint32_t)0x00080000)
```

Falling trigger event configuration bit of line 19

5.173.2.1702 EXTI_FTSR_TR2

```
#define EXTI_FTSR_TR2 ((uint32_t)0x00000004)
```

Falling trigger event configuration bit of line 2

5.173.2.1703 EXTI_FTSR_TR23

```
#define EXTI_FTSR_TR23 ((uint32_t)0x00800000)
```

Falling trigger event configuration bit of line 23

5.173.2.1704 EXTI_FTSR_TR3

```
#define EXTI_FTSR_TR3 ((uint32_t)0x00000008)
```

Falling trigger event configuration bit of line 3

5.173.2.1705 EXTI_FTSR_TR4

```
#define EXTI_FTSR_TR4 ((uint32_t)0x00000010)
```

Falling trigger event configuration bit of line 4

5.173.2.1706 EXTI_FTSR_TR5

```
#define EXTI_FTSR_TR5 ((uint32_t)0x00000020)
```

Falling trigger event configuration bit of line 5

5.173.2.1707 EXTI_FTSR_TR6

```
#define EXTI_FTSR_TR6 ((uint32_t)0x00000040)
```

Falling trigger event configuration bit of line 6

5.173.2.1708 EXTI_FTSR_TR7

```
#define EXTI_FTSR_TR7 ((uint32_t)0x00000080)
```

Falling trigger event configuration bit of line 7

5.173.2.1709 EXTI_FTSR_TR8

```
#define EXTI_FTSR_TR8 ((uint32_t)0x00000100)
```

Falling trigger event configuration bit of line 8

5.173.2.1710 EXTI_FTSR_TR9

```
#define EXTI_FTSR_TR9 ((uint32_t)0x00000200)
```

Falling trigger event configuration bit of line 9

5.173.2.1711 EXTI_IMR_MR0

```
#define EXTI_IMR_MR0 ((uint32_t)0x00000001)
```

Interrupt Mask on line 0

5.173.2.1712 EXTI_IMR_MR1

```
#define EXTI_IMR_MR1 ((uint32_t)0x00000002)
```

Interrupt Mask on line 1

5.173.2.1713 EXTI_IMR_MR10

```
#define EXTI_IMR_MR10 ((uint32_t)0x00000400)
```

Interrupt Mask on line 10

5.173.2.1714 EXTI_IMR_MR11

```
#define EXTI_IMR_MR11 ((uint32_t)0x00000800)
```

Interrupt Mask on line 11

5.173.2.1715 EXTI_IMR_MR12

```
#define EXTI_IMR_MR12 ((uint32_t)0x00001000)
```

Interrupt Mask on line 12

5.173.2.1716 EXTI_IMR_MR13

```
#define EXTI_IMR_MR13 ((uint32_t)0x00002000)
```

Interrupt Mask on line 13

5.173.2.1717 EXTI_IMR_MR14

```
#define EXTI_IMR_MR14 ((uint32_t)0x00004000)
```

Interrupt Mask on line 14

5.173.2.1718 EXTI_IMR_MR15

```
#define EXTI_IMR_MR15 ((uint32_t)0x00008000)
```

Interrupt Mask on line 15

5.173.2.1719 EXTI_IMR_MR16

```
#define EXTI_IMR_MR16 ((uint32_t)0x00010000)
```

Interrupt Mask on line 16

5.173.2.1720 EXTI_IMR_MR17

```
#define EXTI_IMR_MR17 ((uint32_t)0x00020000)
```

Interrupt Mask on line 17

5.173.2.1721 EXTI_IMR_MR18

```
#define EXTI_IMR_MR18 ((uint32_t)0x00040000)
```

Interrupt Mask on line 18

5.173.2.1722 EXTI_IMR_MR19

```
#define EXTI_IMR_MR19 ((uint32_t)0x00080000)
```

Interrupt Mask on line 19

5.173.2.1723 EXTI_IMR_MR2

```
#define EXTI_IMR_MR2 ((uint32_t)0x00000004)
```

Interrupt Mask on line 2

5.173.2.1724 EXTI_IMR_MR23

```
#define EXTI_IMR_MR23 ((uint32_t)0x00800000)
```

Interrupt Mask on line 23

5.173.2.1725 EXTI_IMR_MR3

```
#define EXTI_IMR_MR3 ((uint32_t)0x00000008)
```

Interrupt Mask on line 3

5.173.2.1726 EXTI_IMR_MR4

```
#define EXTI_IMR_MR4 ((uint32_t)0x00000010)
```

Interrupt Mask on line 4

5.173.2.1727 EXTI_IMR_MR5

```
#define EXTI_IMR_MR5 ((uint32_t)0x00000020)
```

Interrupt Mask on line 5

5.173.2.1728 EXTI_IMR_MR6

```
#define EXTI_IMR_MR6 ((uint32_t)0x00000040)
```

Interrupt Mask on line 6

5.173.2.1729 EXTI_IMR_MR7

```
#define EXTI_IMR_MR7 ((uint32_t)0x00000080)
```

Interrupt Mask on line 7

5.173.2.1730 EXTI_IMR_MR8

```
#define EXTI_IMR_MR8 ((uint32_t)0x00000100)
```

Interrupt Mask on line 8

5.173.2.1731 EXTI_IMR_MR9

```
#define EXTI_IMR_MR9 ((uint32_t)0x00000200)
```

Interrupt Mask on line 9

5.173.2.1732 EXTI_PR_PR0

```
#define EXTI_PR_PR0 ((uint32_t)0x00000001)
```

Pending bit for line 0

5.173.2.1733 EXTI_PR_PR1

```
#define EXTI_PR_PR1 ((uint32_t)0x00000002)
```

Pending bit for line 1

5.173.2.1734 EXTI_PR_PR10

```
#define EXTI_PR_PR10 ((uint32_t)0x00000400)
```

Pending bit for line 10

5.173.2.1735 EXTI_PR_PR11

```
#define EXTI_PR_PR11 ((uint32_t)0x00000800)
```

Pending bit for line 11

5.173.2.1736 EXTI_PR_PR12

```
#define EXTI_PR_PR12 ((uint32_t)0x00001000)
```

Pending bit for line 12

5.173.2.1737 EXTI_PR_PR13

```
#define EXTI_PR_PR13 ((uint32_t)0x00002000)
```

Pending bit for line 13

5.173.2.1738 EXTI_PR_PR14

```
#define EXTI_PR_PR14 ((uint32_t)0x00004000)
```

Pending bit for line 14

5.173.2.1739 EXTI_PR_PR15

```
#define EXTI_PR_PR15 ((uint32_t)0x00008000)
```

Pending bit for line 15

5.173.2.1740 EXTI_PR_PR16

```
#define EXTI_PR_PR16 ((uint32_t)0x00010000)
```

Pending bit for line 16

5.173.2.1741 EXTI_PR_PR17

```
#define EXTI_PR_PR17 ((uint32_t)0x00020000)
```

Pending bit for line 17

5.173.2.1742 EXTI_PR_PR18

```
#define EXTI_PR_PR18 ((uint32_t)0x00040000)
```

Pending bit for line 18

5.173.2.1743 EXTI_PR_PR19

```
#define EXTI_PR_PR19 ((uint32_t)0x00080000)
```

Pending bit for line 19

5.173.2.1744 EXTI_PR_PR2

```
#define EXTI_PR_PR2 ((uint32_t)0x00000004)
```

Pending bit for line 2

5.173.2.1745 EXTI_PR_PR23

```
#define EXTI_PR_PR23 ((uint32_t)0x00800000)
```

Pending bit for line 23

5.173.2.1746 EXTI_PR_PR3

```
#define EXTI_PR_PR3 ((uint32_t)0x00000008)
```

Pending bit for line 3

5.173.2.1747 EXTI_PR_PR4

```
#define EXTI_PR_PR4 ((uint32_t)0x00000010)
```

Pending bit for line 4

5.173.2.1748 EXTI_PR_PR5

```
#define EXTI_PR_PR5 ((uint32_t)0x00000020)
```

Pending bit for line 5

5.173.2.1749 EXTI_PR_PR6

```
#define EXTI_PR_PR6 ((uint32_t)0x00000040)
```

Pending bit for line 6

5.173.2.1750 EXTI_PR_PR7

```
#define EXTI_PR_PR7 ((uint32_t)0x00000080)
```

Pending bit for line 7

5.173.2.1751 EXTI_PR_PR8

```
#define EXTI_PR_PR8 ((uint32_t)0x00000100)
```

Pending bit for line 8

5.173.2.1752 EXTI_PR_PR9

```
#define EXTI_PR_PR9 ((uint32_t)0x00000200)
```

Pending bit for line 9

5.173.2.1753 EXTI_RTSR_TR0

```
#define EXTI_RTSR_TR0 ((uint32_t)0x00000001)
```

Rising trigger event configuration bit of line 0

5.173.2.1754 EXTI_RTSR_TR1

```
#define EXTI_RTSR_TR1 ((uint32_t)0x00000002)
```

Rising trigger event configuration bit of line 1

5.173.2.1755 EXTI_RTSR_TR10

```
#define EXTI_RTSR_TR10 ((uint32_t)0x00000400)
```

Rising trigger event configuration bit of line 10

5.173.2.1756 EXTI_RTSR_TR11

```
#define EXTI_RTSR_TR11 ((uint32_t)0x00000800)
```

Rising trigger event configuration bit of line 11

5.173.2.1757 EXTI_RTSR_TR12

```
#define EXTI_RTSR_TR12 ((uint32_t)0x00001000)
```

Rising trigger event configuration bit of line 12

5.173.2.1758 EXTI_RTSR_TR13

```
#define EXTI_RTSR_TR13 ((uint32_t)0x00002000)
```

Rising trigger event configuration bit of line 13

5.173.2.1759 EXTI_RTSR_TR14

```
#define EXTI_RTSR_TR14 ((uint32_t)0x00004000)
```

Rising trigger event configuration bit of line 14

5.173.2.1760 EXTI_RTSR_TR15

```
#define EXTI_RTSR_TR15 ((uint32_t)0x00008000)
```

Rising trigger event configuration bit of line 15

5.173.2.1761 EXTI_RTSR_TR16

```
#define EXTI_RTSR_TR16 ((uint32_t)0x00010000)
```

Rising trigger event configuration bit of line 16

5.173.2.1762 EXTI_RTSR_TR17

```
#define EXTI_RTSR_TR17 ((uint32_t)0x00020000)
```

Rising trigger event configuration bit of line 17

5.173.2.1763 EXTI_RTSR_TR18

```
#define EXTI_RTSR_TR18 ((uint32_t)0x00040000)
```

Rising trigger event configuration bit of line 18

5.173.2.1764 EXTI_RTSR_TR19

```
#define EXTI_RTSR_TR19 ((uint32_t)0x00080000)
```

Rising trigger event configuration bit of line 19

5.173.2.1765 EXTI_RTSR_TR2

```
#define EXTI_RTSR_TR2 ((uint32_t)0x00000004)
```

Rising trigger event configuration bit of line 2

5.173.2.1766 EXTI_RTSR_TR23

```
#define EXTI_RTSR_TR23 ((uint32_t)0x00800000)
```

Rising trigger event configuration bit of line 23

5.173.2.1767 EXTI_RTSR_TR3

```
#define EXTI_RTSR_TR3 ((uint32_t)0x00000008)
```

Rising trigger event configuration bit of line 3

5.173.2.1768 EXTI_RTSR_TR4

```
#define EXTI_RTSR_TR4 ((uint32_t)0x00000010)
```

Rising trigger event configuration bit of line 4

5.173.2.1769 EXTI_RTSR_TR5

```
#define EXTI_RTSR_TR5 ((uint32_t)0x00000020)
```

Rising trigger event configuration bit of line 5

5.173.2.1770 EXTI_RTSR_TR6

```
#define EXTI_RTSR_TR6 ((uint32_t)0x00000040)
```

Rising trigger event configuration bit of line 6

5.173.2.1771 EXTI_RTSR_TR7

```
#define EXTI_RTSR_TR7 ((uint32_t)0x00000080)
```

Rising trigger event configuration bit of line 7

5.173.2.1772 EXTI_RTSR_TR8

```
#define EXTI_RTSR_TR8 ((uint32_t)0x00000100)
```

Rising trigger event configuration bit of line 8

5.173.2.1773 EXTI_RTSR_TR9

```
#define EXTI_RTSR_TR9 ((uint32_t)0x00000200)
```

Rising trigger event configuration bit of line 9

5.173.2.1774 EXTI_SWIER_SWIER0

```
#define EXTI_SWIER_SWIER0 ((uint32_t)0x00000001)
```

Software Interrupt on line 0

5.173.2.1775 EXTI_SWIER_SWIER1

```
#define EXTI_SWIER_SWIER1 ((uint32_t)0x00000002)
```

Software Interrupt on line 1

5.173.2.1776 EXTI_SWIER_SWIER10

```
#define EXTI_SWIER_SWIER10 ((uint32_t)0x00000400)
```

Software Interrupt on line 10

5.173.2.1777 EXTI_SWIER_SWIER11

```
#define EXTI_SWIER_SWIER11 ((uint32_t)0x00000800)
```

Software Interrupt on line 11

5.173.2.1778 EXTI_SWIER_SWIER12

```
#define EXTI_SWIER_SWIER12 ((uint32_t)0x00001000)
```

Software Interrupt on line 12

5.173.2.1779 EXTI_SWIER_SWIER13

```
#define EXTI_SWIER_SWIER13 ((uint32_t)0x00002000)
```

Software Interrupt on line 13

5.173.2.1780 EXTI_SWIER_SWIER14

```
#define EXTI_SWIER_SWIER14 ((uint32_t)0x00004000)
```

Software Interrupt on line 14

5.173.2.1781 EXTI_SWIER_SWIER15

```
#define EXTI_SWIER_SWIER15 ((uint32_t)0x00008000)
```

Software Interrupt on line 15

5.173.2.1782 EXTI_SWIER_SWIER16

```
#define EXTI_SWIER_SWIER16 ((uint32_t)0x00010000)
```

Software Interrupt on line 16

5.173.2.1783 EXTI_SWIER_SWIER17

```
#define EXTI_SWIER_SWIER17 ((uint32_t)0x00020000)
```

Software Interrupt on line 17

5.173.2.1784 EXTI_SWIER_SWIER18

```
#define EXTI_SWIER_SWIER18 ((uint32_t)0x00040000)
```

Software Interrupt on line 18

5.173.2.1785 EXTI_SWIER_SWIER19

```
#define EXTI_SWIER_SWIER19 ((uint32_t)0x00080000)
```

Software Interrupt on line 19

5.173.2.1786 EXTI_SWIER_SWIER2

```
#define EXTI_SWIER_SWIER2 ((uint32_t)0x00000004)
```

Software Interrupt on line 2

5.173.2.1787 EXTI_SWIER_SWIER23

```
#define EXTI_SWIER_SWIER23 ((uint32_t)0x00800000)
```

Software Interrupt on line 23

5.173.2.1788 EXTI_SWIER_SWIER3

```
#define EXTI_SWIER_SWIER3 ((uint32_t)0x00000008)
```

Software Interrupt on line 3

5.173.2.1789 EXTI_SWIER_SWIER4

```
#define EXTI_SWIER_SWIER4 ((uint32_t)0x00000010)
```

Software Interrupt on line 4

5.173.2.1790 EXTI_SWIER_SWIER5

```
#define EXTI_SWIER_SWIER5 ((uint32_t)0x00000020)
```

Software Interrupt on line 5

5.173.2.1791 EXTI_SWIER_SWIER6

```
#define EXTI_SWIER_SWIER6 ((uint32_t)0x00000040)
```

Software Interrupt on line 6

5.173.2.1792 EXTI_SWIER_SWIER7

```
#define EXTI_SWIER_SWIER7 ((uint32_t)0x00000080)
```

Software Interrupt on line 7

5.173.2.1793 EXTI_SWIER_SWIER8

```
#define EXTI_SWIER_SWIER8 ((uint32_t)0x00000100)
```

Software Interrupt on line 8

5.173.2.1794 EXTI_SWIER_SWIER9

```
#define EXTI_SWIER_SWIER9 ((uint32_t)0x00000200)
```

Software Interrupt on line 9

5.173.2.1795 I2C_CCR_CCR

```
#define I2C_CCR_CCR ((uint16_t)0xFFFF)
```

Clock Control Register in Fast/Standard mode (Master mode)

5.173.2.1796 I2C_CCR_DUTY

```
#define I2C_CCR_DUTY ((uint16_t)0x4000)
```

Fast Mode Duty Cycle

5.173.2.1797 I2C_CCR_FS

```
#define I2C_CCR_FS ((uint16_t)0x8000)
```

I2C Master Mode Selection

5.173.2.1798 I2C_CR1_ACK

```
#define I2C_CR1_ACK ((uint16_t)0x0400)
```

Acknowledge Enable

5.173.2.1799 I2C_CR1_ALERT

```
#define I2C_CR1_ALERT ((uint16_t)0x2000)
```

SMBus Alert

5.173.2.1800 I2C_CR1_ENARP

```
#define I2C_CR1_ENARP ((uint16_t)0x0010)
```

ARP Enable

5.173.2.1801 I2C_CR1_ENGC

```
#define I2C_CR1_ENGC ((uint16_t)0x0040)
```

General Call Enable

5.173.2.1802 I2C_CR1_ENPEC

```
#define I2C_CR1_ENPEC ((uint16_t)0x0020)
```

PEC Enable

5.173.2.1803 I2C_CR1_NOSTRETCH

```
#define I2C_CR1_NOSTRETCH ((uint16_t)0x0080)
```

Clock Stretching Disable (Slave mode)

5.173.2.1804 I2C_CR1_PE

```
#define I2C_CR1_PE ((uint16_t)0x0001)
```

Peripheral Enable

5.173.2.1805 I2C_CR1_PEC

```
#define I2C_CR1_PEC ((uint16_t)0x1000)
```

Packet Error Checking

5.173.2.1806 I2C_CR1_POS

```
#define I2C_CR1_POS ((uint16_t)0x0800)
```

Acknowledge/PEC Position (for data reception)

5.173.2.1807 I2C_CR1_SMBTYPE

```
#define I2C_CR1_SMBTYPE ((uint16_t)0x0008)
```

SMBus Type

5.173.2.1808 I2C_CR1_SMBUS

```
#define I2C_CR1_SMBUS ((uint16_t)0x0002)
```

SMBus Mode

5.173.2.1809 I2C_CR1_START

```
#define I2C_CR1_START ((uint16_t)0x0100)
```

Start Generation

5.173.2.1810 I2C_CR1_STOP

```
#define I2C_CR1_STOP ((uint16_t)0x0200)
```

Stop Generation

5.173.2.1811 I2C_CR1_SWRST

```
#define I2C_CR1_SWRST ((uint16_t)0x8000)
```

Software Reset

5.173.2.1812 I2C_CR2_DMAEN

```
#define I2C_CR2_DMAEN ((uint16_t)0x0800)
```

DMA Requests Enable

5.173.2.1813 I2C_CR2_FREQ

```
#define I2C_CR2_FREQ ((uint16_t)0x003F)
```

FREQ[5:0] bits (Peripheral Clock Frequency)

5.173.2.1814 I2C_CR2_FREQ_0

```
#define I2C_CR2_FREQ_0 ((uint16_t)0x0001)
```

Bit 0

5.173.2.1815 I2C_CR2_FREQ_1

```
#define I2C_CR2_FREQ_1 ((uint16_t)0x0002)
```

Bit 1

5.173.2.1816 I2C_CR2_FREQ_2

```
#define I2C_CR2_FREQ_2 ((uint16_t)0x0004)
```

Bit 2

5.173.2.1817 I2C_CR2_FREQ_3

```
#define I2C_CR2_FREQ_3 ((uint16_t)0x0008)
```

Bit 3

5.173.2.1818 I2C_CR2_FREQ_4

```
#define I2C_CR2_FREQ_4 ((uint16_t)0x0010)
```

Bit 4

5.173.2.1819 I2C_CR2_FREQ_5

```
#define I2C_CR2_FREQ_5 ((uint16_t)0x0020)
```

Bit 5

5.173.2.1820 I2C_CR2_ITBUFEN

```
#define I2C_CR2_ITBUFEN ((uint16_t)0x0400)
```

Buffer Interrupt Enable

5.173.2.1821 I2C_CR2_ITERREN

```
#define I2C_CR2_ITERREN ((uint16_t)0x0100)
```

Error Interrupt Enable

5.173.2.1822 I2C_CR2 ITEVTEN

```
#define I2C_CR2_ITEVVTEN ((uint16_t)0x0200)
```

Event Interrupt Enable

5.173.2.1823 I2C_CR2_LAST

```
#define I2C_CR2_LAST ((uint16_t)0x1000)
```

DMA Last Transfer

5.173.2.1824 I2C_DR_DR

```
#define I2C_DR_DR ((uint8_t)0xFF)
```

8-bit Data Register

5.173.2.1825 I2C_FLTR_ANOFF

```
#define I2C_FLTR_ANOFF ((uint8_t)0x10)
```

Analog Noise Filter OFF

5.173.2.1826 I2C_FLTR_DNF

```
#define I2C_FLTR_DNF ((uint8_t)0x0F)
```

Digital Noise Filter

5.173.2.1827 I2C_OAR1_ADD0

```
#define I2C_OAR1_ADD0 ((uint16_t)0x0001)
```

Bit 0

5.173.2.1828 I2C_OAR1_ADD1

```
#define I2C_OAR1_ADD1 ((uint16_t)0x0002)
```

Bit 1

5.173.2.1829 I2C_OAR1_ADD1_7

```
#define I2C_OAR1_ADD1_7 ((uint16_t)0x00FE)
```

Interface Address

5.173.2.1830 I2C_OAR1_ADD2

```
#define I2C_OAR1_ADD2 ((uint16_t)0x0004)
```

Bit 2

5.173.2.1831 I2C_OAR1_ADD3

```
#define I2C_OAR1_ADD3 ((uint16_t)0x0008)
```

Bit 3

5.173.2.1832 I2C_OAR1_ADD4

```
#define I2C_OAR1_ADD4 ((uint16_t)0x0010)
```

Bit 4

5.173.2.1833 I2C_OAR1_ADD5

```
#define I2C_OAR1_ADD5 ((uint16_t)0x0020)
```

Bit 5

5.173.2.1834 I2C_OAR1_ADD6

```
#define I2C_OAR1_ADD6 ((uint16_t)0x0040)
```

Bit 6

5.173.2.1835 I2C_OAR1_ADD7

```
#define I2C_OAR1_ADD7 ((uint16_t)0x0080)
```

Bit 7

5.173.2.1836 I2C_OAR1_ADD8

```
#define I2C_OAR1_ADD8 ((uint16_t)0x0100)
```

Bit 8

5.173.2.1837 I2C_OAR1_ADD8_9

```
#define I2C_OAR1_ADD8_9 ((uint16_t)0x0300)
```

Interface Address

5.173.2.1838 I2C_OAR1_ADD9

```
#define I2C_OAR1_ADD9 ((uint16_t)0x0200)
```

Bit 9

5.173.2.1839 I2C_OAR1_ADDMODE

```
#define I2C_OAR1_ADDMODE ((uint16_t)0x8000)
```

Addressing Mode (Slave mode)

5.173.2.1840 I2C_OAR2_ADD2

```
#define I2C_OAR2_ADD2 ((uint8_t)0xFE)
```

Interface address

5.173.2.1841 I2C_OAR2_ENDUAL

```
#define I2C_OAR2_ENDUAL ((uint8_t)0x01)
```

Dual addressing mode enable

5.173.2.1842 I2C_SR1_ADD10

```
#define I2C_SR1_ADD10 ((uint16_t)0x0008)
```

10-bit header sent (Master mode)

5.173.2.1843 I2C_SR1_ADDR

```
#define I2C_SR1_ADDR ((uint16_t)0x0002)
```

Address sent (master mode)/matched (slave mode)

5.173.2.1844 I2C_SR1_AF

```
#define I2C_SR1_AF ((uint16_t)0x0400)
```

Acknowledge Failure

5.173.2.1845 I2C_SR1_ARLO

```
#define I2C_SR1_ARLO ((uint16_t)0x0200)
```

Arbitration Lost (master mode)

5.173.2.1846 I2C_SR1_BERR

```
#define I2C_SR1_BERR ((uint16_t)0x0100)
```

Bus Error

5.173.2.1847 I2C_SR1_BTF

```
#define I2C_SR1_BTF ((uint16_t)0x0004)
```

Byte Transfer Finished

5.173.2.1848 I2C_SR1_OVR

```
#define I2C_SR1_OVR ((uint16_t)0x0800)
```

Overrun/Underrun

5.173.2.1849 I2C_SR1_PECERR

```
#define I2C_SR1_PECERR ((uint16_t)0x1000)
```

PEC Error in reception

5.173.2.1850 I2C_SR1_RXNE

```
#define I2C_SR1_RXNE ((uint16_t)0x0040)
```

Data Register not Empty (receivers)

5.173.2.1851 I2C_SR1_SB

```
#define I2C_SR1_SB ((uint16_t)0x0001)
```

Start Bit (Master mode)

5.173.2.1852 I2C_SR1_SMBALERT

```
#define I2C_SR1_SMBALERT ((uint16_t)0x8000)
```

SMBus Alert

5.173.2.1853 I2C_SR1_STOPF

```
#define I2C_SR1_STOPF ((uint16_t)0x0010)
```

Stop detection (Slave mode)

5.173.2.1854 I2C_SR1_TIMEOUT

```
#define I2C_SR1_TIMEOUT ((uint16_t)0x4000)
```

Timeout or Tlow Error

5.173.2.1855 I2C_SR1_TXE

```
#define I2C_SR1_TXE ((uint16_t)0x0080)
```

Data Register Empty (transmitters)

5.173.2.1856 I2C_SR2_BUSY

```
#define I2C_SR2_BUSY ((uint16_t)0x0002)
```

Bus Busy

5.173.2.1857 I2C_SR2_DUALF

```
#define I2C_SR2_DUALF ((uint16_t)0x0080)
```

Dual Flag (Slave mode)

5.173.2.1858 I2C_SR2_GENCALL

```
#define I2C_SR2_GENCALL ((uint16_t)0x0010)
```

General Call Address (Slave mode)

5.173.2.1859 I2C_SR2_MSL

```
#define I2C_SR2_MSL ((uint16_t)0x0001)
```

Master/Slave

5.173.2.1860 I2C_SR2_PEC

```
#define I2C_SR2_PEC ((uint16_t)0xFF00)
```

Packet Error Checking Register

5.173.2.1861 I2C_SR2_SMBDEFAULT

```
#define I2C_SR2_SMBDEFAULT ((uint16_t)0x0020)
```

SMBus Device Default Address (Slave mode)

5.173.2.1862 I2C_SR2_SMBHOST

```
#define I2C_SR2_SMBHOST ((uint16_t)0x0040)
```

SMBus Host Header (Slave mode)

5.173.2.1863 I2C_SR2_TRA

```
#define I2C_SR2_TRA ((uint16_t)0x0004)
```

Transmitter/Receiver

5.173.2.1864 I2C_TRISE_TRISE

```
#define I2C_TRISE_TRISE ((uint8_t)0x3F)
```

Maximum Rise Time in Fast/Standard mode (Master mode)

5.173.2.1865 IWDG_KR_KEY

```
#define IWDG_KR_KEY ((uint16_t)0xFFFF)
```

Key value (write only, read 0000h)

5.173.2.1866 IWDG_PR_PR

```
#define IWDG_PR_PR ((uint8_t)0x07)
```

PR[2:0] (Prescaler divider)

5.173.2.1867 IWDG_PR_PR_0

```
#define IWDG_PR_PR_0 ((uint8_t)0x01)
```

Bit 0

5.173.2.1868 IWDG_PR_PR_1

```
#define IWDG_PR_PR_1 ((uint8_t)0x02)
```

Bit 1

5.173.2.1869 IWDG_PR_PR_2

```
#define IWDG_PR_PR_2 ((uint8_t)0x04)
```

Bit 2

5.173.2.1870 IWDG_RLR_RL

```
#define IWDG_RLR_RL ((uint16_t)0xFFFF)
```

Watchdog counter reload value

5.173.2.1871 IWDG_SR_PVU

```
#define IWDG_SR_PVU ((uint8_t)0x01)
```

Watchdog prescaler value update

5.173.2.1872 IWDG_SR_RVU

```
#define IWDG_SR_RVU ((uint8_t)0x02)
```

Watchdog counter reload value update

5.173.2.1873 LTDC_AWCR_AAH

```
#define LTDC_AWCR_AAH ((uint32_t)0x0000007FF)
```

Accumulated Active heigh

5.173.2.1874 LTDC_AWCR_AAW

```
#define LTDC_AWCR_AAW ((uint32_t)0x0FFF0000)
```

Accumulated Active Width

5.173.2.1875 LTDC_BCCR_BCBLUE

```
#define LTDC_BCCR_BCBLUE ((uint32_t)0x000000FF)
```

Background Blue value

5.173.2.1876 LTDC_BCCR_BCGREEN

```
#define LTDC_BCCR_BCGREEN ((uint32_t)0x0000FF00)
```

Background Green value

5.173.2.1877 LTDC_BCCR_BCRED

```
#define LTDC_BCCR_BCRED ((uint32_t)0x00FF0000)
```

Background Red value

5.173.2.1878 LTDC_BPCR_AHBP

```
#define LTDC_BPCR_AHBP ((uint32_t)0xFFFF0000)
```

Accumulated Horizontal Back Porch

5.173.2.1879 LTDC_BPCR_AVBP

```
#define LTDC_BPCR_AVBP ((uint32_t)0x000007FF)
```

Accumulated Vertical Back Porch

5.173.2.1880 LTDC_CDSR_HDES

```
#define LTDC_CDSR_HDES ((uint32_t)0x00000002)
```

Horizontal Data Enable Status

5.173.2.1881 LTDC_CDSR_HSYNCS

```
#define LTDC_CDSR_HSYNCS ((uint32_t)0x00000008)
```

Horizontal Synchronization Status

5.173.2.1882 LTDC_CDSR_VDES

```
#define LTDC_CDSR_VDES ((uint32_t)0x00000001)
```

Vertical Data Enable Status

5.173.2.1883 LTDC_CDSR_VSYNCS

```
#define LTDC_CDSR_VSYNCS ((uint32_t)0x00000004)
```

Vertical Synchronization Status

5.173.2.1884 LTDC_CPSR_CXPOS

```
#define LTDC_CPSR_CXPOS ((uint32_t)0xFFFF0000)
```

Current X Position

5.173.2.1885 LTDC_CPSR_CYPOS

```
#define LTDC_CPSR_CYPOS ((uint32_t)0x0000FFFF)
```

Current Y Position

5.173.2.1886 LTDC_GCR_DBW

```
#define LTDC_GCR_DBW ((uint32_t)0x00000070)
```

Dither Blue Width

5.173.2.1887 LTDC_GCR_DEN

```
#define LTDC_GCR_DEN ((uint32_t)0x00010000)
```

Dither Enable

5.173.2.1888 LTDC_GCR_DEPOL

```
#define LTDC_GCR_DEPOL ((uint32_t)0x20000000)
```

Data Enable Polarity

5.173.2.1889 LTDC_GCR_DGW

```
#define LTDC_GCR_DGW ((uint32_t)0x00000700)
```

Dither Green Width

5.173.2.1890 LTDC_GCR_DRW

```
#define LTDC_GCR_DRW ((uint32_t)0x00007000)
```

Dither Red Width

5.173.2.1891 LTDC_GCR_HSPOL

```
#define LTDC_GCR_HSPOL ((uint32_t)0x80000000)
```

Horizontal Synchronization Polarity

5.173.2.1892 LTDC_GCR_LTDCEN

```
#define LTDC_GCR_LTDCEN ((uint32_t)0x00000001)
```

LCD-TFT controller enable bit

5.173.2.1893 LTDC_GCR_PCPOL

```
#define LTDC_GCR_PCPOL ((uint32_t)0x10000000)
```

Pixel Clock Polarity

5.173.2.1894 LTDC_GCR_VSPOL

```
#define LTDC_GCR_VSPOL ((uint32_t)0x40000000)
```

Vertical Synchronization Polarity

5.173.2.1895 LTDC_ICR_CFUIF

```
#define LTDC_ICR_CFUIF ((uint32_t)0x00000002)
```

Clears the FIFO Underrun Interrupt Flag

5.173.2.1896 LTDC_ICR_CLIF

```
#define LTDC_ICR_CLIF ((uint32_t)0x00000001)
```

Clears the Line Interrupt Flag

5.173.2.1897 LTDC_ICR_CRRIF

```
#define LTDC_ICR_CRRIF ((uint32_t)0x00000008)
```

Clears Register Reload interrupt Flag

5.173.2.1898 LTDC_ICR_CTERRIF

```
#define LTDC_ICR_CTERRIF ((uint32_t)0x00000004)
```

Clears the Transfer Error Interrupt Flag

5.173.2.1899 LTDC_IER_FUIE

```
#define LTDC_IER_FUIE ((uint32_t)0x00000002)
```

FIFO Underrun Interrupt Enable

5.173.2.1900 LTDC_IER_LIE

```
#define LTDC_IER_LIE ((uint32_t)0x00000001)
```

Line Interrupt Enable

5.173.2.1901 LTDC_IER_RRIE

```
#define LTDC_IER_RRIE ((uint32_t)0x00000008)
```

Register Reload interrupt enable

5.173.2.1902 LTDC_IER_TERRIE

```
#define LTDC_IER_TERRIE ((uint32_t)0x00000004)
```

Transfer Error Interrupt Enable

5.173.2.1903 LTDC_ISR_FUIF

```
#define LTDC_ISR_FUIF ((uint32_t)0x00000002)
```

FIFO Underrun Interrupt Flag

5.173.2.1904 LTDC_ISR_LIF

```
#define LTDC_ISR_LIF ((uint32_t)0x00000001)
```

Line Interrupt Flag

5.173.2.1905 LTDC_ISR_RRIF

```
#define LTDC_ISR_RRIF ((uint32_t)0x00000008)
```

Register Reload interrupt Flag

5.173.2.1906 LTDC_ISR_TERRIF

```
#define LTDC_ISR_TERRIF ((uint32_t)0x00000004)
```

Transfer Error Interrupt Flag

5.173.2.1907 LTDC_LIPCR_LIPOS

```
#define LTDC_LIPCR_LIPOS ((uint32_t)0x000007FF)
```

Line Interrupt Position

5.173.2.1908 LTDC_LxBFCR_BF1

```
#define LTDC_LxBFCR_BF1 ((uint32_t)0x00000700)
```

Blending Factor 1

5.173.2.1909 LTDC_LxBFCR_BF2

```
#define LTDC_LxBFCR_BF2 ((uint32_t)0x00000007)
```

Blending Factor 2

5.173.2.1910 LTDC_LxCACR_CONSTA

```
#define LTDC_LxCACR_CONSTA ((uint32_t)0x000000FF)
```

Constant Alpha

5.173.2.1911 LTDC_LxCFBAR_CFBADD

```
#define LTDC_LxCFBAR_CFBADD ((uint32_t)0xFFFFFFFF)
```

Color Frame Buffer Start Address

5.173.2.1912 LTDC_LxCFBLNR_CFBLNBR

```
#define LTDC_LxCFBLNR_CFBLNBR ((uint32_t)0x000007FF)
```

Frame Buffer Line Number

5.173.2.1913 LTDC_LxCFBLR_CFBLL

```
#define LTDC_LxCFBLR_CFBLL ((uint32_t)0x00001FFF)
```

Color Frame Buffer Line Length

5.173.2.1914 LTDC_LxCFBLR_CFBP

```
#define LTDC_LxCFBLR_CFBP ((uint32_t)0x1FFF0000)
```

Color Frame Buffer Pitch in bytes

5.173.2.1915 LTDC_LxCKCR_CKBLUE

```
#define LTDC_LxCKCR_CKBLUE ((uint32_t)0x000000FF)
```

Color Key Blue value

5.173.2.1916 LTDC_LxCKCR_CKGREEN

```
#define LTDC_LxCKCR_CKGREEN ((uint32_t)0x0000FF00)
```

Color Key Green value

5.173.2.1917 LTDC_LxCKCR_CKRED

```
#define LTDC_LxCKCR_CKRED ((uint32_t)0x00FF0000)
```

Color Key Red value

5.173.2.1918 LTDC_LxCLUTWR_BLUE

```
#define LTDC_LxCLUTWR_BLUE ((uint32_t)0x000000FF)
```

Blue value

5.173.2.1919 LTDC_LxCLUTWR_CLUTADD

```
#define LTDC_LxCLUTWR_CLUTADD ((uint32_t)0xFF000000)
```

CLUT address

5.173.2.1920 LTDC_LxCLUTWR_GREEN

```
#define LTDC_LxCLUTWR_GREEN ((uint32_t)0x0000FF00)
```

Green value

5.173.2.1921 LTDC_LxCLUTWR_RED

```
#define LTDC_LxCLUTWR_RED ((uint32_t)0x00FF0000)
```

Red value

5.173.2.1922 LTDC_LxCR_CLUTEN

```
#define LTDC_LxCR_CLUTEN ((uint32_t)0x00000010)
```

Color Lockup Table Enable

5.173.2.1923 LTDC_LxCR_COLKEN

```
#define LTDC_LxCR_COLKEN ((uint32_t)0x00000002)
```

Color Keying Enable

5.173.2.1924 LTDC_LxCR_LEN

```
#define LTDC_LxCR_LEN ((uint32_t)0x00000001)
```

Layer Enable

5.173.2.1925 LTDC_LxDCCR_DCALPHA

```
#define LTDC_LxDCCR_DCALPHA ((uint32_t)0xFF000000)
```

Default Color Alpha

5.173.2.1926 LTDC_LxDCCR_DCBLUE

```
#define LTDC_LxDCCR_DCBLUE ((uint32_t)0x000000FF)
```

Default Color Blue

5.173.2.1927 LTDC_LxDCCR_DCGREEN

```
#define LTDC_LxDCCR_DCGREEN ((uint32_t)0x0000FF00)
```

Default Color Green

5.173.2.1928 LTDC_LxDCCR_DCRED

```
#define LTDC_LxDCCR_DCRED ((uint32_t)0x00FF0000)
```

Default Color Red

5.173.2.1929 LTDC_LxPFCR_PF

```
#define LTDC_LxPFCR_PF ((uint32_t)0x00000007)
```

Pixel Format

5.173.2.1930 LTDC_LxWHPCR_WHSPPOS

```
#define LTDC_LxWHPCR_WHSPPOS ((uint32_t)0xFFFFF000)
```

Window Horizontal Stop Position

5.173.2.1931 LTDC_LxWHPCR_WHSTPOS

```
#define LTDC_LxWHPCR_WHSTPOS ((uint32_t)0x00000FFF)
```

Window Horizontal Start Position

5.173.2.1932 LTDC_LxWVPCR_WVSPPOS

```
#define LTDC_LxWVPCR_WVSPPOS ((uint32_t)0xFFFF0000)
```

Window Vertical Stop Position

5.173.2.1933 LTDC_LxWVPCR_WVSTPOS

```
#define LTDC_LxWVPCR_WVSTPOS ((uint32_t)0x00000FFF)
```

Window Vertical Start Position

5.173.2.1934 LTDC_SRCR_IMR

```
#define LTDC_SRCR_IMR ((uint32_t)0x00000001)
```

Immediate Reload

5.173.2.1935 LTDC_SRCR_VBR

```
#define LTDC_SRCR_VBR ((uint32_t)0x00000002)
```

Vertical Blanking Reload

5.173.2.1936 LTDC_SSCR_HSW

```
#define LTDC_SSCR_HSW ((uint32_t)0x0FFF0000)
```

Horizontal Synchronization Width

5.173.2.1937 LTDC_SSCR_VSH

```
#define LTDC_SSCR_VSH ((uint32_t)0x000007FF)
```

Vertical Synchronization Height

5.173.2.1938 LTDC_TWCR_TOTALH

```
#define LTDC_TWCR_TOTALH ((uint32_t)0x000007FF)
```

Total Heigh

5.173.2.1939 LTDC_TWCR_TOTALW

```
#define LTDC_TWCR_TOTALW ((uint32_t)0x0FFF0000)
```

Total Width

5.173.2.1940 PWR_CR_ADCDC1

```
#define PWR_CR_ADCDC1 ((uint32_t)0x000002000)
```

Refer to AN4073 on how to use this bit

5.173.2.1941 PWR_CR_CSBF

```
#define PWR_CR_CSBF ((uint32_t)0x00000008)
```

Clear Standby Flag

5.173.2.1942 PWR_CR_CWUF

```
#define PWR_CR_CWUF ((uint32_t)0x00000004)
```

Clear Wakeup Flag

5.173.2.1943 PWR_CR_DBP

```
#define PWR_CR_DBP ((uint32_t)0x000000100)
```

Disable Backup Domain write protection

5.173.2.1944 PWR_CR_FISSR

```
#define PWR_CR_FISSR ((uint32_t)0x002000000)
```

Flash Interface Stop while System Run

5.173.2.1945 PWR_CR_FMSSR

```
#define PWR_CR_FMSSR ((uint32_t)0x001000000)
```

Flash Memory Sleep System Run

5.173.2.1946 PWR_CR_FPDS

```
#define PWR_CR_FPDS ((uint32_t)0x00000200)
```

Flash power down in Stop mode

5.173.2.1947 PWR_CR_LPDS

```
#define PWR_CR_LPDS ((uint32_t)0x00000001)
```

Low-Power Deepsleep

5.173.2.1948 PWR_CR_LPLVDS

```
#define PWR_CR_LPLVDS ((uint32_t)0x00000400)
```

Low-power regulator Low Voltage in Deep Sleep mode

5.173.2.1949 PWR_CR_LPUDS

```
#define PWR_CR_LPUDS ((uint32_t)0x00000400)
```

Low-Power Regulator in Stop under-drive mode

5.173.2.1950 PWR_CR_MRLVDS

```
#define PWR_CR_MRLVDS ((uint32_t)0x00000800)
```

Main regulator Low Voltage in Deep Sleep mode

5.173.2.1951 PWR_CR_MRUDS

```
#define PWR_CR_MRUDS ((uint32_t)0x00000800)
```

Main regulator in Stop under-drive mode

5.173.2.1952 PWR_CR_ODEN

```
#define PWR_CR_ODEN ((uint32_t)0x00010000)
```

Over Drive enable

5.173.2.1953 PWR_CR_ODSWEN

```
#define PWR_CR_ODSWEN ((uint32_t)0x00020000)
```

Over Drive switch enabled

5.173.2.1954 PWR_CR_PDDS

```
#define PWR_CR_PDDS ((uint32_t)0x00000002)
```

Power Down Deepsleep

5.173.2.1955 PWR_CR_PLS

```
#define PWR_CR_PLS ((uint32_t)0x000000E0)
```

PLS[2:0] bits (PVD Level Selection)

5.173.2.1956 PWR_CR_PLS_0

```
#define PWR_CR_PLS_0 ((uint32_t)0x00000020)
```

Bit 0

5.173.2.1957 PWR_CR_PLS_1

```
#define PWR_CR_PLS_1 ((uint32_t)0x00000040)
```

Bit 1

5.173.2.1958 PWR_CR_PLS_2

```
#define PWR_CR_PLS_2 ((uint32_t)0x00000080)
```

Bit 2 PVD level configuration

5.173.2.1959 PWR_CR_PLS_LEV0

```
#define PWR_CR_PLS_LEV0 ((uint32_t)0x00000000)
```

PVD level 0

5.173.2.1960 PWR_CR_PLS_LEV1

```
#define PWR_CR_PLS_LEV1 ((uint32_t)0x00000020)
```

PVD level 1

5.173.2.1961 PWR_CR_PLS_LEV2

```
#define PWR_CR_PLSLEV2 ((uint32_t)0x00000040)
```

PVD level 2

5.173.2.1962 PWR_CR_PLS_LEV3

```
#define PWR_CR_PLSLEV3 ((uint32_t)0x00000060)
```

PVD level 3

5.173.2.1963 PWR_CR_PLS_LEV4

```
#define PWR_CR_PLSLEV4 ((uint32_t)0x00000080)
```

PVD level 4

5.173.2.1964 PWR_CR_PLS_LEV5

```
#define PWR_CR_PLSLEV5 ((uint32_t)0x000000A0)
```

PVD level 5

5.173.2.1965 PWR_CR_PLS_LEV6

```
#define PWR_CR_PLSLEV6 ((uint32_t)0x000000C0)
```

PVD level 6

5.173.2.1966 PWR_CR_PLS_LEV7

```
#define PWR_CR_PLSLEV7 ((uint32_t)0x000000E0)
```

PVD level 7

5.173.2.1967 PWR_CR_PVDE

```
#define PWR_CR_PVDE ((uint32_t)0x00000010)
```

Power Voltage Detector Enable

5.173.2.1968 PWR_CR_UDEN

```
#define PWR_CR_UDEN ((uint32_t)0x000C0000)
```

Under Drive enable in stop mode

5.173.2.1969 PWR_CR_UDEN_0

```
#define PWR_CR_UDEN_0 ((uint32_t)0x00040000)
```

Bit 0

5.173.2.1970 PWR_CR_UDEN_1

```
#define PWR_CR_UDEN_1 ((uint32_t)0x00080000)
```

Bit 1

5.173.2.1971 PWR_CR_VOS

```
#define PWR_CR_VOS ((uint32_t)0x0000C000)
```

VOS[1:0] bits (Regulator voltage scaling output selection)

5.173.2.1972 PWR_CR_VOS_0

```
#define PWR_CR_VOS_0 ((uint32_t)0x00004000)
```

Bit 0

5.173.2.1973 PWR_CR_VOS_1

```
#define PWR_CR_VOS_1 ((uint32_t)0x00008000)
```

Bit 1

5.173.2.1974 PWR_CSR_BRE

```
#define PWR_CSR_BRE ((uint32_t)0x00000200)
```

Backup regulator enable

5.173.2.1975 PWR_CSR_BRR

```
#define PWR_CSR_BRR ((uint32_t)0x00000008)
```

Backup regulator ready

5.173.2.1976 PWR_CSR_EWUP

```
#define PWR_CSR_EWUP ((uint32_t)0x000000100)
```

Enable WKUP pin

5.173.2.1977 PWR_CSR_ODRDY

```
#define PWR_CSR_ODRDY ((uint32_t)0x00010000)
```

Over Drive generator ready

5.173.2.1978 PWR_CSR_ODSWRDY

```
#define PWR_CSR_ODSWRDY ((uint32_t)0x00020000)
```

Over Drive Switch ready

5.173.2.1979 PWR_CSR_PVDO

```
#define PWR_CSR_PVDO ((uint32_t)0x00000004)
```

PVD Output

5.173.2.1980 PWR_CSR_SBF

```
#define PWR_CSR_SBF ((uint32_t)0x00000002)
```

Standby Flag

5.173.2.1981 PWR_CSR_UDSWRDY

```
#define PWR_CSR_UDSWRDY ((uint32_t)0x000C0000)
```

Under Drive ready

5.173.2.1982 PWR_CSR_VOSRDY

```
#define PWR_CSR_VOSRDY ((uint32_t)0x00004000)
```

Regulator voltage scaling output selection ready

5.173.2.1983 PWR_CSR_WUF

```
#define PWR_CSR_WUF ((uint32_t)0x00000001)
```

Wakeup Flag

5.173.2.1984 PWR_CSR_WUPP

```
#define PWR_CSR_WUPP ((uint32_t)0x00000080)
```

WKUP pin Polarity

5.173.2.1985 RCC_CFGR_HPRE

```
#define RCC_CFGR_HPRE ((uint32_t)0x000000F0)
```

< HPRE configuration HPRE[3:0] bits (AHB prescaler)

5.173.2.1986 RCC_CFGR_HPRE_0

```
#define RCC_CFGR_HPRE_0 ((uint32_t)0x00000010)
```

Bit 0

5.173.2.1987 RCC_CFGR_HPRE_1

```
#define RCC_CFGR_HPRE_1 ((uint32_t)0x00000020)
```

Bit 1

5.173.2.1988 RCC_CFGR_HPRE_2

```
#define RCC_CFGR_HPRE_2 ((uint32_t)0x00000040)
```

Bit 2

5.173.2.1989 RCC_CFGR_HPREF_3

```
#define RCC_CFGR_HPREF_3 ((uint32_t)0x00000080)
```

Bit 3

5.173.2.1990 RCC_CFGR_HPREF_DIV1

```
#define RCC_CFGR_HPREF_DIV1 ((uint32_t)0x00000000)
```

SYSCLK not divided

5.173.2.1991 RCC_CFGR_HPREF_DIV128

```
#define RCC_CFGR_HPREF_DIV128 ((uint32_t)0x000000D0)
```

SYSCLK divided by 128

5.173.2.1992 RCC_CFGR_HPREF_DIV16

```
#define RCC_CFGR_HPREF_DIV16 ((uint32_t)0x000000B0)
```

SYSCLK divided by 16

5.173.2.1993 RCC_CFGR_HPREF_DIV2

```
#define RCC_CFGR_HPREF_DIV2 ((uint32_t)0x00000080)
```

SYSCLK divided by 2

5.173.2.1994 RCC_CFGR_HPREF_DIV256

```
#define RCC_CFGR_HPREF_DIV256 ((uint32_t)0x000000E0)
```

SYSCLK divided by 256

5.173.2.1995 RCC_CFGR_HPREF_DIV4

```
#define RCC_CFGR_HPREF_DIV4 ((uint32_t)0x00000090)
```

SYSCLK divided by 4

5.173.2.1996 RCC_CFGR_HPREF_DIV512

```
#define RCC_CFGR_HPREF_DIV512 ((uint32_t)0x000000F0)
```

SYSCLK divided by 512

5.173.2.1997 RCC_CFGR_HPRE_DIV64

```
#define RCC_CFGR_HPRE_DIV64 ((uint32_t)0x000000C0)
```

SYSCLK divided by 64

5.173.2.1998 RCC_CFGR_HPRE_DIV8

```
#define RCC_CFGR_HPRE_DIV8 ((uint32_t)0x000000A0)
```

SYSCLK divided by 8

5.173.2.1999 RCC_CFGR_PPREG1

```
#define RCC_CFGR_PPREG1 ((uint32_t)0x00001C00)
```

< PPREG1 configuration PRE1[2:0] bits (APB1 prescaler)

5.173.2.2000 RCC_CFGR_PPREG1_0

```
#define RCC_CFGR_PPREG1_0 ((uint32_t)0x00000400)
```

Bit 0

5.173.2.2001 RCC_CFGR_PPREG1_1

```
#define RCC_CFGR_PPREG1_1 ((uint32_t)0x00000800)
```

Bit 1

5.173.2.2002 RCC_CFGR_PPREG1_2

```
#define RCC_CFGR_PPREG1_2 ((uint32_t)0x00001000)
```

Bit 2

5.173.2.2003 RCC_CFGR_PPREG1_DIV1

```
#define RCC_CFGR_PPREG1_DIV1 ((uint32_t)0x00000000)
```

HCLK not divided

5.173.2.2004 RCC_CFGR_PPREG1_DIV16

```
#define RCC_CFGR_PPREG1_DIV16 ((uint32_t)0x00001C00)
```

HCLK divided by 16 PPREG2 configuration

5.173.2.2005 RCC_CFGR_PPREG1_DIV2

```
#define RCC_CFGR_PPREG1_DIV2 ((uint32_t)0x00001000)
```

HCLK divided by 2

5.173.2.2006 RCC_CFGR_PPREG1_DIV4

```
#define RCC_CFGR_PPREG1_DIV4 ((uint32_t)0x00001400)
```

HCLK divided by 4

5.173.2.2007 RCC_CFGR_PPREG1_DIV8

```
#define RCC_CFGR_PPREG1_DIV8 ((uint32_t)0x00001800)
```

HCLK divided by 8

5.173.2.2008 RCC_CFGR_PPREG2

```
#define RCC_CFGR_PPREG2 ((uint32_t)0x0000E000)
```

PRE2[2:0] bits (APB2 prescaler)

5.173.2.2009 RCC_CFGR_PPREG2_0

```
#define RCC_CFGR_PPREG2_0 ((uint32_t)0x00002000)
```

Bit 0

5.173.2.2010 RCC_CFGR_PPREG2_1

```
#define RCC_CFGR_PPREG2_1 ((uint32_t)0x00004000)
```

Bit 1

5.173.2.2011 RCC_CFGR_PPREG2_2

```
#define RCC_CFGR_PPREG2_2 ((uint32_t)0x00008000)
```

Bit 2

5.173.2.2012 RCC_CFGR_PPREG2_DIV1

```
#define RCC_CFGR_PPREG2_DIV1 ((uint32_t)0x00000000)
```

HCLK not divided

5.173.2.2013 RCC_CFGR_PPREG_DIV16

```
#define RCC_CFGR_PPREG_DIV16 ((uint32_t)0x0000E000)
```

HCLK divided by 16 RTCPRE configuration

5.173.2.2014 RCC_CFGR_PPREG_DIV2

```
#define RCC_CFGR_PPREG_DIV2 ((uint32_t)0x00008000)
```

HCLK divided by 2

5.173.2.2015 RCC_CFGR_PPREG_DIV4

```
#define RCC_CFGR_PPREG_DIV4 ((uint32_t)0x0000A000)
```

HCLK divided by 4

5.173.2.2016 RCC_CFGR_PPREG_DIV8

```
#define RCC_CFGR_PPREG_DIV8 ((uint32_t)0x0000C000)
```

HCLK divided by 8

5.173.2.2017 RCC_CFGR_RTCPRE_4

```
#define RCC_CFGR_RTCPRE_4 ((uint32_t)0x00100000)
```

MCO1 configuration

5.173.2.2018 RCC_CFGR_SW

```
#define RCC_CFGR_SW ((uint32_t)0x00000003)
```

< SW configuration SW[1:0] bits (System clock Switch)

5.173.2.2019 RCC_CFGR_SW_0

```
#define RCC_CFGR_SW_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.2020 RCC_CFGR_SW_1

```
#define RCC_CFGR_SW_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.2021 RCC_CFGR_SW_HSE

```
#define RCC_CFGR_SW_HSE ((uint32_t)0x00000001)
```

HSE selected as system clock

5.173.2.2022 RCC_CFGR_SW_HSI

```
#define RCC_CFGR_SW_HSI ((uint32_t)0x00000000)
```

HSI selected as system clock

5.173.2.2023 RCC_CFGR_SW_PLL

```
#define RCC_CFGR_SW_PLL ((uint32_t)0x00000002)
```

PLL/PLLP selected as system clock

5.173.2.2024 RCC_CFGR_SWS

```
#define RCC_CFGR_SWS ((uint32_t)0x0000000C)
```

< SWS configuration SWS[1:0] bits (System Clock Switch Status)

5.173.2.2025 RCC_CFGR_SWS_0

```
#define RCC_CFGR_SWS_0 ((uint32_t)0x00000004)
```

Bit 0

5.173.2.2026 RCC_CFGR_SWS_1

```
#define RCC_CFGR_SWS_1 ((uint32_t)0x00000008)
```

Bit 1

5.173.2.2027 RCC_CFGR_SWS_HSE

```
#define RCC_CFGR_SWS_HSE ((uint32_t)0x00000004)
```

HSE oscillator used as system clock

5.173.2.2028 RCC_CFGR_SWS_HSI

```
#define RCC_CFGR_SWS_HSI ((uint32_t)0x00000000)
```

HSI oscillator used as system clock

5.173.2.2029 RCC_CFGR_SWS_PLL

```
#define RCC_CFGR_SWS_PLL ((uint32_t)0x00000008)
```

PLL/PLLP used as system clock

5.173.2.2030 RCC_CR_HSICAL_0

```
#define RCC_CR_HSICAL_0 ((uint32_t)0x00000100)
```

Bit 0

5.173.2.2031 RCC_CR_HSICAL_1

```
#define RCC_CR_HSICAL_1 ((uint32_t)0x00000200)
```

Bit 1

5.173.2.2032 RCC_CR_HSICAL_2

```
#define RCC_CR_HSICAL_2 ((uint32_t)0x00000400)
```

Bit 2

5.173.2.2033 RCC_CR_HSICAL_3

```
#define RCC_CR_HSICAL_3 ((uint32_t)0x00000800)
```

Bit 3

5.173.2.2034 RCC_CR_HSICAL_4

```
#define RCC_CR_HSICAL_4 ((uint32_t)0x00001000)
```

Bit 4

5.173.2.2035 RCC_CR_HSICAL_5

```
#define RCC_CR_HSICAL_5 ((uint32_t)0x00002000)
```

Bit 5

5.173.2.2036 RCC_CR_HSICAL_6

```
#define RCC_CR_HSICAL_6 ((uint32_t)0x00004000)
```

Bit 6

5.173.2.2037 RCC_CR_HSICAL_7

```
#define RCC_CR_HSICAL_7 ((uint32_t)0x00008000)
```

Bit 7

5.173.2.2038 RCC_CR_HSITRIM_0

```
#define RCC_CR_HSITRIM_0 ((uint32_t)0x00000008)
```

Bit 0

5.173.2.2039 RCC_CR_HSITRIM_1

```
#define RCC_CR_HSITRIM_1 ((uint32_t)0x00000010)
```

Bit 1

5.173.2.2040 RCC_CR_HSITRIM_2

```
#define RCC_CR_HSITRIM_2 ((uint32_t)0x00000020)
```

Bit 2

5.173.2.2041 RCC_CR_HSITRIM_3

```
#define RCC_CR_HSITRIM_3 ((uint32_t)0x00000040)
```

Bit 3

5.173.2.2042 RCC_CR_HSITRIM_4

```
#define RCC_CR_HSITRIM_4 ((uint32_t)0x00000080)
```

Bit 4

5.173.2.2043 SAI_GCR_SYNCIN

```
#define SAI_GCR_SYNCIN ((uint32_t)0x00000003)
```

SYNCIN[1:0] bits (Synchronization Inputs)

5.173.2.2044 SAI_GCR_SYNCIN_0

```
#define SAI_GCR_SYNCIN_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.2045 SAI_GCR_SYNCIN_1

```
#define SAI_GCR_SYNCIN_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.2046 SAI_GCR_SYNCOUT

```
#define SAI_GCR_SYNCOUT ((uint32_t)0x00000030)
```

SYNCOUT[1:0] bits (Synchronization Outputs)

5.173.2.2047 SAI_GCR_SYNCOUT_0

```
#define SAI_GCR_SYNCOUT_0 ((uint32_t)0x00000010)
```

Bit 0

5.173.2.2048 SAI_GCR_SYNCOUT_1

```
#define SAI_GCR_SYNCOUT_1 ((uint32_t)0x00000020)
```

Bit 1

5.173.2.2049 SAI_xCLRFR_CAFSDET

```
#define SAI_xCLRFR_CAFSDET ((uint32_t)0x00000020)
```

Clear Anticipated frame synchronization detection

5.173.2.2050 SAI_xCLRFR_CCNRDY

```
#define SAI_xCLRFR_CCNRDY ((uint32_t)0x00000010)
```

Clear Codec not ready

5.173.2.2051 SAI_xCLRFR_CFREQ

```
#define SAI_xCLRFR_CFREQ ((uint32_t)0x00000008)
```

Clear FIFO request

5.173.2.2052 SAI_xCLRFR_CLFSDET

```
#define SAI_xCLRFR_CLFSDET ((uint32_t)0x00000040)
```

Clear Late frame synchronization detection

5.173.2.2053 SAI_xCLRFR_CMUTEDET

```
#define SAI_xCLRFR_CMUTEDET ((uint32_t)0x00000002)
```

Clear Mute detection

5.173.2.2054 SAI_xCLRFR_COVRUDR

```
#define SAI_xCLRFR_COVRUDR ((uint32_t)0x00000001)
```

Clear Overrun underrun

5.173.2.2055 SAI_xCLRFR_CWCKCFG

```
#define SAI_xCLRFR_CWCKCFG ((uint32_t)0x00000004)
```

Clear Wrong Clock Configuration

5.173.2.2056 SAI_xCR1_CKSTR

```
#define SAI_xCR1_CKSTR ((uint32_t)0x00000200)
```

ClocK STRobing edge

5.173.2.2057 SAI_xCR1_DMAEN

```
#define SAI_xCR1_DMAEN ((uint32_t)0x00020000)
```

DMA enable

5.173.2.2058 SAI_xCR1_DS

```
#define SAI_xCR1_DS ((uint32_t)0x000000E0)
```

DS[1:0] bits (Data Size)

5.173.2.2059 SAI_xCR1_DS_0

```
#define SAI_xCR1_DS_0 ((uint32_t)0x00000020)
```

Bit 0

5.173.2.2060 SAI_xCR1_DS_1

```
#define SAI_xCR1_DS_1 ((uint32_t)0x00000040)
```

Bit 1

5.173.2.2061 SAI_xCR1_DS_2

```
#define SAI_xCR1_DS_2 ((uint32_t)0x00000080)
```

Bit 2

5.173.2.2062 SAI_xCR1_LSBFIRST

```
#define SAI_xCR1_LSBFIRST ((uint32_t)0x00000100)
```

LSB First Configuration

5.173.2.2063 SAI_xCR1_MCKDIV

```
#define SAI_xCR1_MCKDIV ((uint32_t)0x00780000)
```

MCKDIV[3:0] (Master ClocK Divider)

5.173.2.2064 SAI_xCR1_MCKDIV_0

```
#define SAI_xCR1_MCKDIV_0 ((uint32_t)0x00080000)
```

Bit 0

5.173.2.2065 SAI_xCR1_MCKDIV_1

```
#define SAI_xCR1_MCKDIV_1 ((uint32_t)0x00100000)
```

Bit 1

5.173.2.2066 SAI_xCR1_MCKDIV_2

```
#define SAI_xCR1_MCKDIV_2 ((uint32_t)0x00200000)
```

Bit 2

5.173.2.2067 SAI_xCR1_MCKDIV_3

```
#define SAI_xCR1_MCKDIV_3 ((uint32_t)0x00400000)
```

Bit 3

5.173.2.2068 SAI_xCR1_MODE

```
#define SAI_xCR1_MODE ((uint32_t)0x00000003)
```

MODE[1:0] bits (Audio Block Mode)

5.173.2.2069 SAI_xCR1_MODE_0

```
#define SAI_xCR1_MODE_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.2070 SAI_xCR1_MODE_1

```
#define SAI_xCR1_MODE_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.2071 SAI_xCR1_MONO

```
#define SAI_xCR1_MONO ((uint32_t)0x00001000)
```

Mono mode

5.173.2.2072 SAI_xCR1_NODIV

```
#define SAI_xCR1_NODIV ((uint32_t)0x00080000)
```

No Divider Configuration

5.173.2.2073 SAI_xCR1_OUTDRIV

```
#define SAI_xCR1_OUTDRIV ((uint32_t)0x00002000)
```

Output Drive

5.173.2.2074 SAI_xCR1_PRTCFG

```
#define SAI_xCR1_PRTCFG ((uint32_t)0x0000000C)
```

PRTCFG[1:0] bits (Protocol Configuration)

5.173.2.2075 SAI_xCR1_PRTCFG_0

```
#define SAI_xCR1_PRTCFG_0 ((uint32_t)0x00000004)
```

Bit 0

5.173.2.2076 SAI_xCR1_PRTCFG_1

```
#define SAI_xCR1_PRTCFG_1 ((uint32_t)0x00000008)
```

Bit 1

5.173.2.2077 SAI_xCR1_SAIEN

```
#define SAI_xCR1_SAIEN ((uint32_t)0x00010000)
```

Audio Block enable

5.173.2.2078 SAI_xCR1_SYNCEN

```
#define SAI_xCR1_SYNCEN ((uint32_t)0x00000C00)
```

SYNCEN[1:0](SYNChronization ENable)

5.173.2.2079 SAI_xCR1_SYNCEN_0

```
#define SAI_xCR1_SYNCEN_0 ((uint32_t)0x00000400)
```

Bit 0

5.173.2.2080 SAI_xCR1_SYNCEN_1

```
#define SAI_xCR1_SYNCEN_1 ((uint32_t)0x00000800)
```

Bit 1

5.173.2.2081 SAI_xCR2_COMP

```
#define SAI_xCR2_COMP ((uint32_t)0x0000C000)
```

COMP[1:0] (Companding mode)

5.173.2.2082 SAI_xCR2_COMP_0

```
#define SAI_xCR2_COMP_0 ((uint32_t)0x00004000)
```

Bit 0

5.173.2.2083 SAI_xCR2_COMP_1

```
#define SAI_xCR2_COMP_1 ((uint32_t)0x00008000)
```

Bit 1

5.173.2.2084 SAI_xCR2_CPL

```
#define SAI_xCR2_CPL ((uint32_t)0x00002000)
```

Complement Bit

5.173.2.2085 SAI_xCR2_FFLUSH

```
#define SAI_xCR2_FFLUSH ((uint32_t)0x00000008)
```

Fifo FLUSH

5.173.2.2086 SAI_xCR2_FTH

```
#define SAI_xCR2_FTH ((uint32_t)0x00000003)

FTH[1:0](Fifo THreshold)
```

5.173.2.2087 SAI_xCR2_FTH_0

```
#define SAI_xCR2_FTH_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.2088 SAI_xCR2_FTH_1

```
#define SAI_xCR2_FTH_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.2089 SAI_xCR2_MUTE

```
#define SAI_xCR2_MUTE ((uint32_t)0x00000020)
```

Mute mode

5.173.2.2090 SAI_xCR2_MUTECNT

```
#define SAI_xCR2_MUTECNT ((uint32_t)0x00001F80)
```

MUTECNT[5:0] (MUTE counter)

5.173.2.2091 SAI_xCR2_MUTECNT_0

```
#define SAI_xCR2_MUTECNT_0 ((uint32_t)0x00000080)
```

Bit 0

5.173.2.2092 SAI_xCR2_MUTECNT_1

```
#define SAI_xCR2_MUTECNT_1 ((uint32_t)0x00000100)
```

Bit 1

5.173.2.2093 SAI_xCR2_MUTECNT_2

```
#define SAI_xCR2_MUTECNT_2 ((uint32_t)0x00000200)
```

Bit 2

5.173.2.2094 SAI_xCR2_MUTECNT_3

```
#define SAI_xCR2_MUTECNT_3 ((uint32_t)0x00000400)
```

Bit 3

5.173.2.2095 SAI_xCR2_MUTECNT_4

```
#define SAI_xCR2_MUTECNT_4 ((uint32_t)0x00000800)
```

Bit 4

5.173.2.2096 SAI_xCR2_MUTECNT_5

```
#define SAI_xCR2_MUTECNT_5 ((uint32_t)0x00001000)
```

Bit 5

5.173.2.2097 SAI_xCR2_MUTEVAL

```
#define SAI_xCR2_MUTEVAL ((uint32_t)0x00000040)
```

Muate value

5.173.2.2098 SAI_xCR2_TRIS

```
#define SAI_xCR2_TRIS ((uint32_t)0x00000010)
```

TRISState Management on data line

5.173.2.2099 SAI_xFRCR_FRL

```
#define SAI_xFRCR_FRL ((uint32_t)0x000000FF)
```

FRL[1:0](Frame length)

5.173.2.2100 SAI_xFRCR_FRL_0

```
#define SAI_xFRCR_FRL_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.2101 SAI_xFRCR_FRL_1

```
#define SAI_xFRCR_FRL_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.2102 SAI_xFRCR_FRL_2

```
#define SAI_xFRCR_FRL_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.2103 SAI_xFRCR_FRL_3

```
#define SAI_xFRCR_FRL_3 ((uint32_t)0x00000008)
```

Bit 3

5.173.2.2104 SAI_xFRCR_FRL_4

```
#define SAI_xFRCR_FRL_4 ((uint32_t)0x00000010)
```

Bit 4

5.173.2.2105 SAI_xFRCR_FRL_5

```
#define SAI_xFRCR_FRL_5 ((uint32_t)0x00000020)
```

Bit 5

5.173.2.2106 SAI_xFRCR_FRL_6

```
#define SAI_xFRCR_FRL_6 ((uint32_t)0x00000040)
```

Bit 6

5.173.2.2107 SAI_xFRCR_FRL_7

```
#define SAI_xFRCR_FRL_7 ((uint32_t)0x00000080)
```

Bit 7

5.173.2.2108 SAI_xFRCR_FSALL

```
#define SAI_xFRCR_FSALL ((uint32_t)0x00007F00)
```

FRL[1:0] (Frame synchronization active level length)

5.173.2.2109 SAI_xFRCR_FSALL_0

```
#define SAI_xFRCR_FSALL_0 ((uint32_t)0x00000100)
```

Bit 0

5.173.2.2110 SAI_xFRCR_FSALL_1

```
#define SAI_xFRCR_FSALL_1 ((uint32_t)0x00000200)
```

Bit 1

5.173.2.2111 SAI_xFRCR_FSALL_2

```
#define SAI_xFRCR_FSALL_2 ((uint32_t)0x00000400)
```

Bit 2

5.173.2.2112 SAI_xFRCR_FSALL_3

```
#define SAI_xFRCR_FSALL_3 ((uint32_t)0x00000800)
```

Bit 3

5.173.2.2113 SAI_xFRCR_FSALL_4

```
#define SAI_xFRCR_FSALL_4 ((uint32_t)0x00001000)
```

Bit 4

5.173.2.2114 SAI_xFRCR_FSALL_5

```
#define SAI_xFRCR_FSALL_5 ((uint32_t)0x00002000)
```

Bit 5

5.173.2.2115 SAI_xFRCR_FSALL_6

```
#define SAI_xFRCR_FSALL_6 ((uint32_t)0x00004000)
```

Bit 6

5.173.2.2116 SAI_xFRCR_FSDEF

```
#define SAI_xFRCR_FSDEF ((uint32_t)0x00010000)
```

Frame Synchronization Definition

5.173.2.2117 SAI_xFRCR_FSOFF

```
#define SAI_xFRCR_FSOFF ((uint32_t)0x00040000)
```

Frame Synchronization OFFset

5.173.2.2118 SAI_xFRCR_FSPOL

```
#define SAI_xFRCR_FSPOL ((uint32_t)0x00020000)
```

Frame Synchronization POLarity

5.173.2.2119 SAI_xIMR_AFSDETIE

```
#define SAI_xIMR_AFSDETIE ((uint32_t)0x00000020)
```

Anticipated frame synchronization detection interrupt enable

5.173.2.2120 SAI_xIMR_CNRDYIE

```
#define SAI_xIMR_CNRDYIE ((uint32_t)0x00000010)
```

Codec not ready interrupt enable

5.173.2.2121 SAI_xIMR_FREQIE

```
#define SAI_xIMR_FREQIE ((uint32_t)0x00000008)
```

FIFO request interrupt enable

5.173.2.2122 SAI_xIMR_LFSDETIE

```
#define SAI_xIMR_LFSDETIE ((uint32_t)0x00000040)
```

Late frame synchronization detection interrupt enable

5.173.2.2123 SAI_xIMR_MUTEDETIE

```
#define SAI_xIMR_MUTEDETIE ((uint32_t)0x00000002)
```

Mute detection interrupt enable

5.173.2.2124 SAI_xIMR_OVRUDRIE

```
#define SAI_xIMR_OVRUDRIE ((uint32_t)0x00000001)
```

Overrun underrun interrupt enable

5.173.2.2125 SAI_xIMR_WCKCFGIE

```
#define SAI_xIMR_WCKCFGIE ((uint32_t)0x00000004)
```

Wrong Clock Configuration interrupt enable

5.173.2.2126 SAI_xSLOTR_FBOFF

```
#define SAI_xSLOTR_FBOFF ((uint32_t)0x0000001F)
```

FRL[4:0](First Bit Offset)

5.173.2.2127 SAI_xSLOTR_FBOFF_0

```
#define SAI_xSLOTR_FBOFF_0 ((uint32_t)0x00000001)
```

Bit 0

5.173.2.2128 SAI_xSLOTR_FBOFF_1

```
#define SAI_xSLOTR_FBOFF_1 ((uint32_t)0x00000002)
```

Bit 1

5.173.2.2129 SAI_xSLOTR_FBOFF_2

```
#define SAI_xSLOTR_FBOFF_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.2130 SAI_xSLOTR_FBOFF_3

```
#define SAI_xSLOTR_FBOFF_3 ((uint32_t)0x00000008)
```

Bit 3

5.173.2.2131 SAI_xSLOTR_FBOFF_4

```
#define SAI_xSLOTR_FBOFF_4 ((uint32_t)0x00000010)
```

Bit 4

5.173.2.2132 SAI_xSLOTR_NBSLOT

```
#define SAI_xSLOTR_NBSLOT ((uint32_t)0x00000F00)
```

NBSLOT[3:0] (Number of Slot in audio Frame)

5.173.2.2133 SAI_xSLOTR_NBSLOT_0

```
#define SAI_xSLOTR_NBSLOT_0 ((uint32_t)0x00000100)
```

Bit 0

5.173.2.2134 SAI_xSLOTR_NBSLOT_1

```
#define SAI_xSLOTR_NBSLOT_1 ((uint32_t)0x00000200)
```

Bit 1

5.173.2.2135 SAI_xSLOTR_NBSLOT_2

```
#define SAI_xSLOTR_NBSLOT_2 ((uint32_t)0x00000400)
```

Bit 2

5.173.2.2136 SAI_xSLOTR_NBSLOT_3

```
#define SAI_xSLOTR_NBSLOT_3 ((uint32_t)0x00000800)
```

Bit 3

5.173.2.2137 SAI_xSLOTR_SLOTEN

```
#define SAI_xSLOTR_SLOTEN ((uint32_t)0xFFFF0000)
```

SLOTEN[15:0] (Slot Enable)

5.173.2.2138 SAI_xSLOTR_SLOTSZ

```
#define SAI_xSLOTR_SLOTSZ ((uint32_t)0x000000C0)
```

SLOTSZ[1:0] (Slot size)

5.173.2.2139 SAI_xSLOTR_SLOTSZ_0

```
#define SAI_xSLOTR_SLOTSZ_0 ((uint32_t)0x00000040)
```

Bit 0

5.173.2.2140 SAI_xSLOTR_SLOTSZ_1

```
#define SAI_xSLOTR_SLOTSZ_1 ((uint32_t)0x00000080)
```

Bit 1

5.173.2.2141 SAI_xSR_AFSDET

```
#define SAI_xSR_AFSDET ((uint32_t)0x00000020)
```

Anticipated frame synchronization detection

5.173.2.2142 SAI_xSR_CNRDY

```
#define SAI_xSR_CNRDY ((uint32_t)0x00000010)
```

Codec not ready

5.173.2.2143 SAI_xSR_FLVL

```
#define SAI_xSR_FLVL ((uint32_t)0x00070000)
```

FLVL[2:0] (FIFO Level Threshold)

5.173.2.2144 SAI_xSR_FLVL_0

```
#define SAI_xSR_FLVL_0 ((uint32_t)0x00010000)
```

Bit 0

5.173.2.2145 SAI_xSR_FLVL_1

```
#define SAI_xSR_FLVL_1 ((uint32_t)0x00020000)
```

Bit 1

5.173.2.2146 SAI_xSR_FLVL_2

```
#define SAI_xSR_FLVL_2 ((uint32_t)0x00030000)
```

Bit 2

5.173.2.2147 SAI_xSR_FREQ

```
#define SAI_xSR_FREQ ((uint32_t)0x00000008)
```

FIFO request

5.173.2.2148 SAI_xSR_LFSDET

```
#define SAI_xSR_LFSDET ((uint32_t)0x00000040)
```

Late frame synchronization detection

5.173.2.2149 SAI_xSR_MUTEDET

```
#define SAI_xSR_MUTEDET ((uint32_t)0x00000002)
```

Mute detection

5.173.2.2150 SAI_xSR_OVRUDR

```
#define SAI_xSR_OVRUDR ((uint32_t)0x00000001)
```

Overrun underrun

5.173.2.2151 SAI_xSR_WCKCFG

```
#define SAI_xSR_WCKCFG ((uint32_t)0x00000004)
```

Wrong Clock Configuration

5.173.2.2152 SDIO_ARG_CMDARG

```
#define SDIO_ARG_CMDARG ((uint32_t)0xFFFFFFFF)
```

Command argument

5.173.2.2153 SDIO_CLKCR_BYPASS

```
#define SDIO_CLKCR_BYPASS ((uint16_t)0x0400)
```

Clock divider bypass enable bit

5.173.2.2154 SDIO_CLKCR_CLKDIV

```
#define SDIO_CLKCR_CLKDIV ((uint16_t)0x00FF)
```

Clock divide factor

5.173.2.2155 SDIO_CLKCR_CLKEN

```
#define SDIO_CLKCR_CLKEN ((uint16_t)0x0100)
```

Clock enable bit

5.173.2.2156 SDIO_CLKCR_HWFC_EN

```
#define SDIO_CLKCR_HWFC_EN ((uint16_t)0x4000)
```

HW Flow Control enable

5.173.2.2157 SDIO_CLKCR_NEGEDGE

```
#define SDIO_CLKCR_NEGEDGE ((uint16_t)0x2000)
```

SDIO_CK dephasing selection bit

5.173.2.2158 SDIO_CLKCR_PWRSAV

```
#define SDIO_CLKCR_PWRSAV ((uint16_t)0x0200)
```

Power saving configuration bit

5.173.2.2159 SDIO_CLKCR_WIDBUS

```
#define SDIO_CLKCR_WIDBUS ((uint16_t)0x1800)
```

WIDBUS[1:0] bits (Wide bus mode enable bit)

5.173.2.2160 SDIO_CLKCR_WIDBUS_0

```
#define SDIO_CLKCR_WIDBUS_0 ((uint16_t)0x0800)
```

Bit 0

5.173.2.2161 SDIO_CLKCR_WIDBUS_1

```
#define SDIO_CLKCR_WIDBUS_1 ((uint16_t)0x1000)
```

Bit 1

5.173.2.2162 SDIO_CMD_CEATACMD

```
#define SDIO_CMD_CEATACMD ((uint16_t)0x4000)
```

CE-ATA command

5.173.2.2163 SDIO_CMD_CMDINDEX

```
#define SDIO_CMD_CMDINDEX ((uint16_t)0x003F)
```

Command Index

5.173.2.2164 SDIO_CMD_CPSMEN

```
#define SDIO_CMD_CPSMEN ((uint16_t)0x0400)
```

Command path state machine (CPSM) Enable bit

5.173.2.2165 SDIO_CMD_ENCMDCOMPL

```
#define SDIO_CMD_ENCMDCOMPL ((uint16_t)0x1000)
```

Enable CMD completion

5.173.2.2166 SDIO_CMD_NIEN

```
#define SDIO_CMD_NIEN ((uint16_t)0x2000)
```

Not Interrupt Enable

5.173.2.2167 SDIO_CMD_SDIOSUSPEND

```
#define SDIO_CMD_SDIOSUSPEND ((uint16_t)0x0800)
```

SD I/O suspend command

5.173.2.2168 SDIO_CMD_WAITINT

```
#define SDIO_CMD_WAITINT ((uint16_t)0x0100)
```

CPSM Waits for Interrupt Request

5.173.2.2169 SDIO_CMD_WAITPEND

```
#define SDIO_CMD_WAITPEND ((uint16_t)0x0200)
```

CPSM Waits for ends of data transfer (CmdPend internal signal)

5.173.2.2170 SDIO_CMD_WAITRESP

```
#define SDIO_CMD_WAITRESP ((uint16_t)0x00C0)
```

WAITRESP[1:0] bits (Wait for response bits)

5.173.2.2171 SDIO_CMD_WAITRESP_0

```
#define SDIO_CMD_WAITRESP_0 ((uint16_t)0x0040)
```

Bit 0

5.173.2.2172 SDIO_CMD_WAITRESP_1

```
#define SDIO_CMD_WAITRESP_1 ((uint16_t)0x0080)
```

Bit 1

5.173.2.2173 SDIO_DCOUNT_DATACOUNT

```
#define SDIO_DCOUNT_DATACOUNT ((uint32_t)0x01FFFFFF)
```

Data count value

5.173.2.2174 SDIO_DCTRL_DBLOCKSIZE

```
#define SDIO_DCTRL_DBLOCKSIZE ((uint16_t)0x00F0)
```

DBLOCKSIZE[3:0] bits (Data block size)

5.173.2.2175 SDIO_DCTRL_DBLOCKSIZE_0

```
#define SDIO_DCTRL_DBLOCKSIZE_0 ((uint16_t)0x0010)
```

Bit 0

5.173.2.2176 SDIO_DCTRL_DBLOCKSIZE_1

```
#define SDIO_DCTRL_DBLOCKSIZE_1 ((uint16_t)0x0020)
```

Bit 1

5.173.2.2177 SDIO_DCTRL_DBLOCKSIZE_2

```
#define SDIO_DCTRL_DBLOCKSIZE_2 ((uint16_t)0x0040)
```

Bit 2

5.173.2.2178 SDIO_DCTRL_DBLOCKSIZE_3

```
#define SDIO_DCTRL_DBLOCKSIZE_3 ((uint16_t)0x0080)
```

Bit 3

5.173.2.2179 SDIO_DCTRL_DMAEN

```
#define SDIO_DCTRL_DMAEN ((uint16_t)0x0008)
```

DMA enabled bit

5.173.2.2180 SDIO_DCTRL_DTDIR

```
#define SDIO_DCTRL_DTDIR ((uint16_t)0x0002)
```

Data transfer direction selection

5.173.2.2181 SDIO_DCTRL_DTEN

```
#define SDIO_DCTRL_DTEN ((uint16_t)0x0001)
```

Data transfer enabled bit

5.173.2.2182 SDIO_DCTRL_DTMODE

```
#define SDIO_DCTRL_DTMODE ((uint16_t)0x0004)
```

Data transfer mode selection

5.173.2.2183 SDIO_DCTRL_RWMOD

```
#define SDIO_DCTRL_RWMOD ((uint16_t)0x0400)
```

Read wait mode

5.173.2.2184 SDIO_DCTRL_RWSTART

```
#define SDIO_DCTRL_RWSTART ((uint16_t)0x0100)
```

Read wait start

5.173.2.2185 SDIO_DCTRL_RWSTOP

```
#define SDIO_DCTRL_RWSTOP ((uint16_t)0x0200)
```

Read wait stop

5.173.2.2186 SDIO_DCTRL_SDIOEN

```
#define SDIO_DCTRL_SDIOEN ((uint16_t)0x0800)
```

SD I/O enable functions

5.173.2.2187 SDIO_DLEN_DATALENGTH

```
#define SDIO_DLEN_DATALENGTH ((uint32_t)0x01FFFFFF)
```

Data length value

5.173.2.2188 SDIO_DTIMER_DATATIME

```
#define SDIO_DTIMER_DATATIME ((uint32_t)0xFFFFFFFF)
```

Data timeout period.

5.173.2.2189 SDIO_FIFO_FIFODATA

```
#define SDIO_FIFO_FIFODATA ((uint32_t)0xFFFFFFFF)
```

Receive and transmit FIFO data

5.173.2.2190 SDIO_FIFOCNT_FIFOCOUNT

```
#define SDIO_FIFOCNT_FIFOCOUNT ((uint32_t)0x00FFFFFF)
```

Remaining number of words to be written to or read from the FIFO

5.173.2.2191 SDIO_ICR_CCRCFAILC

```
#define SDIO_ICR_CCRCFAILC ((uint32_t)0x00000001)
```

CCRCFAIL flag clear bit

5.173.2.2192 SDIO_ICR_CEATAENDC

```
#define SDIO_ICR_CEATAENDC ((uint32_t)0x00800000)
```

CEATAEND flag clear bit

5.173.2.2193 SDIO_ICR_CMDRENDC

```
#define SDIO_ICR_CMDRENDC ((uint32_t)0x00000040)
```

CMDREN flag clear bit

5.173.2.2194 SDIO_ICR_CMDSENTC

```
#define SDIO_ICR_CMDSENTC ((uint32_t)0x00000080)
```

CMDSENT flag clear bit

5.173.2.2195 SDIO_ICR_CTIMEOUTC

```
#define SDIO_ICR_CTIMEOUTC ((uint32_t)0x00000004)
```

CTIMEOUT flag clear bit

5.173.2.2196 SDIO_ICR_DATAENDC

```
#define SDIO_ICR_DATAENDC ((uint32_t)0x00000100)
```

DATAEND flag clear bit

5.173.2.2197 SDIO_ICR_DBCKENDC

```
#define SDIO_ICR_DBCKENDC ((uint32_t)0x00000400)
```

DBCKEND flag clear bit

5.173.2.2198 SDIO_ICR_DCRCFAILC

```
#define SDIO_ICR_DCRCFAILC ((uint32_t)0x00000002)
```

DCRCFAIL flag clear bit

5.173.2.2199 SDIO_ICR_DTIMEOUTC

```
#define SDIO_ICR_DTIMEOUTC ((uint32_t)0x00000008)
```

DTIMEOUT flag clear bit

5.173.2.2200 SDIO_ICR_RXOVERRC

```
#define SDIO_ICR_RXOVERRC ((uint32_t)0x00000020)
```

RXOVERR flag clear bit

5.173.2.2201 SDIO_ICR_SDIOITC

```
#define SDIO_ICR_SDIOITC ((uint32_t)0x00400000)
```

SDIOIT flag clear bit

5.173.2.2202 SDIO_ICR_STBITERRC

```
#define SDIO_ICR_STBITERRC ((uint32_t)0x000000200)
```

STBITERR flag clear bit

5.173.2.2203 SDIO_ICR_TXUNDERRC

```
#define SDIO_ICR_TXUNDERRC ((uint32_t)0x000000010)
```

TXUNDERR flag clear bit

5.173.2.2204 SDIO_MASK_CCRCFAILIE

```
#define SDIO_MASK_CCRCFAILIE ((uint32_t)0x00000001)
```

Command CRC Fail Interrupt Enable

5.173.2.2205 SDIO_MASK_CEATAENDIE

```
#define SDIO_MASK_CEATAENDIE ((uint32_t)0x00800000)
```

CE-ATA command completion signal received Interrupt Enable

5.173.2.2206 SDIO_MASK_CMDACTIE

```
#define SDIO_MASK_CMDACTIE ((uint32_t)0x000000800)
```

CCommand Acting Interrupt Enable

5.173.2.2207 SDIO_MASK_CMDRENDIE

```
#define SDIO_MASK_CMDRENDIE ((uint32_t)0x00000040)
```

Command Response Received Interrupt Enable

5.173.2.2208 SDIO_MASK_CMDSENTIE

```
#define SDIO_MASK_CMDSENTIE ((uint32_t)0x00000080)
```

Command Sent Interrupt Enable

5.173.2.2209 SDIO_MASK_CTIMEOUTIE

```
#define SDIO_MASK_CTIMEOUTIE ((uint32_t)0x00000004)
```

Command TimeOut Interrupt Enable

5.173.2.2210 SDIO_MASK_DATAENDIE

```
#define SDIO_MASK_DATAENDIE ((uint32_t)0x00000100)
```

Data End Interrupt Enable

5.173.2.2211 SDIO_MASK_DBCKENDIE

```
#define SDIO_MASK_DBCKENDIE ((uint32_t)0x00000400)
```

Data Block End Interrupt Enable

5.173.2.2212 SDIO_MASK_DCRCFAILIE

```
#define SDIO_MASK_DCRCFAILIE ((uint32_t)0x00000002)
```

Data CRC Fail Interrupt Enable

5.173.2.2213 SDIO_MASK_DTIMEOUTIE

```
#define SDIO_MASK_DTIMEOUTIE ((uint32_t)0x00000008)
```

Data TimeOut Interrupt Enable

5.173.2.2214 SDIO_MASK_RXACTIE

```
#define SDIO_MASK_RXACTIE ((uint32_t)0x00002000)
```

Data receive acting interrupt enabled

5.173.2.2215 SDIO_MASK_RXDAVLIE

```
#define SDIO_MASK_RXDAVLIE ((uint32_t)0x00200000)
```

Data available in Rx FIFO interrupt Enable

5.173.2.2216 SDIO_MASK_RXFIFOEIE

```
#define SDIO_MASK_RXFIFOEIE ((uint32_t)0x00080000)
```

Rx FIFO Empty interrupt Enable

5.173.2.2217 SDIO_MASK_RXFIFOFIE

```
#define SDIO_MASK_RXFIFOFIE ((uint32_t)0x00020000)
```

Rx FIFO Full interrupt Enable

5.173.2.2218 SDIO_MASK_RXFIFOHFIE

```
#define SDIO_MASK_RXFIFOHFIE ((uint32_t)0x00008000)
```

Rx FIFO Half Full interrupt Enable

5.173.2.2219 SDIO_MASK_RXOVERRIE

```
#define SDIO_MASK_RXOVERRIE ((uint32_t)0x00000020)
```

Rx FIFO OverRun Error Interrupt Enable

5.173.2.2220 SDIO_MASK_SDIOITIE

```
#define SDIO_MASK_SDIOITIE ((uint32_t)0x00400000)
```

SDIO Mode Interrupt Received interrupt Enable

5.173.2.2221 SDIO_MASK_STBITERRIE

```
#define SDIO_MASK_STBITERRIE ((uint32_t)0x00000200)
```

Start Bit Error Interrupt Enable

5.173.2.2222 SDIO_MASK_TXACTIE

```
#define SDIO_MASK_TXACTIE ((uint32_t)0x00001000)
```

Data Transmit Acting Interrupt Enable

5.173.2.2223 SDIO_MASK_TXDAVLIE

```
#define SDIO_MASK_TXDAVLIE ((uint32_t)0x00100000)
```

Data available in Tx FIFO interrupt Enable

5.173.2.2224 SDIO_MASK_TXFIFOEIE

```
#define SDIO_MASK_TXFIFOEIE ((uint32_t)0x00040000)
```

Tx FIFO Empty interrupt Enable

5.173.2.2225 SDIO_MASK_TXFIFOFIE

```
#define SDIO_MASK_TXFIFOFIE ((uint32_t)0x00010000)
```

Tx FIFO Full interrupt Enable

5.173.2.2226 SDIO_MASK_TXFIFOHEIE

```
#define SDIO_MASK_TXFIFOHEIE ((uint32_t)0x00004000)
```

Tx FIFO Half Empty interrupt Enable

5.173.2.2227 SDIO_MASK_TXUNDERRIE

```
#define SDIO_MASK_TXUNDERRIE ((uint32_t)0x00000010)
```

Tx FIFO UnderRun Error Interrupt Enable

5.173.2.2228 SDIO_POWER_PWRCTRL

```
#define SDIO_POWER_PWRCTRL ((uint8_t)0x03)
```

PWRCTRL[1:0] bits (Power supply control bits)

5.173.2.2229 SDIO_POWER_PWRCTRL_0

```
#define SDIO_POWER_PWRCTRL_0 ((uint8_t)0x01)
```

Bit 0

5.173.2.2230 SDIO_POWER_PWRCTRL_1

```
#define SDIO_POWER_PWRCTRL_1 ((uint8_t)0x02)
```

Bit 1

5.173.2.2231 SDIO_RESP0_CARDSTATUS0

```
#define SDIO_RESP0_CARDSTATUS0 ((uint32_t)0xFFFFFFFF)
```

Card Status

5.173.2.2232 SDIO_RESP1_CARDSTATUS1

```
#define SDIO_RESP1_CARDSTATUS1 ((uint32_t)0xFFFFFFFF)
```

Card Status

5.173.2.2233 SDIO_RESP2_CARDSTATUS2

```
#define SDIO_RESP2_CARDSTATUS2 ((uint32_t)0xFFFFFFFF)
```

Card Status

5.173.2.2234 SDIO_RESP3_CARDSTATUS3

```
#define SDIO_RESP3_CARDSTATUS3 ((uint32_t)0xFFFFFFFF)
```

Card Status

5.173.2.2235 SDIO_RESP4_CARDSTATUS4

```
#define SDIO_RESP4_CARDSTATUS4 ((uint32_t)0xFFFFFFFF)
```

Card Status

5.173.2.2236 SDIO_RESPCMD_RESPCMD

```
#define SDIO_RESPCMD_RESPCMD ((uint8_t)0x3F)
```

Response command index

5.173.2.2237 SDIO_STA_CCRCFAIL

```
#define SDIO_STA_CCRCFAIL ((uint32_t)0x00000001)
```

Command response received (CRC check failed)

5.173.2.2238 SDIO_STA_CEATAEND

```
#define SDIO_STA_CEATAEND ((uint32_t)0x00800000)
```

CE-ATA command completion signal received for CMD61

5.173.2.2239 SDIO_STA_CMDACT

```
#define SDIO_STA_CMDACT ((uint32_t)0x00000800)
```

Command transfer in progress

5.173.2.2240 SDIO_STA_CMDREND

```
#define SDIO_STA_CMDREND ((uint32_t)0x00000040)
```

Command response received (CRC check passed)

5.173.2.2241 SDIO_STA_CMDSENT

```
#define SDIO_STA_CMDSENT ((uint32_t)0x00000080)
```

Command sent (no response required)

5.173.2.2242 SDIO_STA_CTIMEOUT

```
#define SDIO_STA_CTIMEOUT ((uint32_t)0x00000004)
```

Command response timeout

5.173.2.2243 SDIO_STA_DATAEND

```
#define SDIO_STA_DATAEND ((uint32_t)0x000000100)
```

Data end (data counter, SDIDCOUNT, is zero)

5.173.2.2244 SDIO_STA_DBCKEND

```
#define SDIO_STA_DBCKEND ((uint32_t)0x000000400)
```

Data block sent/received (CRC check passed)

5.173.2.2245 SDIO_STA_DCRCFAIL

```
#define SDIO_STA_DCRCFAIL ((uint32_t)0x000000002)
```

Data block sent/received (CRC check failed)

5.173.2.2246 SDIO_STA_DTIMEOUT

```
#define SDIO_STA_DTIMEOUT ((uint32_t)0x000000008)
```

Data timeout

5.173.2.2247 SDIO_STA_RXACT

```
#define SDIO_STA_RXACT ((uint32_t)0x00002000)
```

Data receive in progress

5.173.2.2248 SDIO_STA_RXDAVL

```
#define SDIO_STA_RXDAVL ((uint32_t)0x00200000)
```

Data available in receive FIFO

5.173.2.2249 SDIO_STA_RXFIFOE

```
#define SDIO_STA_RXFIFOE ((uint32_t)0x00080000)
```

Receive FIFO empty

5.173.2.2250 SDIO_STA_RXFIFOF

```
#define SDIO_STA_RXFIFOF ((uint32_t)0x00020000)
```

Receive FIFO full

5.173.2.2251 SDIO_STA_RXFIFOHF

```
#define SDIO_STA_RXFIFOHF ((uint32_t)0x00008000)
```

Receive FIFO Half Full: there are at least 8 words in the FIFO

5.173.2.2252 SDIO_STA_RXOVERR

```
#define SDIO_STA_RXOVERR ((uint32_t)0x00000020)
```

Received FIFO overrun error

5.173.2.2253 SDIO_STA_SDIOIT

```
#define SDIO_STA_SDIOIT ((uint32_t)0x00400000)
```

SDIO interrupt received

5.173.2.2254 SDIO_STA_STBITERR

```
#define SDIO_STA_STBITERR ((uint32_t)0x00000200)
```

Start bit not detected on all data signals in wide bus mode

5.173.2.2255 SDIO_STA_TXACT

```
#define SDIO_STA_TXACT ((uint32_t)0x00001000)
```

Data transmit in progress

5.173.2.2256 SDIO_STA_TXDAVL

```
#define SDIO_STA_TXDAVL ((uint32_t)0x00100000)
```

Data available in transmit FIFO

5.173.2.2257 SDIO_STA_TXFIFOE

```
#define SDIO_STA_TXFIFOE ((uint32_t)0x00040000)
```

Transmit FIFO empty

5.173.2.2258 SDIO_STA_TXFIFOF

```
#define SDIO_STA_TXFIFOF ((uint32_t)0x00010000)
```

Transmit FIFO full

5.173.2.2259 SDIO_STA_TXFIFOHE

```
#define SDIO_STA_TXFIFOHE ((uint32_t)0x00004000)
```

Transmit FIFO Half Empty: at least 8 words can be written into the FIFO

5.173.2.2260 SDIO_STA_TXUNDERR

```
#define SDIO_STA_TXUNDERR ((uint32_t)0x00000010)
```

Transmit FIFO underrun error

5.173.2.2261 SPI_CR1_BIDIMODE

```
#define SPI_CR1_BIDIMODE ((uint16_t)0x8000)
```

Bidirectional data mode enable

5.173.2.2262 SPI_CR1_BIDIOE

```
#define SPI_CR1_BIDIOE ((uint16_t)0x4000)
```

Output enable in bidirectional mode

5.173.2.2263 SPI_CR1_BR

```
#define SPI_CR1_BR ((uint16_t)0x0038)
```

BR[2:0] bits (Baud Rate Control)

5.173.2.2264 SPI_CR1_BR_0

```
#define SPI_CR1_BR_0 ((uint16_t)0x0008)
```

Bit 0

5.173.2.2265 SPI_CR1_BR_1

```
#define SPI_CR1_BR_1 ((uint16_t)0x0010)
```

Bit 1

5.173.2.2266 SPI_CR1_BR_2

```
#define SPI_CR1_BR_2 ((uint16_t)0x0020)
```

Bit 2

5.173.2.2267 SPI_CR1_CPHA

```
#define SPI_CR1_CPHA ((uint16_t)0x0001)
```

Clock Phase

5.173.2.2268 SPI_CR1_CPOL

```
#define SPI_CR1_CPOL ((uint16_t)0x0002)
```

Clock Polarity

5.173.2.2269 SPI_CR1_CRCEN

```
#define SPI_CR1_CRCEN ((uint16_t)0x2000)
```

Hardware CRC calculation enable

5.173.2.2270 SPI_CR1_CRCNEXT

```
#define SPI_CR1_CRCNEXT ((uint16_t)0x1000)
```

Transmit CRC next

5.173.2.2271 SPI_CR1_DFF

```
#define SPI_CR1_DFF ((uint16_t)0x0800)
```

Data Frame Format

5.173.2.2272 SPI_CR1_LSBFIRST

```
#define SPI_CR1_LSBFIRST ((uint16_t)0x0080)
```

Frame Format

5.173.2.2273 SPI_CR1_MSTR

```
#define SPI_CR1_MSTR ((uint16_t)0x0004)
```

Master Selection

5.173.2.2274 SPI_CR1_RXONLY

```
#define SPI_CR1_RXONLY ((uint16_t)0x0400)
```

Receive only

5.173.2.2275 SPI_CR1_SPE

```
#define SPI_CR1_SPE ((uint16_t)0x0040)
```

SPI Enable

5.173.2.2276 SPI_CR1_SSI

```
#define SPI_CR1_SSI ((uint16_t)0x0100)
```

Internal slave select

5.173.2.2277 SPI_CR1_SSM

```
#define SPI_CR1_SSM ((uint16_t)0x0200)
```

Software slave management

5.173.2.2278 SPI_CR2_ERRIE

```
#define SPI_CR2_ERRIE ((uint8_t)0x20)
```

Error Interrupt Enable

5.173.2.2279 SPI_CR2_RXDMAEN

```
#define SPI_CR2_RXDMAEN ((uint8_t)0x01)
```

Rx Buffer DMA Enable

5.173.2.2280 SPI_CR2_RXNEIE

```
#define SPI_CR2_RXNEIE ((uint8_t)0x40)
```

RX buffer Not Empty Interrupt Enable

5.173.2.2281 SPI_CR2_SSOE

```
#define SPI_CR2_SSOE ((uint8_t)0x04)
```

SS Output Enable

5.173.2.2282 SPI_CR2_TXDMAEN

```
#define SPI_CR2_TXDMAEN ((uint8_t)0x02)
```

Tx Buffer DMA Enable

5.173.2.2283 SPI_CR2_TXEIE

```
#define SPI_CR2_TXEIE ((uint8_t)0x80)
```

Tx buffer Empty Interrupt Enable

5.173.2.2284 SPI_CRCPR_CRCPOLY

```
#define SPI_CRCPR_CRCPOLY ((uint16_t)0xFFFF)
```

CRC polynomial register

5.173.2.2285 SPI_DR_DR

```
#define SPI_DR_DR ((uint16_t)0xFFFF)
```

Data Register

5.173.2.2286 SPI_I2SCFGR_CHLEN

```
#define SPI_I2SCFGR_CHLEN ((uint16_t)0x0001)
```

Channel length (number of bits per audio channel)

5.173.2.2287 SPI_I2SCFGR_CKPOL

```
#define SPI_I2SCFGR_CKPOL ((uint16_t)0x0008)
```

steady state clock polarity

5.173.2.2288 SPI_I2SCFGR_DATLEN

```
#define SPI_I2SCFGR_DATLEN ((uint16_t)0x0006)
```

DATLEN[1:0] bits (Data length to be transferred)

5.173.2.2289 SPI_I2SCFGR_DATLEN_0

```
#define SPI_I2SCFGR_DATLEN_0 ((uint16_t)0x0002)
```

Bit 0

5.173.2.2290 SPI_I2SCFGR_DATLEN_1

```
#define SPI_I2SCFGR_DATLEN_1 ((uint16_t)0x0004)
```

Bit 1

5.173.2.2291 SPI_I2SCFGR_I2SCFG

```
#define SPI_I2SCFGR_I2SCFG ((uint16_t)0x0300)
```

I2SCFG[1:0] bits (I2S configuration mode)

5.173.2.2292 SPI_I2SCFGR_I2SCFG_0

```
#define SPI_I2SCFGR_I2SCFG_0 ((uint16_t)0x0100)
```

Bit 0

5.173.2.2293 SPI_I2SCFGR_I2SCFG_1

```
#define SPI_I2SCFGR_I2SCFG_1 ((uint16_t)0x0200)
```

Bit 1

5.173.2.2294 SPI_I2SCFGR_I2SE

```
#define SPI_I2SCFGR_I2SE ((uint16_t)0x0400)
```

I2S Enable

5.173.2.2295 SPI_I2SCFGR_I2SMOD

```
#define SPI_I2SCFGR_I2SMOD ((uint16_t)0x0800)
```

I2S mode selection

5.173.2.2296 SPI_I2SCFGR_I2SSSTD

```
#define SPI_I2SCFGR_I2SSSTD ((uint16_t)0x0030)
```

I2SSSTD[1:0] bits (I2S standard selection)

5.173.2.2297 SPI_I2SCFGR_I2SSSTD_0

```
#define SPI_I2SCFGR_I2SSSTD_0 ((uint16_t)0x0010)
```

Bit 0

5.173.2.2298 SPI_I2SCFGR_I2SSSTD_1

```
#define SPI_I2SCFGR_I2SSSTD_1 ((uint16_t)0x0020)
```

Bit 1

5.173.2.2299 SPI_I2SCFGR_PCMSYNC

```
#define SPI_I2SCFGR_PCMSYNC ((uint16_t)0x0080)
```

PCM frame synchronization

5.173.2.2300 SPI_I2SPR_I2SDIV

```
#define SPI_I2SPR_I2SDIV ((uint16_t)0x00FF)
```

I2S Linear prescaler

5.173.2.2301 SPI_I2SPR_MCKOE

```
#define SPI_I2SPR_MCKOE ((uint16_t)0x0200)
```

Master Clock Output Enable

5.173.2.2302 SPI_I2SPR_ODD

```
#define SPI_I2SPR_ODD ((uint16_t)0x0100)
```

Odd factor for the prescaler

5.173.2.2303 SPI_RXCRCR_RXCRC

```
#define SPI_RXCRCR_RXCRC ((uint16_t)0xFFFF)
```

Rx CRC Register

5.173.2.2304 SPI_SR_BSY

```
#define SPI_SR_BSY ((uint8_t)0x80)
```

Busy flag

5.173.2.2305 SPI_SR_CHSIDE

```
#define SPI_SR_CHSIDE ((uint8_t)0x04)
```

Channel side

5.173.2.2306 SPI_SR_CRCERR

```
#define SPI_SR_CRCERR ((uint8_t)0x10)
```

CRC Error flag

5.173.2.2307 SPI_SR_MODF

```
#define SPI_SR_MODF ((uint8_t)0x20)
```

Mode fault

5.173.2.2308 SPI_SR_OVR

```
#define SPI_SR_OVR ((uint8_t)0x40)
```

Overrun flag

5.173.2.2309 SPI_SR_RXNE

```
#define SPI_SR_RXNE ((uint8_t)0x01)
```

Receive buffer Not Empty

5.173.2.2310 SPI_SR_TXE

```
#define SPI_SR_TXE ((uint8_t)0x02)
```

Transmit buffer Empty

5.173.2.2311 SPI_SR_UDR

```
#define SPI_SR_UDR ((uint8_t)0x08)
```

Underrun flag

5.173.2.2312 SPI_TXCRCR_TXCRC

```
#define SPI_TXCRCR_TXCRC ((uint16_t)0xFFFF)
```

Tx CRC Register

5.173.2.2313 SYSCFG_CMPCR_CMP_PD

```
#define SYSCFG_CMPCR_CMP_PD ((uint32_t)0x00000001)
```

Compensation cell ready flag

5.173.2.2314 SYSCFG_CMPCR_READY

```
#define SYSCFG_CMPCR_READY ((uint32_t)0x00000100)
```

Compensation cell power-down

5.173.2.2315 SYSCFG_EXTICR1_EXTI0

```
#define SYSCFG_EXTICR1_EXTI0 ((uint16_t)0x000F)
```

EXTI 0 configuration

5.173.2.2316 SYSCFG_EXTICR1_EXTI0_PA

```
#define SYSCFG_EXTICR1_EXTI0_PA ((uint16_t)0x0000)
```

EXTI0 configuration

PA[0] pin

5.173.2.2317 SYSCFG_EXTICR1_EXTI0_PB

```
#define SYSCFG_EXTICR1_EXTI0_PB ((uint16_t)0x0001)
```

PB[0] pin

5.173.2.2318 SYSCFG_EXTICR1_EXTI0_PC

```
#define SYSCFG_EXTICR1_EXTI0_PC ((uint16_t)0x0002)
```

PC[0] pin

5.173.2.2319 SYSCFG_EXTICR1_EXTI0_PD

```
#define SYSCFG_EXTICR1_EXTI0_PD ((uint16_t)0x0003)
```

PD[0] pin

5.173.2.2320 SYSCFG_EXTICR1_EXTI0_PE

```
#define SYSCFG_EXTICR1_EXTI0_PE ((uint16_t)0x0004)

PE[0] pin
```

5.173.2.2321 SYSCFG_EXTICR1_EXTI0_PF

```
#define SYSCFG_EXTICR1_EXTI0_PF ((uint16_t)0x0005)

PF[0] pin
```

5.173.2.2322 SYSCFG_EXTICR1_EXTI0_PG

```
#define SYSCFG_EXTICR1_EXTI0_PG ((uint16_t)0x0006)

PG[0] pin
```

5.173.2.2323 SYSCFG_EXTICR1_EXTI0_PH

```
#define SYSCFG_EXTICR1_EXTI0_PH ((uint16_t)0x0007)

PH[0] pin
```

5.173.2.2324 SYSCFG_EXTICR1_EXTI0_PI

```
#define SYSCFG_EXTICR1_EXTI0_PI ((uint16_t)0x0008)

PI[0] pin
```

5.173.2.2325 SYSCFG_EXTICR1_EXTI0_PJ

```
#define SYSCFG_EXTICR1_EXTI0_PJ ((uint16_t)0x0009)

PJ[0] pin
```

5.173.2.2326 SYSCFG_EXTICR1_EXTI0_PK

```
#define SYSCFG_EXTICR1_EXTI0_PK ((uint16_t)0x000A)

PK[0] pin
```

5.173.2.2327 SYSCFG_EXTICR1_EXTI1

```
#define SYSCFG_EXTICR1_EXTI1 ((uint16_t)0x00F0)

EXTI 1 configuration
```

5.173.2.2328 SYSCFG_EXTICR1_EXTI1_PA

```
#define SYSCFG_EXTICR1_EXTI1_PA ((uint16_t)0x0000)
```

EXTI1 configuration

PA[1] pin

5.173.2.2329 SYSCFG_EXTICR1_EXTI1_PB

```
#define SYSCFG_EXTICR1_EXTI1_PB ((uint16_t)0x0010)
```

PB[1] pin

5.173.2.2330 SYSCFG_EXTICR1_EXTI1_PC

```
#define SYSCFG_EXTICR1_EXTI1_PC ((uint16_t)0x0020)
```

PC[1] pin

5.173.2.2331 SYSCFG_EXTICR1_EXTI1_PD

```
#define SYSCFG_EXTICR1_EXTI1_PD ((uint16_t)0x0030)
```

PD[1] pin

5.173.2.2332 SYSCFG_EXTICR1_EXTI1_PE

```
#define SYSCFG_EXTICR1_EXTI1_PE ((uint16_t)0x0040)
```

PE[1] pin

5.173.2.2333 SYSCFG_EXTICR1_EXTI1_PF

```
#define SYSCFG_EXTICR1_EXTI1_PF ((uint16_t)0x0050)
```

PF[1] pin

5.173.2.2334 SYSCFG_EXTICR1_EXTI1_PG

```
#define SYSCFG_EXTICR1_EXTI1_PG ((uint16_t)0x0060)
```

PG[1] pin

5.173.2.2335 SYSCFG_EXTICR1_EXTI1_PH

```
#define SYSCFG_EXTICR1_EXTI1_PH ((uint16_t)0x0070)
```

PH[1] pin

5.173.2.2336 SYSCFG_EXTICR1_EXTI1_PI

```
#define SYSCFG_EXTICR1_EXTI1_PI ((uint16_t)0x0080)
```

PI[1] pin

5.173.2.2337 SYSCFG_EXTICR1_EXTI1_PJ

```
#define SYSCFG_EXTICR1_EXTI1_PJ ((uint16_t)0x0090)
```

PJ[1] pin

5.173.2.2338 SYSCFG_EXTICR1_EXTI1_PK

```
#define SYSCFG_EXTICR1_EXTI1_PK ((uint16_t)0x00A0)
```

PK[1] pin

5.173.2.2339 SYSCFG_EXTICR1_EXTI2

```
#define SYSCFG_EXTICR1_EXTI2 ((uint16_t)0x0F00)
```

EXTI 2 configuration

5.173.2.2340 SYSCFG_EXTICR1_EXTI2_PA

```
#define SYSCFG_EXTICR1_EXTI2_PA ((uint16_t)0x0000)
```

EXTI2 configuration

PA[2] pin

5.173.2.2341 SYSCFG_EXTICR1_EXTI2_PB

```
#define SYSCFG_EXTICR1_EXTI2_PB ((uint16_t)0x0100)
```

PB[2] pin

5.173.2.2342 SYSCFG_EXTICR1_EXTI2_PC

```
#define SYSCFG_EXTICR1_EXTI2_PC ((uint16_t)0x0200)
```

PC[2] pin

5.173.2.2343 SYSCFG_EXTICR1_EXTI2_PD

```
#define SYSCFG_EXTICR1_EXTI2_PD ((uint16_t)0x0300)
```

PD[2] pin

5.173.2.2344 SYSCFG_EXTICR1_EXTI2_PE

```
#define SYSCFG_EXTICR1_EXTI2_PE ((uint16_t)0x0400)
```

PE[2] pin

5.173.2.2345 SYSCFG_EXTICR1_EXTI2_PF

```
#define SYSCFG_EXTICR1_EXTI2_PF ((uint16_t)0x0500)
```

PF[2] pin

5.173.2.2346 SYSCFG_EXTICR1_EXTI2_PG

```
#define SYSCFG_EXTICR1_EXTI2_PG ((uint16_t)0x0600)
```

PG[2] pin

5.173.2.2347 SYSCFG_EXTICR1_EXTI2_PH

```
#define SYSCFG_EXTICR1_EXTI2_PH ((uint16_t)0x0700)
```

PH[2] pin

5.173.2.2348 SYSCFG_EXTICR1_EXTI2_PI

```
#define SYSCFG_EXTICR1_EXTI2_PI ((uint16_t)0x0800)
```

PI[2] pin

5.173.2.2349 SYSCFG_EXTICR1_EXTI2_PJ

```
#define SYSCFG_EXTICR1_EXTI2_PJ ((uint16_t)0x0900)
```

PJ[2] pin

5.173.2.2350 SYSCFG_EXTICR1_EXTI2_PK

```
#define SYSCFG_EXTICR1_EXTI2_PK ((uint16_t)0x0A00)
```

PK[2] pin

5.173.2.2351 SYSCFG_EXTICR1_EXTI3

```
#define SYSCFG_EXTICR1_EXTI3 ((uint16_t)0xF000)
```

EXTI 3 configuration

5.173.2.2352 SYSCFG_EXTICR1_EXTI3_PA

```
#define SYSCFG_EXTICR1_EXTI3_PA ((uint16_t)0x0000)
```

EXTI3 configuration

PA[3] pin

5.173.2.2353 SYSCFG_EXTICR1_EXTI3_PB

```
#define SYSCFG_EXTICR1_EXTI3_PB ((uint16_t)0x1000)
```

PB[3] pin

5.173.2.2354 SYSCFG_EXTICR1_EXTI3_PC

```
#define SYSCFG_EXTICR1_EXTI3_PC ((uint16_t)0x2000)
```

PC[3] pin

5.173.2.2355 SYSCFG_EXTICR1_EXTI3_PD

```
#define SYSCFG_EXTICR1_EXTI3_PD ((uint16_t)0x3000)
```

PD[3] pin

5.173.2.2356 SYSCFG_EXTICR1_EXTI3_PE

```
#define SYSCFG_EXTICR1_EXTI3_PE ((uint16_t)0x4000)
```

PE[3] pin

5.173.2.2357 SYSCFG_EXTICR1_EXTI3_PF

```
#define SYSCFG_EXTICR1_EXTI3_PF ((uint16_t)0x5000)
```

PF[3] pin

5.173.2.2358 SYSCFG_EXTICR1_EXTI3_PG

```
#define SYSCFG_EXTICR1_EXTI3_PG ((uint16_t)0x6000)
```

PG[3] pin

5.173.2.2359 SYSCFG_EXTICR1_EXTI3_PH

```
#define SYSCFG_EXTICR1_EXTI3_PH ((uint16_t)0x7000)
```

PH[3] pin

5.173.2.2360 SYSCFG_EXTICR1_EXTI3_PI

```
#define SYSCFG_EXTICR1_EXTI3_PI ((uint16_t)0x8000)
```

PI[3] pin

5.173.2.2361 SYSCFG_EXTICR1_EXTI3_PJ

```
#define SYSCFG_EXTICR1_EXTI3_PJ ((uint16_t)0x9000)
```

PJ[3] pin

5.173.2.2362 SYSCFG_EXTICR1_EXTI3_PK

```
#define SYSCFG_EXTICR1_EXTI3_PK ((uint16_t)0xA000)
```

PK[3] pin

5.173.2.2363 SYSCFG_EXTICR2_EXTI4

```
#define SYSCFG_EXTICR2_EXTI4 ((uint16_t)0x000F)
```

EXTI 4 configuration

5.173.2.2364 SYSCFG_EXTICR2_EXTI4_PA

```
#define SYSCFG_EXTICR2_EXTI4_PA ((uint16_t)0x0000)
```

EXTI4 configuration

PA[4] pin

5.173.2.2365 SYSCFG_EXTICR2_EXTI4_PB

```
#define SYSCFG_EXTICR2_EXTI4_PB ((uint16_t)0x0001)
```

PB[4] pin

5.173.2.2366 SYSCFG_EXTICR2_EXTI4_PC

```
#define SYSCFG_EXTICR2_EXTI4_PC ((uint16_t)0x0002)
```

PC[4] pin

5.173.2.2367 SYSCFG_EXTICR2_EXTI4_PD

```
#define SYSCFG_EXTICR2_EXTI4_PD ((uint16_t)0x0003)
```

PD[4] pin

5.173.2.2368 SYSCFG_EXTICR2_EXTI4_PE

```
#define SYSCFG_EXTICR2_EXTI4_PE ((uint16_t)0x0004)
```

PE[4] pin

5.173.2.2369 SYSCFG_EXTICR2_EXTI4_PF

```
#define SYSCFG_EXTICR2_EXTI4_PF ((uint16_t)0x0005)
```

PF[4] pin

5.173.2.2370 SYSCFG_EXTICR2_EXTI4_PG

```
#define SYSCFG_EXTICR2_EXTI4_PG ((uint16_t)0x0006)
```

PG[4] pin

5.173.2.2371 SYSCFG_EXTICR2_EXTI4_PH

```
#define SYSCFG_EXTICR2_EXTI4_PH ((uint16_t)0x0007)
```

PH[4] pin

5.173.2.2372 SYSCFG_EXTICR2_EXTI4_PI

```
#define SYSCFG_EXTICR2_EXTI4_PI ((uint16_t)0x0008)
```

PI[4] pin

5.173.2.2373 SYSCFG_EXTICR2_EXTI4_PJ

```
#define SYSCFG_EXTICR2_EXTI4_PJ ((uint16_t)0x0009)
```

PJ[4] pin

5.173.2.2374 SYSCFG_EXTICR2_EXTI4_PK

```
#define SYSCFG_EXTICR2_EXTI4_PK ((uint16_t)0x000A)
```

PK[4] pin

5.173.2.2375 SYSCFG_EXTICR2_EXTI5

```
#define SYSCFG_EXTICR2_EXTI5 ((uint16_t)0x00F0)
```

EXTI 5 configuration

5.173.2.2376 SYSCFG_EXTICR2_EXTI5_PA

```
#define SYSCFG_EXTICR2_EXTI5_PA ((uint16_t)0x0000)
```

EXTI5 configuration

PA[5] pin

5.173.2.2377 SYSCFG_EXTICR2_EXTI5_PB

```
#define SYSCFG_EXTICR2_EXTI5_PB ((uint16_t)0x0010)
```

PB[5] pin

5.173.2.2378 SYSCFG_EXTICR2_EXTI5_PC

```
#define SYSCFG_EXTICR2_EXTI5_PC ((uint16_t)0x0020)

PC[5] pin
```

5.173.2.2379 SYSCFG_EXTICR2_EXTI5_PD

```
#define SYSCFG_EXTICR2_EXTI5_PD ((uint16_t)0x0030)

PD[5] pin
```

5.173.2.2380 SYSCFG_EXTICR2_EXTI5_PE

```
#define SYSCFG_EXTICR2_EXTI5_PE ((uint16_t)0x0040)

PE[5] pin
```

5.173.2.2381 SYSCFG_EXTICR2_EXTI5_PF

```
#define SYSCFG_EXTICR2_EXTI5_PF ((uint16_t)0x0050)

PF[5] pin
```

5.173.2.2382 SYSCFG_EXTICR2_EXTI5_PG

```
#define SYSCFG_EXTICR2_EXTI5_PG ((uint16_t)0x0060)

PG[5] pin
```

5.173.2.2383 SYSCFG_EXTICR2_EXTI5_PH

```
#define SYSCFG_EXTICR2_EXTI5_PH ((uint16_t)0x0070)

PH[5] pin
```

5.173.2.2384 SYSCFG_EXTICR2_EXTI5_PI

```
#define SYSCFG_EXTICR2_EXTI5_PI ((uint16_t)0x0080)

PI[5] pin
```

5.173.2.2385 SYSCFG_EXTICR2_EXTI5_PJ

```
#define SYSCFG_EXTICR2_EXTI5_PJ ((uint16_t)0x0090)

PJ[5] pin
```

5.173.2.2386 SYSCFG_EXTICR2_EXTI5_PK

```
#define SYSCFG_EXTICR2_EXTI5_PK ((uint16_t)0x00A0)
```

PK[5] pin

5.173.2.2387 SYSCFG_EXTICR2_EXTI6

```
#define SYSCFG_EXTICR2_EXTI6 ((uint16_t)0x0F00)
```

EXTI 6 configuration

5.173.2.2388 SYSCFG_EXTICR2_EXTI6_PA

```
#define SYSCFG_EXTICR2_EXTI6_PA ((uint16_t)0x0000)
```

EXTI6 configuration

PA[6] pin

5.173.2.2389 SYSCFG_EXTICR2_EXTI6_PB

```
#define SYSCFG_EXTICR2_EXTI6_PB ((uint16_t)0x0100)
```

PB[6] pin

5.173.2.2390 SYSCFG_EXTICR2_EXTI6_PC

```
#define SYSCFG_EXTICR2_EXTI6_PC ((uint16_t)0x0200)
```

PC[6] pin

5.173.2.2391 SYSCFG_EXTICR2_EXTI6_PD

```
#define SYSCFG_EXTICR2_EXTI6_PD ((uint16_t)0x0300)
```

PD[6] pin

5.173.2.2392 SYSCFG_EXTICR2_EXTI6_PE

```
#define SYSCFG_EXTICR2_EXTI6_PE ((uint16_t)0x0400)
```

PE[6] pin

5.173.2.2393 SYSCFG_EXTICR2_EXTI6_PF

```
#define SYSCFG_EXTICR2_EXTI6_PF ((uint16_t)0x0500)
```

PF[6] pin

5.173.2.2394 SYSCFG_EXTICR2_EXTI6_PG

```
#define SYSCFG_EXTICR2_EXTI6_PG ((uint16_t)0x0600)
```

PG[6] pin

5.173.2.2395 SYSCFG_EXTICR2_EXTI6_PH

```
#define SYSCFG_EXTICR2_EXTI6_PH ((uint16_t)0x0700)
```

PH[6] pin

5.173.2.2396 SYSCFG_EXTICR2_EXTI6_PI

```
#define SYSCFG_EXTICR2_EXTI6_PI ((uint16_t)0x0800)
```

PI[6] pin

5.173.2.2397 SYSCFG_EXTICR2_EXTI6_PJ

```
#define SYSCFG_EXTICR2_EXTI6_PJ ((uint16_t)0x0900)
```

PJ[6] pin

5.173.2.2398 SYSCFG_EXTICR2_EXTI6_PK

```
#define SYSCFG_EXTICR2_EXTI6_PK ((uint16_t)0x0A00)
```

PK[6] pin

5.173.2.2399 SYSCFG_EXTICR2_EXTI7

```
#define SYSCFG_EXTICR2_EXTI7 ((uint16_t)0xF000)
```

EXTI 7 configuration

5.173.2.2400 SYSCFG_EXTICR2_EXTI7_PA

```
#define SYSCFG_EXTICR2_EXTI7_PA ((uint16_t)0x0000)
```

EXTI7 configuration

PA[7] pin

5.173.2.2401 SYSCFG_EXTICR2_EXTI7_PB

```
#define SYSCFG_EXTICR2_EXTI7_PB ((uint16_t)0x1000)
```

PB[7] pin

5.173.2.2402 SYSCFG_EXTICR2_EXTI7_PC

```
#define SYSCFG_EXTICR2_EXTI7_PC ((uint16_t)0x2000)
```

PC[7] pin

5.173.2.2403 SYSCFG_EXTICR2_EXTI7_PD

```
#define SYSCFG_EXTICR2_EXTI7_PD ((uint16_t)0x3000)
```

PD[7] pin

5.173.2.2404 SYSCFG_EXTICR2_EXTI7_PE

```
#define SYSCFG_EXTICR2_EXTI7_PE ((uint16_t)0x4000)
```

PE[7] pin

5.173.2.2405 SYSCFG_EXTICR2_EXTI7_PF

```
#define SYSCFG_EXTICR2_EXTI7_PF ((uint16_t)0x5000)
```

PF[7] pin

5.173.2.2406 SYSCFG_EXTICR2_EXTI7_PG

```
#define SYSCFG_EXTICR2_EXTI7_PG ((uint16_t)0x6000)
```

PG[7] pin

5.173.2.2407 SYSCFG_EXTICR2_EXTI7_PH

```
#define SYSCFG_EXTICR2_EXTI7_PH ((uint16_t)0x7000)
```

PH[7] pin

5.173.2.2408 SYSCFG_EXTICR2_EXTI7_PI

```
#define SYSCFG_EXTICR2_EXTI7_PI ((uint16_t)0x8000)
```

PI[7] pin

5.173.2.2409 SYSCFG_EXTICR2_EXTI7_PJ

```
#define SYSCFG_EXTICR2_EXTI7_PJ ((uint16_t)0x9000)
```

PJ[7] pin

5.173.2.2410 SYSCFG_EXTICR2_EXTI7_PK

```
#define SYSCFG_EXTICR2_EXTI7_PK ((uint16_t)0xA000)
```

PK[7] pin

5.173.2.2411 SYSCFG_EXTICR3_EXTI10

```
#define SYSCFG_EXTICR3_EXTI10 ((uint16_t)0x0F00)
```

EXTI 10 configuration

5.173.2.2412 SYSCFG_EXTICR3_EXTI10_PA

```
#define SYSCFG_EXTICR3_EXTI10_PA ((uint16_t)0x0000)
```

EXTI10 configuration

PA[10] pin

5.173.2.2413 SYSCFG_EXTICR3_EXTI10_PB

```
#define SYSCFG_EXTICR3_EXTI10_PB ((uint16_t)0x0100)
```

PB[10] pin

5.173.2.2414 SYSCFG_EXTICR3_EXTI10_PC

```
#define SYSCFG_EXTICR3_EXTI10_PC ((uint16_t)0x0200)  
PC[10] pin
```

5.173.2.2415 SYSCFG_EXTICR3_EXTI10_PD

```
#define SYSCFG_EXTICR3_EXTI10_PD ((uint16_t)0x0300)  
PD[10] pin
```

5.173.2.2416 SYSCFG_EXTICR3_EXTI10_PE

```
#define SYSCFG_EXTICR3_EXTI10_PE ((uint16_t)0x0400)  
PE[10] pin
```

5.173.2.2417 SYSCFG_EXTICR3_EXTI10_PF

```
#define SYSCFG_EXTICR3_EXTI10_PF ((uint16_t)0x0500)  
PF[10] pin
```

5.173.2.2418 SYSCFG_EXTICR3_EXTI10_PG

```
#define SYSCFG_EXTICR3_EXTI10_PG ((uint16_t)0x0600)  
PG[10] pin
```

5.173.2.2419 SYSCFG_EXTICR3_EXTI10_PH

```
#define SYSCFG_EXTICR3_EXTI10_PH ((uint16_t)0x0700)  
PH[10] pin
```

5.173.2.2420 SYSCFG_EXTICR3_EXTI10_PI

```
#define SYSCFG_EXTICR3_EXTI10_PI ((uint16_t)0x0800)  
PI[10] pin
```

5.173.2.2421 SYSCFG_EXTICR3_EXTI10_PJ

```
#define SYSCFG_EXTICR3_EXTI10_PJ ((uint16_t)0x0900)  
PJ[10] pin
```

5.173.2.2422 SYSCFG_EXTICR3_EXTI11

```
#define SYSCFG_EXTICR3_EXTI11 ((uint16_t)0xF000)
```

EXTI 11 configuration

5.173.2.2423 SYSCFG_EXTICR3_EXTI11_PA

```
#define SYSCFG_EXTICR3_EXTI11_PA ((uint16_t)0x0000)
```

EXTI11 configuration

PA[11] pin

5.173.2.2424 SYSCFG_EXTICR3_EXTI11_PB

```
#define SYSCFG_EXTICR3_EXTI11_PB ((uint16_t)0x1000)
```

PB[11] pin

5.173.2.2425 SYSCFG_EXTICR3_EXTI11_PC

```
#define SYSCFG_EXTICR3_EXTI11_PC ((uint16_t)0x2000)
```

PC[11] pin

5.173.2.2426 SYSCFG_EXTICR3_EXTI11_PD

```
#define SYSCFG_EXTICR3_EXTI11_PD ((uint16_t)0x3000)
```

PD[11] pin

5.173.2.2427 SYSCFG_EXTICR3_EXTI11_PE

```
#define SYSCFG_EXTICR3_EXTI11_PE ((uint16_t)0x4000)
```

PE[11] pin

5.173.2.2428 SYSCFG_EXTICR3_EXTI11_PF

```
#define SYSCFG_EXTICR3_EXTI11_PF ((uint16_t)0x5000)
```

PF[11] pin

5.173.2.2429 SYSCFG_EXTICR3_EXTI11_PG

```
#define SYSCFG_EXTICR3_EXTI11_PG ((uint16_t)0x6000)
```

PG[11] pin

5.173.2.2430 SYSCFG_EXTICR3_EXTI11_PH

```
#define SYSCFG_EXTICR3_EXTI11_PH ((uint16_t)0x7000)
```

PH[11] pin

5.173.2.2431 SYSCFG_EXTICR3_EXTI11_PI

```
#define SYSCFG_EXTICR3_EXTI11_PI ((uint16_t)0x8000)
```

PI[11] pin

5.173.2.2432 SYSCFG_EXTICR3_EXTI11_PJ

```
#define SYSCFG_EXTICR3_EXTI11_PJ ((uint16_t)0x9000)
```

PJ[11] pin

5.173.2.2433 SYSCFG_EXTICR3_EXTI8

```
#define SYSCFG_EXTICR3_EXTI8 ((uint16_t)0x000F)
```

EXTI 8 configuration

5.173.2.2434 SYSCFG_EXTICR3_EXTI8_PA

```
#define SYSCFG_EXTICR3_EXTI8_PA ((uint16_t)0x0000)
```

EXTI8 configuration

PA[8] pin

5.173.2.2435 SYSCFG_EXTICR3_EXTI8_PB

```
#define SYSCFG_EXTICR3_EXTI8_PB ((uint16_t)0x0001)
```

PB[8] pin

5.173.2.2436 SYSCFG_EXTICR3_EXTI8_PC

```
#define SYSCFG_EXTICR3_EXTI8_PC ((uint16_t)0x0002)

PC[8] pin
```

5.173.2.2437 SYSCFG_EXTICR3_EXTI8_PD

```
#define SYSCFG_EXTICR3_EXTI8_PD ((uint16_t)0x0003)

PD[8] pin
```

5.173.2.2438 SYSCFG_EXTICR3_EXTI8_PE

```
#define SYSCFG_EXTICR3_EXTI8_PE ((uint16_t)0x0004)

PE[8] pin
```

5.173.2.2439 SYSCFG_EXTICR3_EXTI8_PF

```
#define SYSCFG_EXTICR3_EXTI8_PF ((uint16_t)0x0005)

PF[8] pin
```

5.173.2.2440 SYSCFG_EXTICR3_EXTI8_PG

```
#define SYSCFG_EXTICR3_EXTI8_PG ((uint16_t)0x0006)

PG[8] pin
```

5.173.2.2441 SYSCFG_EXTICR3_EXTI8_PH

```
#define SYSCFG_EXTICR3_EXTI8_PH ((uint16_t)0x0007)

PH[8] pin
```

5.173.2.2442 SYSCFG_EXTICR3_EXTI8_PI

```
#define SYSCFG_EXTICR3_EXTI8_PI ((uint16_t)0x0008)

PI[8] pin
```

5.173.2.2443 SYSCFG_EXTICR3_EXTI8_PJ

```
#define SYSCFG_EXTICR3_EXTI8_PJ ((uint16_t)0x0009)

PJ[8] pin
```

5.173.2.2444 SYSCFG_EXTICR3_EXTI9

```
#define SYSCFG_EXTICR3_EXTI9 ((uint16_t)0x00F0)
```

EXTI 9 configuration

5.173.2.2445 SYSCFG_EXTICR3_EXTI9_PA

```
#define SYSCFG_EXTICR3_EXTI9_PA ((uint16_t)0x0000)
```

EXTI9 configuration

PA[9] pin

5.173.2.2446 SYSCFG_EXTICR3_EXTI9_PB

```
#define SYSCFG_EXTICR3_EXTI9_PB ((uint16_t)0x0010)
```

PB[9] pin

5.173.2.2447 SYSCFG_EXTICR3_EXTI9_PC

```
#define SYSCFG_EXTICR3_EXTI9_PC ((uint16_t)0x0020)
```

PC[9] pin

5.173.2.2448 SYSCFG_EXTICR3_EXTI9_PD

```
#define SYSCFG_EXTICR3_EXTI9_PD ((uint16_t)0x0030)
```

PD[9] pin

5.173.2.2449 SYSCFG_EXTICR3_EXTI9_PE

```
#define SYSCFG_EXTICR3_EXTI9_PE ((uint16_t)0x0040)
```

PE[9] pin

5.173.2.2450 SYSCFG_EXTICR3_EXTI9_PF

```
#define SYSCFG_EXTICR3_EXTI9_PF ((uint16_t)0x0050)
```

PF[9] pin

5.173.2.2451 SYSCFG_EXTICR3_EXTI9_PG

```
#define SYSCFG_EXTICR3_EXTI9_PG ((uint16_t)0x0060)
```

PG[9] pin

5.173.2.2452 SYSCFG_EXTICR3_EXTI9_PH

```
#define SYSCFG_EXTICR3_EXTI9_PH ((uint16_t)0x0070)
```

PH[9] pin

5.173.2.2453 SYSCFG_EXTICR3_EXTI9_PI

```
#define SYSCFG_EXTICR3_EXTI9_PI ((uint16_t)0x0080)
```

PI[9] pin

5.173.2.2454 SYSCFG_EXTICR3_EXTI9_PJ

```
#define SYSCFG_EXTICR3_EXTI9_PJ ((uint16_t)0x0090)
```

PJ[9] pin

5.173.2.2455 SYSCFG_EXTICR4_EXTI12

```
#define SYSCFG_EXTICR4_EXTI12 ((uint16_t)0x000F)
```

EXTI 12 configuration

5.173.2.2456 SYSCFG_EXTICR4_EXTI12_PA

```
#define SYSCFG_EXTICR4_EXTI12_PA ((uint16_t)0x0000)
```

EXTI12 configuration

PA[12] pin

5.173.2.2457 SYSCFG_EXTICR4_EXTI12_PB

```
#define SYSCFG_EXTICR4_EXTI12_PB ((uint16_t)0x0001)
```

PB[12] pin

5.173.2.2458 SYSCFG_EXTICR4 EXTI12_PC

```
#define SYSCFG_EXTICR4_EXTI12_PC ((uint16_t)0x0002)  
PC[12] pin
```

5.173.2.2459 SYSCFG_EXTICR4 EXTI12_PD

```
#define SYSCFG_EXTICR4_EXTI12_PD ((uint16_t)0x0003)  
PD[12] pin
```

5.173.2.2460 SYSCFG_EXTICR4 EXTI12_PE

```
#define SYSCFG_EXTICR4_EXTI12_PE ((uint16_t)0x0004)  
PE[12] pin
```

5.173.2.2461 SYSCFG_EXTICR4 EXTI12_PF

```
#define SYSCFG_EXTICR4_EXTI12_PF ((uint16_t)0x0005)  
PF[12] pin
```

5.173.2.2462 SYSCFG_EXTICR4 EXTI12_PG

```
#define SYSCFG_EXTICR4_EXTI12_PG ((uint16_t)0x0006)  
PG[12] pin
```

5.173.2.2463 SYSCFG_EXTICR4 EXTI12_PH

```
#define SYSCFG_EXTICR4_EXTI12_PH ((uint16_t)0x0007)  
PH[12] pin
```

5.173.2.2464 SYSCFG_EXTICR4 EXTI12_PI

```
#define SYSCFG_EXTICR4_EXTI12_PI ((uint16_t)0x0008)  
PI[12] pin
```

5.173.2.2465 SYSCFG_EXTICR4 EXTI12_PJ

```
#define SYSCFG_EXTICR4_EXTI12_PJ ((uint16_t)0x0009)  
PJ[12] pin
```

5.173.2.2466 SYSCFG_EXTICR4_EXTI13

```
#define SYSCFG_EXTICR4_EXTI13 ((uint16_t)0x00F0)
```

EXTI 13 configuration

5.173.2.2467 SYSCFG_EXTICR4_EXTI13_PA

```
#define SYSCFG_EXTICR4_EXTI13_PA ((uint16_t)0x0000)
```

EXTI13 configuration

PA[13] pin

5.173.2.2468 SYSCFG_EXTICR4_EXTI13_PB

```
#define SYSCFG_EXTICR4_EXTI13_PB ((uint16_t)0x0010)
```

PB[13] pin

5.173.2.2469 SYSCFG_EXTICR4_EXTI13_PC

```
#define SYSCFG_EXTICR4_EXTI13_PC ((uint16_t)0x0020)
```

PC[13] pin

5.173.2.2470 SYSCFG_EXTICR4_EXTI13_PD

```
#define SYSCFG_EXTICR4_EXTI13_PD ((uint16_t)0x0030)
```

PD[13] pin

5.173.2.2471 SYSCFG_EXTICR4_EXTI13_PE

```
#define SYSCFG_EXTICR4_EXTI13_PE ((uint16_t)0x0040)
```

PE[13] pin

5.173.2.2472 SYSCFG_EXTICR4_EXTI13_PF

```
#define SYSCFG_EXTICR4_EXTI13_PF ((uint16_t)0x0050)
```

PF[13] pin

5.173.2.2473 SYSCFG_EXTICR4 EXTI13 PG

```
#define SYSCFG_EXTICR4_EXTI13_PG ((uint16_t)0x0060)
```

PG[13] pin

5.173.2.2474 SYSCFG_EXTICR4 EXTI13 PH

```
#define SYSCFG_EXTICR4_EXTI13_PH ((uint16_t)0x0070)
```

PH[13] pin

5.173.2.2475 SYSCFG_EXTICR4 EXTI13 PI

```
#define SYSCFG_EXTICR4_EXTI13_PI ((uint16_t)0x0008)
```

PI[13] pin

5.173.2.2476 SYSCFG_EXTICR4 EXTI13 PJ

```
#define SYSCFG_EXTICR4_EXTI13_PJ ((uint16_t)0x0009)
```

PJ[13] pin

5.173.2.2477 SYSCFG_EXTICR4 EXTI14

```
#define SYSCFG_EXTICR4_EXTI14 ((uint16_t)0x0F00)
```

EXTI 14 configuration

5.173.2.2478 SYSCFG_EXTICR4 EXTI14 PA

```
#define SYSCFG_EXTICR4_EXTI14_PA ((uint16_t)0x0000)
```

EXTI14 configuration

PA[14] pin

5.173.2.2479 SYSCFG_EXTICR4 EXTI14 PB

```
#define SYSCFG_EXTICR4_EXTI14_PB ((uint16_t)0x0100)
```

PB[14] pin

5.173.2.2480 SYSCFG_EXTICR4 EXTI14_PC

```
#define SYSCFG_EXTICR4_EXTI14_PC ((uint16_t)0x0200)  
PC[14] pin
```

5.173.2.2481 SYSCFG_EXTICR4 EXTI14_PD

```
#define SYSCFG_EXTICR4_EXTI14_PD ((uint16_t)0x0300)  
PD[14] pin
```

5.173.2.2482 SYSCFG_EXTICR4 EXTI14_PE

```
#define SYSCFG_EXTICR4_EXTI14_PE ((uint16_t)0x0400)  
PE[14] pin
```

5.173.2.2483 SYSCFG_EXTICR4 EXTI14_PF

```
#define SYSCFG_EXTICR4_EXTI14_PF ((uint16_t)0x0500)  
PF[14] pin
```

5.173.2.2484 SYSCFG_EXTICR4 EXTI14_PG

```
#define SYSCFG_EXTICR4_EXTI14_PG ((uint16_t)0x0600)  
PG[14] pin
```

5.173.2.2485 SYSCFG_EXTICR4 EXTI14_PH

```
#define SYSCFG_EXTICR4_EXTI14_PH ((uint16_t)0x0700)  
PH[14] pin
```

5.173.2.2486 SYSCFG_EXTICR4 EXTI14_PI

```
#define SYSCFG_EXTICR4_EXTI14_PI ((uint16_t)0x0800)  
PI[14] pin
```

5.173.2.2487 SYSCFG_EXTICR4 EXTI14_PJ

```
#define SYSCFG_EXTICR4_EXTI14_PJ ((uint16_t)0x0900)  
PJ[14] pin
```

5.173.2.2488 SYSCFG_EXTICR4_EXTI15

```
#define SYSCFG_EXTICR4_EXTI15 ((uint16_t)0xF000)
```

EXTI 15 configuration

5.173.2.2489 SYSCFG_EXTICR4_EXTI15_PA

```
#define SYSCFG_EXTICR4_EXTI15_PA ((uint16_t)0x0000)
```

EXTI15 configuration

PA[15] pin

5.173.2.2490 SYSCFG_EXTICR4_EXTI15_PB

```
#define SYSCFG_EXTICR4_EXTI15_PB ((uint16_t)0x1000)
```

PB[15] pin

5.173.2.2491 SYSCFG_EXTICR4_EXTI15_PC

```
#define SYSCFG_EXTICR4_EXTI15_PC ((uint16_t)0x2000)
```

PC[15] pin

5.173.2.2492 SYSCFG_EXTICR4_EXTI15_PD

```
#define SYSCFG_EXTICR4_EXTI15_PD ((uint16_t)0x3000)
```

PD[15] pin

5.173.2.2493 SYSCFG_EXTICR4_EXTI15_PE

```
#define SYSCFG_EXTICR4_EXTI15_PE ((uint16_t)0x4000)
```

PE[15] pin

5.173.2.2494 SYSCFG_EXTICR4_EXTI15_PF

```
#define SYSCFG_EXTICR4_EXTI15_PF ((uint16_t)0x5000)
```

PF[15] pin

5.173.2.2495 SYSCFG_EXTICR4 EXTI15 PG

```
#define SYSCFG_EXTICR4_EXTI15_PG ((uint16_t)0x6000)  
PG[15] pin
```

5.173.2.2496 SYSCFG_EXTICR4 EXTI15 PH

```
#define SYSCFG_EXTICR4_EXTI15_PH ((uint16_t)0x7000)  
PH[15] pin
```

5.173.2.2497 SYSCFG_EXTICR4 EXTI15 PI

```
#define SYSCFG_EXTICR4_EXTI15_PI ((uint16_t)0x8000)  
PI[15] pin
```

5.173.2.2498 SYSCFG_EXTICR4 EXTI15 PJ

```
#define SYSCFG_EXTICR4_EXTI15_PJ ((uint16_t)0x9000)  
PJ[15] pin
```

5.173.2.2499 SYSCFG_MEMRMP_FB_MODE

```
#define SYSCFG_MEMRMP_FB_MODE ((uint32_t)0x00000100)  
User Flash Bank mode
```

5.173.2.2500 SYSCFG_MEMRMP_MEM_MODE

```
#define SYSCFG_MEMRMP_MEM_MODE ((uint32_t)0x00000007)  
SYSCFG_Memory Remap Config
```

5.173.2.2501 SYSCFG_MEMRMP_MEM_MODE_0

```
#define SYSCFG_MEMRMP_MEM_MODE_0 ((uint32_t)0x00000001)  
Bit 0
```

5.173.2.2502 SYSCFG_MEMRMP_MEM_MODE_1

```
#define SYSCFG_MEMRMP_MEM_MODE_1 ((uint32_t)0x00000002)  
Bit 1
```

5.173.2.2503 SYSCFG_MEMRMP_MEM_MODE_2

```
#define SYSCFG_MEMRMP_MEM_MODE_2 ((uint32_t)0x00000004)
```

Bit 2

5.173.2.2504 SYSCFG_MEMRMP_SWP_FMC

```
#define SYSCFG_MEMRMP_SWP_FMC ((uint32_t)0x00000C00)
```

FMC memory mapping swap

5.173.2.2505 SYSCFG_MEMRMP_SWP_FMC_0

```
#define SYSCFG_MEMRMP_SWP_FMC_0 ((uint32_t)0x00000400)
```

Bit 0

5.173.2.2506 SYSCFG_MEMRMP_SWP_FMC_1

```
#define SYSCFG_MEMRMP_SWP_FMC_1 ((uint32_t)0x00000800)
```

Bit 1

5.173.2.2507 SYSCFG_PMC_ADC1DC2

```
#define SYSCFG_PMC_ADC1DC2 ((uint32_t)0x00010000)
```

Refer to AN4073 on how to use this bit

5.173.2.2508 SYSCFG_PMC_ADC2DC2

```
#define SYSCFG_PMC_ADC2DC2 ((uint32_t)0x00020000)
```

Refer to AN4073 on how to use this bit

5.173.2.2509 SYSCFG_PMC_ADC3DC2

```
#define SYSCFG_PMC_ADC3DC2 ((uint32_t)0x00040000)
```

Refer to AN4073 on how to use this bit

5.173.2.2510 SYSCFG_PMC_ADCxDC2

```
#define SYSCFG_PMC_ADCxDC2 ((uint32_t)0x00070000)
```

Refer to AN4073 on how to use this bit

5.173.2.2511 SYSCFG_PMC_MII_RMII_SEL

```
#define SYSCFG_PMC_MII_RMII_SEL ((uint32_t)0x00800000)
```

Ethernet PHY interface selection

5.173.2.2512 TIM_ARR_ARR

```
#define TIM_ARR_ARR ((uint16_t)0xFFFF)
```

actual auto-reload Value

5.173.2.2513 TIM_BDTR_AOE

```
#define TIM_BDTR_AOE ((uint16_t)0x4000)
```

Automatic Output enable

5.173.2.2514 TIM_BDTR_BKE

```
#define TIM_BDTR_BKE ((uint16_t)0x1000)
```

Break enable

5.173.2.2515 TIM_BDTR_BKP

```
#define TIM_BDTR_BKP ((uint16_t)0x2000)
```

Break Polarity

5.173.2.2516 TIM_BDTR_DTG

```
#define TIM_BDTR_DTG ((uint16_t)0x00FF)
```

DTG[0:7] bits (Dead-Time Generator set-up)

5.173.2.2517 TIM_BDTR_DTG_0

```
#define TIM_BDTR_DTG_0 ((uint16_t)0x0001)
```

Bit 0

5.173.2.2518 TIM_BDTR_DTG_1

```
#define TIM_BDTR_DTG_1 ((uint16_t)0x0002)
```

Bit 1

5.173.2.2519 TIM_BDTR_DTG_2

```
#define TIM_BDTR_DTG_2 ((uint16_t)0x0004)
```

Bit 2

5.173.2.2520 TIM_BDTR_DTG_3

```
#define TIM_BDTR_DTG_3 ((uint16_t)0x0008)
```

Bit 3

5.173.2.2521 TIM_BDTR_DTG_4

```
#define TIM_BDTR_DTG_4 ((uint16_t)0x0010)
```

Bit 4

5.173.2.2522 TIM_BDTR_DTG_5

```
#define TIM_BDTR_DTG_5 ((uint16_t)0x0020)
```

Bit 5

5.173.2.2523 TIM_BDTR_DTG_6

```
#define TIM_BDTR_DTG_6 ((uint16_t)0x0040)
```

Bit 6

5.173.2.2524 TIM_BDTR_DTG_7

```
#define TIM_BDTR_DTG_7 ((uint16_t)0x0080)
```

Bit 7

5.173.2.2525 TIM_BDTR_LOCK

```
#define TIM_BDTR_LOCK ((uint16_t)0x0300)
```

LOCK[1:0] bits (Lock Configuration)

5.173.2.2526 TIM_BDTR_LOCK_0

```
#define TIM_BDTR_LOCK_0 ((uint16_t)0x0100)
```

Bit 0

5.173.2.2527 TIM_BDTR_LOCK_1

```
#define TIM_BDTR_LOCK_1 ((uint16_t)0x0200)
```

Bit 1

5.173.2.2528 TIM_BDTR_MOE

```
#define TIM_BDTR_MOE ((uint16_t)0x8000)
```

Main Output enable

5.173.2.2529 TIM_BDTR_OSSI

```
#define TIM_BDTR_OSSI ((uint16_t)0x0400)
```

Off-State Selection for Idle mode

5.173.2.2530 TIM_BDTR_OSSR

```
#define TIM_BDTR_OSSR ((uint16_t)0x0800)
```

Off-State Selection for Run mode

5.173.2.2531 TIM_CCER_CC1E

```
#define TIM_CCER_CC1E ((uint16_t)0x0001)
```

Capture/Compare 1 output enable

5.173.2.2532 TIM_CCER_CC1NE

```
#define TIM_CCER_CC1NE ((uint16_t)0x0004)
```

Capture/Compare 1 Complementary output enable

5.173.2.2533 TIM_CCER_CC1NP

```
#define TIM_CCER_CC1NP ((uint16_t)0x0008)
```

Capture/Compare 1 Complementary output Polarity

5.173.2.2534 TIM_CCER_CC1P

```
#define TIM_CCER_CC1P ((uint16_t)0x0002)
```

Capture/Compare 1 output Polarity

5.173.2.2535 TIM_CCER_CC2E

```
#define TIM_CCER_CC2E ((uint16_t)0x0010)
```

Capture/Compare 2 output enable

5.173.2.2536 TIM_CCER_CC2NE

```
#define TIM_CCER_CC2NE ((uint16_t)0x0040)
```

Capture/Compare 2 Complementary output enable

5.173.2.2537 TIM_CCER_CC2NP

```
#define TIM_CCER_CC2NP ((uint16_t)0x0080)
```

Capture/Compare 2 Complementary output Polarity

5.173.2.2538 TIM_CCER_CC2P

```
#define TIM_CCER_CC2P ((uint16_t)0x0020)
```

Capture/Compare 2 output Polarity

5.173.2.2539 TIM_CCER_CC3E

```
#define TIM_CCER_CC3E ((uint16_t)0x0100)
```

Capture/Compare 3 output enable

5.173.2.2540 TIM_CCER_CC3NE

```
#define TIM_CCER_CC3NE ((uint16_t)0x0400)
```

Capture/Compare 3 Complementary output enable

5.173.2.2541 TIM_CCER_CC3NP

```
#define TIM_CCER_CC3NP ((uint16_t)0x0800)
```

Capture/Compare 3 Complementary output Polarity

5.173.2.2542 TIM_CCER_CC3P

```
#define TIM_CCER_CC3P ((uint16_t)0x0200)
```

Capture/Compare 3 output Polarity

5.173.2.2543 TIM_CCER_CC4E

```
#define TIM_CCER_CC4E ((uint16_t)0x1000)
```

Capture/Compare 4 output enable

5.173.2.2544 TIM_CCER_CC4NP

```
#define TIM_CCER_CC4NP ((uint16_t)0x8000)
```

Capture/Compare 4 Complementary output Polarity

5.173.2.2545 TIM_CCER_CC4P

```
#define TIM_CCER_CC4P ((uint16_t)0x2000)
```

Capture/Compare 4 output Polarity

5.173.2.2546 TIM_CCMR1_CC1S

```
#define TIM_CCMR1_CC1S ((uint16_t)0x0003)
```

CC1S[1:0] bits (Capture/Compare 1 Selection)

5.173.2.2547 TIM_CCMR1_CC1S_0

```
#define TIM_CCMR1_CC1S_0 ((uint16_t)0x0001)
```

Bit 0

5.173.2.2548 TIM_CCMR1_CC1S_1

```
#define TIM_CCMR1_CC1S_1 ((uint16_t)0x0002)
```

Bit 1

5.173.2.2549 TIM_CCMR1_CC2S

```
#define TIM_CCMR1_CC2S ((uint16_t)0x0300)
```

CC2S[1:0] bits (Capture/Compare 2 Selection)

5.173.2.2550 TIM_CCMR1_CC2S_0

```
#define TIM_CCMR1_CC2S_0 ((uint16_t)0x0100)
```

Bit 0

5.173.2.2551 TIM_CCMR1_CC2S_1

```
#define TIM_CCMR1_CC2S_1 ((uint16_t)0x0200)
```

Bit 1

5.173.2.2552 TIM_CCMR1_IC1F

```
#define TIM_CCMR1_IC1F ((uint16_t)0x00F0)
```

IC1F[3:0] bits (Input Capture 1 Filter)

5.173.2.2553 TIM_CCMR1_IC1F_0

```
#define TIM_CCMR1_IC1F_0 ((uint16_t)0x0010)
```

Bit 0

5.173.2.2554 TIM_CCMR1_IC1F_1

```
#define TIM_CCMR1_IC1F_1 ((uint16_t)0x0020)
```

Bit 1

5.173.2.2555 TIM_CCMR1_IC1F_2

```
#define TIM_CCMR1_IC1F_2 ((uint16_t)0x0040)
```

Bit 2

5.173.2.2556 TIM_CCMR1_IC1F_3

```
#define TIM_CCMR1_IC1F_3 ((uint16_t)0x0080)
```

Bit 3

5.173.2.2557 TIM_CCMR1_IC1PSC

```
#define TIM_CCMR1_IC1PSC ((uint16_t)0x000C)
```

IC1PSC[1:0] bits (Input Capture 1 Prescaler)

5.173.2.2558 TIM_CCMR1_IC1PSC_0

```
#define TIM_CCMR1_IC1PSC_0 ((uint16_t)0x0004)
```

Bit 0

5.173.2.2559 TIM_CCMR1_IC1PSC_1

```
#define TIM_CCMR1_IC1PSC_1 ((uint16_t)0x0008)
```

Bit 1

5.173.2.2560 TIM_CCMR1_IC2F

```
#define TIM_CCMR1_IC2F ((uint16_t)0xF000)
```

IC2F[3:0] bits (Input Capture 2 Filter)

5.173.2.2561 TIM_CCMR1_IC2F_0

```
#define TIM_CCMR1_IC2F_0 ((uint16_t)0x1000)
```

Bit 0

5.173.2.2562 TIM_CCMR1_IC2F_1

```
#define TIM_CCMR1_IC2F_1 ((uint16_t)0x2000)
```

Bit 1

5.173.2.2563 TIM_CCMR1_IC2F_2

```
#define TIM_CCMR1_IC2F_2 ((uint16_t)0x4000)
```

Bit 2

5.173.2.2564 TIM_CCMR1_IC2F_3

```
#define TIM_CCMR1_IC2F_3 ((uint16_t)0x8000)
```

Bit 3

5.173.2.2565 TIM_CCMR1_IC2PSC

```
#define TIM_CCMR1_IC2PSC ((uint16_t)0x0C00)
```

IC2PSC[1:0] bits (Input Capture 2 Prescaler)

5.173.2.2566 TIM_CCMR1_IC2PSC_0

```
#define TIM_CCMR1_IC2PSC_0 ((uint16_t)0x0400)
```

Bit 0

5.173.2.2567 TIM_CCMR1_IC2PSC_1

```
#define TIM_CCMR1_IC2PSC_1 ((uint16_t)0x0800)
```

Bit 1

5.173.2.2568 TIM_CCMR1_OC1CE

```
#define TIM_CCMR1_OC1CE ((uint16_t)0x0080)
```

Output Compare 1Clear Enable

5.173.2.2569 TIM_CCMR1_OC1FE

```
#define TIM_CCMR1_OC1FE ((uint16_t)0x0004)
```

Output Compare 1 Fast enable

5.173.2.2570 TIM_CCMR1_OC1M

```
#define TIM_CCMR1_OC1M ((uint16_t)0x0070)
```

OC1M[2:0] bits (Output Compare 1 Mode)

5.173.2.2571 TIM_CCMR1_OC1M_0

```
#define TIM_CCMR1_OC1M_0 ((uint16_t)0x0010)
```

Bit 0

5.173.2.2572 TIM_CCMR1_OC1M_1

```
#define TIM_CCMR1_OC1M_1 ((uint16_t)0x0020)
```

Bit 1

5.173.2.2573 TIM_CCMR1_OC1M_2

```
#define TIM_CCMR1_OC1M_2 ((uint16_t)0x0040)
```

Bit 2

5.173.2.2574 TIM_CCMR1_OC1PE

```
#define TIM_CCMR1_OC1PE ((uint16_t)0x0008)
```

Output Compare 1 Preload enable

5.173.2.2575 TIM_CCMR1_OC2CE

```
#define TIM_CCMR1_OC2CE ((uint16_t)0x8000)
```

Output Compare 2 Clear Enable

5.173.2.2576 TIM_CCMR1_OC2FE

```
#define TIM_CCMR1_OC2FE ((uint16_t)0x0400)
```

Output Compare 2 Fast enable

5.173.2.2577 TIM_CCMR1_OC2M

```
#define TIM_CCMR1_OC2M ((uint16_t)0x7000)
```

OC2M[2:0] bits (Output Compare 2 Mode)

5.173.2.2578 TIM_CCMR1_OC2M_0

```
#define TIM_CCMR1_OC2M_0 ((uint16_t)0x1000)
```

Bit 0

5.173.2.2579 TIM_CCMR1_OC2M_1

```
#define TIM_CCMR1_OC2M_1 ((uint16_t)0x2000)
```

Bit 1

5.173.2.2580 TIM_CCMR1_OC2M_2

```
#define TIM_CCMR1_OC2M_2 ((uint16_t)0x4000)
```

Bit 2

5.173.2.2581 TIM_CCMR1_OC2PE

```
#define TIM_CCMR1_OC2PE ((uint16_t)0x0800)
```

Output Compare 2 Preload enable

5.173.2.2582 TIM_CCMR2_CC3S

```
#define TIM_CCMR2_CC3S ((uint16_t)0x0003)
```

CC3S[1:0] bits (Capture/Compare 3 Selection)

5.173.2.2583 TIM_CCMR2_CC3S_0

```
#define TIM_CCMR2_CC3S_0 ((uint16_t)0x0001)
```

Bit 0

5.173.2.2584 TIM_CCMR2_CC3S_1

```
#define TIM_CCMR2_CC3S_1 ((uint16_t)0x0002)
```

Bit 1

5.173.2.2585 TIM_CCMR2_CC4S

```
#define TIM_CCMR2_CC4S ((uint16_t)0x0300)
```

CC4S[1:0] bits (Capture/Compare 4 Selection)

5.173.2.2586 TIM_CCMR2_CC4S_0

```
#define TIM_CCMR2_CC4S_0 ((uint16_t)0x0100)
```

Bit 0

5.173.2.2587 TIM_CCMR2_CC4S_1

```
#define TIM_CCMR2_CC4S_1 ((uint16_t)0x0200)
```

Bit 1

5.173.2.2588 TIM_CCMR2_IC3F

```
#define TIM_CCMR2_IC3F ((uint16_t)0x00F0)
```

IC3F[3:0] bits (Input Capture 3 Filter)

5.173.2.2589 TIM_CCMR2_IC3F_0

```
#define TIM_CCMR2_IC3F_0 ((uint16_t)0x0010)
```

Bit 0

5.173.2.2590 TIM_CCMR2_IC3F_1

```
#define TIM_CCMR2_IC3F_1 ((uint16_t)0x0020)
```

Bit 1

5.173.2.2591 TIM_CCMR2_IC3F_2

```
#define TIM_CCMR2_IC3F_2 ((uint16_t)0x0040)
```

Bit 2

5.173.2.2592 TIM_CCMR2_IC3F_3

```
#define TIM_CCMR2_IC3F_3 ((uint16_t)0x0080)
```

Bit 3

5.173.2.2593 TIM_CCMR2_IC3PSC

```
#define TIM_CCMR2_IC3PSC ((uint16_t)0x000C)
```

IC3PSC[1:0] bits (Input Capture 3 Prescaler)

5.173.2.2594 TIM_CCMR2_IC3PSC_0

```
#define TIM_CCMR2_IC3PSC_0 ((uint16_t)0x0004)
```

Bit 0

5.173.2.2595 TIM_CCMR2_IC3PSC_1

```
#define TIM_CCMR2_IC3PSC_1 ((uint16_t)0x0008)
```

Bit 1

5.173.2.2596 TIM_CCMR2_IC4F

```
#define TIM_CCMR2_IC4F ((uint16_t)0xF000)
```

IC4F[3:0] bits (Input Capture 4 Filter)

5.173.2.2597 TIM_CCMR2_IC4F_0

```
#define TIM_CCMR2_IC4F_0 ((uint16_t)0x1000)
```

Bit 0

5.173.2.2598 TIM_CCMR2_IC4F_1

```
#define TIM_CCMR2_IC4F_1 ((uint16_t)0x2000)
```

Bit 1

5.173.2.2599 TIM_CCMR2_IC4F_2

```
#define TIM_CCMR2_IC4F_2 ((uint16_t)0x4000)
```

Bit 2

5.173.2.2600 TIM_CCMR2_IC4F_3

```
#define TIM_CCMR2_IC4F_3 ((uint16_t)0x8000)
```

Bit 3

5.173.2.2601 TIM_CCMR2_IC4PSC

```
#define TIM_CCMR2_IC4PSC ((uint16_t)0x0C00)
```

IC4PSC[1:0] bits (Input Capture 4 Prescaler)

5.173.2.2602 TIM_CCMR2_IC4PSC_0

```
#define TIM_CCMR2_IC4PSC_0 ((uint16_t)0x0400)
```

Bit 0

5.173.2.2603 TIM_CCMR2_IC4PSC_1

```
#define TIM_CCMR2_IC4PSC_1 ((uint16_t)0x0800)
```

Bit 1

5.173.2.2604 TIM_CCMR2_OC3CE

```
#define TIM_CCMR2_OC3CE ((uint16_t)0x0080)
```

Output Compare 3 Clear Enable

5.173.2.2605 TIM_CCMR2_OC3FE

```
#define TIM_CCMR2_OC3FE ((uint16_t)0x0004)
```

Output Compare 3 Fast enable

5.173.2.2606 TIM_CCMR2_OC3M

```
#define TIM_CCMR2_OC3M ((uint16_t)0x0070)
```

OC3M[2:0] bits (Output Compare 3 Mode)

5.173.2.2607 TIM_CCMR2_OC3M_0

```
#define TIM_CCMR2_OC3M_0 ((uint16_t)0x0010)
```

Bit 0

5.173.2.2608 TIM_CCMR2_OC3M_1

```
#define TIM_CCMR2_OC3M_1 ((uint16_t)0x0020)
```

Bit 1

5.173.2.2609 TIM_CCMR2_OC3M_2

```
#define TIM_CCMR2_OC3M_2 ((uint16_t)0x0040)
```

Bit 2

5.173.2.2610 TIM_CCMR2_OC3PE

```
#define TIM_CCMR2_OC3PE ((uint16_t)0x0008)
```

Output Compare 3 Preload enable

5.173.2.2611 TIM_CCMR2_OC4CE

```
#define TIM_CCMR2_OC4CE ((uint16_t)0x8000)
```

Output Compare 4 Clear Enable

5.173.2.2612 TIM_CCMR2_OC4FE

```
#define TIM_CCMR2_OC4FE ((uint16_t)0x0400)
```

Output Compare 4 Fast enable

5.173.2.2613 TIM_CCMR2_OC4M

```
#define TIM_CCMR2_OC4M ((uint16_t)0x7000)
```

OC4M[2:0] bits (Output Compare 4 Mode)

5.173.2.2614 TIM_CCMR2_OC4M_0

```
#define TIM_CCMR2_OC4M_0 ((uint16_t)0x1000)
```

Bit 0

5.173.2.2615 TIM_CCMR2_OC4M_1

```
#define TIM_CCMR2_OC4M_1 ((uint16_t)0x2000)
```

Bit 1

5.173.2.2616 TIM_CCMR2_OC4M_2

```
#define TIM_CCMR2_OC4M_2 ((uint16_t)0x4000)
```

Bit 2

5.173.2.2617 TIM_CCMR2_OC4PE

```
#define TIM_CCMR2_OC4PE ((uint16_t)0x0800)
```

Output Compare 4 Preload enable

5.173.2.2618 TIM_CCR1_CCR1

```
#define TIM_CCR1_CCR1 ((uint16_t)0xFFFF)
```

Capture/Compare 1 Value

5.173.2.2619 TIM_CCR2_CCR2

```
#define TIM_CCR2_CCR2 ((uint16_t)0xFFFF)
```

Capture/Compare 2 Value

5.173.2.2620 TIM_CCR3_CCR3

```
#define TIM_CCR3_CCR3 ((uint16_t)0xFFFF)
```

Capture/Compare 3 Value

5.173.2.2621 TIM_CCR4_CCR4

```
#define TIM_CCR4_CCR4 ((uint16_t)0xFFFF)
```

Capture/Compare 4 Value

5.173.2.2622 TIM_CNT_CNT

```
#define TIM_CNT_CNT ((uint16_t)0xFFFF)
```

Counter Value

5.173.2.2623 TIM_CR1_ARPE

```
#define TIM_CR1_ARPE ((uint16_t)0x0080)
```

Auto-reload preload enable

5.173.2.2624 TIM_CR1_CEN

```
#define TIM_CR1_CEN ((uint16_t)0x0001)
```

Counter enable

5.173.2.2625 TIM_CR1_CKD

```
#define TIM_CR1_CKD ((uint16_t)0x0300)
```

CKD[1:0] bits (clock division)

5.173.2.2626 TIM_CR1_CKD_0

```
#define TIM_CR1_CKD_0 ((uint16_t)0x0100)
```

Bit 0

5.173.2.2627 TIM_CR1_CKD_1

```
#define TIM_CR1_CKD_1 ((uint16_t)0x0200)
```

Bit 1

5.173.2.2628 TIM_CR1_CMS

```
#define TIM_CR1_CMS ((uint16_t)0x0060)
```

CMS[1:0] bits (Center-aligned mode selection)

5.173.2.2629 TIM_CR1_CMS_0

```
#define TIM_CR1_CMS_0 ((uint16_t)0x0020)
```

Bit 0

5.173.2.2630 TIM_CR1_CMS_1

```
#define TIM_CR1_CMS_1 ((uint16_t)0x0040)
```

Bit 1

5.173.2.2631 TIM_CR1_DIR

```
#define TIM_CR1_DIR ((uint16_t)0x0010)
```

Direction

5.173.2.2632 TIM_CR1_OPM

```
#define TIM_CR1_OPM ((uint16_t)0x0008)
```

One pulse mode

5.173.2.2633 TIM_CR1_UDIS

```
#define TIM_CR1_UDIS ((uint16_t)0x0002)
```

Update disable

5.173.2.2634 TIM_CR1_URS

```
#define TIM_CR1_URS ((uint16_t)0x0004)
```

Update request source

5.173.2.2635 TIM_CR2_CCDS

```
#define TIM_CR2_CCDS ((uint16_t)0x0008)
```

Capture/Compare DMA Selection

5.173.2.2636 TIM_CR2_CCPC

```
#define TIM_CR2_CCPC ((uint16_t)0x0001)
```

Capture/Compare Preloaded Control

5.173.2.2637 TIM_CR2_CCUS

```
#define TIM_CR2_CCUS ((uint16_t)0x0004)
```

Capture/Compare Control Update Selection

5.173.2.2638 TIM_CR2_MMS

```
#define TIM_CR2_MMS ((uint16_t)0x0070)
```

MMS[2:0] bits (Master Mode Selection)

5.173.2.2639 TIM_CR2_MMS_0

```
#define TIM_CR2_MMS_0 ((uint16_t)0x0010)
```

Bit 0

5.173.2.2640 TIM_CR2_MMS_1

```
#define TIM_CR2_MMS_1 ((uint16_t)0x0020)
```

Bit 1

5.173.2.2641 TIM_CR2_MMS_2

```
#define TIM_CR2_MMS_2 ((uint16_t)0x0040)
```

Bit 2

5.173.2.2642 TIM_CR2_OIS1

```
#define TIM_CR2_OIS1 ((uint16_t)0x0100)
```

Output Idle state 1 (OC1 output)

5.173.2.2643 TIM_CR2_OIS1N

```
#define TIM_CR2_OIS1N ((uint16_t)0x0200)
```

Output Idle state 1 (OC1N output)

5.173.2.2644 TIM_CR2_OIS2

```
#define TIM_CR2_OIS2 ((uint16_t)0x0400)
```

Output Idle state 2 (OC2 output)

5.173.2.2645 TIM_CR2_OIS2N

```
#define TIM_CR2_OIS2N ((uint16_t)0x0800)
```

Output Idle state 2 (OC2N output)

5.173.2.2646 TIM_CR2_OIS3

```
#define TIM_CR2_OIS3 ((uint16_t)0x1000)
```

Output Idle state 3 (OC3 output)

5.173.2.2647 TIM_CR2_OIS3N

```
#define TIM_CR2_OIS3N ((uint16_t)0x2000)
```

Output Idle state 3 (OC3N output)

5.173.2.2648 TIM_CR2_OIS4

```
#define TIM_CR2_OIS4 ((uint16_t)0x4000)
```

Output Idle state 4 (OC4 output)

5.173.2.2649 TIM_CR2_TI1S

```
#define TIM_CR2_TI1S ((uint16_t)0x0080)
```

TI1 Selection

5.173.2.2650 TIM_DCR_DBA

```
#define TIM_DCR_DBA ((uint16_t)0x001F)
```

DBA[4:0] bits (DMA Base Address)

5.173.2.2651 TIM_DCR_DBA_0

```
#define TIM_DCR_DBA_0 ((uint16_t)0x0001)
```

Bit 0

5.173.2.2652 TIM_DCR_DBA_1

```
#define TIM_DCR_DBA_1 ((uint16_t)0x0002)
```

Bit 1

5.173.2.2653 TIM_DCR_DBA_2

```
#define TIM_DCR_DBA_2 ((uint16_t)0x0004)
```

Bit 2

5.173.2.2654 TIM_DCR_DBAA_3

```
#define TIM_DCR_DBAA_3 ((uint16_t)0x0008)
```

Bit 3

5.173.2.2655 TIM_DCR_DBAA_4

```
#define TIM_DCR_DBAA_4 ((uint16_t)0x0010)
```

Bit 4

5.173.2.2656 TIM_DCR_DBBL

```
#define TIM_DCR_DBBL ((uint16_t)0x1F00)
```

DBL[4:0] bits (DMA Burst Length)

5.173.2.2657 TIM_DCR_DBBL_0

```
#define TIM_DCR_DBBL_0 ((uint16_t)0x0100)
```

Bit 0

5.173.2.2658 TIM_DCR_DBBL_1

```
#define TIM_DCR_DBBL_1 ((uint16_t)0x0200)
```

Bit 1

5.173.2.2659 TIM_DCR_DBBL_2

```
#define TIM_DCR_DBBL_2 ((uint16_t)0x0400)
```

Bit 2

5.173.2.2660 TIM_DCR_DBBL_3

```
#define TIM_DCR_DBBL_3 ((uint16_t)0x0800)
```

Bit 3

5.173.2.2661 TIM_DCR_DBBL_4

```
#define TIM_DCR_DBBL_4 ((uint16_t)0x1000)
```

Bit 4

5.173.2.2662 TIM_DIER_BIE

```
#define TIM_DIER_BIE ((uint16_t)0x0080)
```

Break interrupt enable

5.173.2.2663 TIM_DIER_CC1DE

```
#define TIM_DIER_CC1DE ((uint16_t)0x0200)
```

Capture/Compare 1 DMA request enable

5.173.2.2664 TIM_DIER_CC1IE

```
#define TIM_DIER_CC1IE ((uint16_t)0x0002)
```

Capture/Compare 1 interrupt enable

5.173.2.2665 TIM_DIER_CC2DE

```
#define TIM_DIER_CC2DE ((uint16_t)0x0400)
```

Capture/Compare 2 DMA request enable

5.173.2.2666 TIM_DIER_CC2IE

```
#define TIM_DIER_CC2IE ((uint16_t)0x0004)
```

Capture/Compare 2 interrupt enable

5.173.2.2667 TIM_DIER_CC3DE

```
#define TIM_DIER_CC3DE ((uint16_t)0x0800)
```

Capture/Compare 3 DMA request enable

5.173.2.2668 TIM_DIER_CC3IE

```
#define TIM_DIER_CC3IE ((uint16_t)0x0008)
```

Capture/Compare 3 interrupt enable

5.173.2.2669 TIM_DIER_CC4DE

```
#define TIM_DIER_CC4DE ((uint16_t)0x1000)
```

Capture/Compare 4 DMA request enable

5.173.2.2670 TIM_DIER_CC4IE

```
#define TIM_DIER_CC4IE ((uint16_t)0x0010)
```

Capture/Compare 4 interrupt enable

5.173.2.2671 TIM_DIER_COMDE

```
#define TIM_DIER_COMDE ((uint16_t)0x2000)
```

COM DMA request enable

5.173.2.2672 TIM_DIER_COMIE

```
#define TIM_DIER_COMIE ((uint16_t)0x0020)
```

COM interrupt enable

5.173.2.2673 TIM_DIER_TDE

```
#define TIM_DIER_TDE ((uint16_t)0x4000)
```

Trigger DMA request enable

5.173.2.2674 TIM_DIER_TIE

```
#define TIM_DIER_TIE ((uint16_t)0x0040)
```

Trigger interrupt enable

5.173.2.2675 TIM_DIER_UDE

```
#define TIM_DIER_UDE ((uint16_t)0x0100)
```

Update DMA request enable

5.173.2.2676 TIM_DIER_UIE

```
#define TIM_DIER_UIE ((uint16_t)0x0001)
```

Update interrupt enable

5.173.2.2677 TIM_DMAR_DMAB

```
#define TIM_DMAR_DMAB ((uint16_t)0xFFFF)
```

DMA register for burst accesses

5.173.2.2678 TIM_EGR_BG

```
#define TIM_EGR_BG ((uint8_t)0x80)
```

Break Generation

5.173.2.2679 TIM_EGR_CC1G

```
#define TIM_EGR_CC1G ((uint8_t)0x02)
```

Capture/Compare 1 Generation

5.173.2.2680 TIM_EGR_CC2G

```
#define TIM_EGR_CC2G ((uint8_t)0x04)
```

Capture/Compare 2 Generation

5.173.2.2681 TIM_EGR_CC3G

```
#define TIM_EGR_CC3G ((uint8_t)0x08)
```

Capture/Compare 3 Generation

5.173.2.2682 TIM_EGR_CC4G

```
#define TIM_EGR_CC4G ((uint8_t)0x10)
```

Capture/Compare 4 Generation

5.173.2.2683 TIM_EGR_COMG

```
#define TIM_EGR_COMG ((uint8_t)0x20)
```

Capture/Compare Control Update Generation

5.173.2.2684 TIM_EGR_TG

```
#define TIM_EGR_TG ((uint8_t)0x40)
```

Trigger Generation

5.173.2.2685 TIM_EGR_UG

```
#define TIM_EGR_UG ((uint8_t)0x01)
```

Update Generation

5.173.2.2686 TIM_OR_ITR1_RMP

```
#define TIM_OR_ITR1_RMP ((uint16_t)0x0C00)
```

ITR1_RMP[1:0] bits (TIM2 Internal trigger 1 remap)

5.173.2.2687 TIM_OR_ITR1_RMP_0

```
#define TIM_OR_ITR1_RMP_0 ((uint16_t)0x0400)
```

Bit 0

5.173.2.2688 TIM_OR_ITR1_RMP_1

```
#define TIM_OR_ITR1_RMP_1 ((uint16_t)0x0800)
```

Bit 1

5.173.2.2689 TIM_OR_TI4_RMP

```
#define TIM_OR_TI4_RMP ((uint16_t)0x00C0)
```

TI4_RMP[1:0] bits (TIM5 Input 4 remap)

5.173.2.2690 TIM_OR_TI4_RMP_0

```
#define TIM_OR_TI4_RMP_0 ((uint16_t)0x0040)
```

Bit 0

5.173.2.2691 TIM_OR_TI4_RMP_1

```
#define TIM_OR_TI4_RMP_1 ((uint16_t)0x0080)
```

Bit 1

5.173.2.2692 TIM_PSC_PSC

```
#define TIM_PSC_PSC ((uint16_t)0xFFFF)
```

Prescaler Value

5.173.2.2693 TIM_RCR REP

```
#define TIM_RCR REP ((uint8_t)0xFF)
```

Repetition Counter Value

5.173.2.2694 TIM_SMCR_ECE

```
#define TIM_SMCR_ECE ((uint16_t)0x4000)
```

External clock enable

5.173.2.2695 TIM_SMCR ETF

```
#define TIM_SMCR ETF ((uint16_t)0x0F00)
```

ETF[3:0] bits (External trigger filter)

5.173.2.2696 TIM_SMCR ETF_0

```
#define TIM_SMCR ETF_0 ((uint16_t)0x0100)
```

Bit 0

5.173.2.2697 TIM_SMCR_ETF_1

```
#define TIM_SMCR_ETF_1 ((uint16_t)0x0200)
```

Bit 1

5.173.2.2698 TIM_SMCR_ETF_2

```
#define TIM_SMCR_ETF_2 ((uint16_t)0x0400)
```

Bit 2

5.173.2.2699 TIM_SMCR_ETF_3

```
#define TIM_SMCR_ETF_3 ((uint16_t)0x0800)
```

Bit 3

5.173.2.2700 TIM_SMCR_ETP

```
#define TIM_SMCR_ETP ((uint16_t)0x8000)
```

External trigger polarity

5.173.2.2701 TIM_SMCR_ETPS

```
#define TIM_SMCR_ETPS ((uint16_t)0x3000)
```

ETPS[1:0] bits (External trigger prescaler)

5.173.2.2702 TIM_SMCR_ETPS_0

```
#define TIM_SMCR_ETPS_0 ((uint16_t)0x1000)
```

Bit 0

5.173.2.2703 TIM_SMCR_ETPS_1

```
#define TIM_SMCR_ETPS_1 ((uint16_t)0x2000)
```

Bit 1

5.173.2.2704 TIM_SMCR_MSM

```
#define TIM_SMCR_MSM ((uint16_t)0x0080)
```

Master/slave mode

5.173.2.2705 TIM_SMCR_SMS

```
#define TIM_SMCR_SMS ((uint16_t)0x0007)
```

SMS[2:0] bits (Slave mode selection)

5.173.2.2706 TIM_SMCR_SMS_0

```
#define TIM_SMCR_SMS_0 ((uint16_t)0x0001)
```

Bit 0

5.173.2.2707 TIM_SMCR_SMS_1

```
#define TIM_SMCR_SMS_1 ((uint16_t)0x0002)
```

Bit 1

5.173.2.2708 TIM_SMCR_SMS_2

```
#define TIM_SMCR_SMS_2 ((uint16_t)0x0004)
```

Bit 2

5.173.2.2709 TIM_SMCR_TS

```
#define TIM_SMCR_TS ((uint16_t)0x0070)
```

TS[2:0] bits (Trigger selection)

5.173.2.2710 TIM_SMCR_TS_0

```
#define TIM_SMCR_TS_0 ((uint16_t)0x0010)
```

Bit 0

5.173.2.2711 TIM_SMCR_TS_1

```
#define TIM_SMCR_TS_1 ((uint16_t)0x0020)
```

Bit 1

5.173.2.2712 TIM_SMCR_TS_2

```
#define TIM_SMCR_TS_2 ((uint16_t)0x0040)
```

Bit 2

5.173.2.2713 TIM_SR_BIF

```
#define TIM_SR_BIF ((uint16_t)0x0080)
```

Break interrupt Flag

5.173.2.2714 TIM_SR_CC1IF

```
#define TIM_SR_CC1IF ((uint16_t)0x0002)
```

Capture/Compare 1 interrupt Flag

5.173.2.2715 TIM_SR_CC1OF

```
#define TIM_SR_CC1OF ((uint16_t)0x0200)
```

Capture/Compare 1 Overcapture Flag

5.173.2.2716 TIM_SR_CC2IF

```
#define TIM_SR_CC2IF ((uint16_t)0x0004)
```

Capture/Compare 2 interrupt Flag

5.173.2.2717 TIM_SR_CC2OF

```
#define TIM_SR_CC2OF ((uint16_t)0x0400)
```

Capture/Compare 2 Overcapture Flag

5.173.2.2718 TIM_SR_CC3IF

```
#define TIM_SR_CC3IF ((uint16_t)0x0008)
```

Capture/Compare 3 interrupt Flag

5.173.2.2719 TIM_SR_CC3OF

```
#define TIM_SR_CC3OF ((uint16_t)0x0800)
```

Capture/Compare 3 Overcapture Flag

5.173.2.2720 TIM_SR_CC4IF

```
#define TIM_SR_CC4IF ((uint16_t)0x0010)
```

Capture/Compare 4 interrupt Flag

5.173.2.2721 TIM_SR_CC4OF

```
#define TIM_SR_CC4OF ((uint16_t)0x1000)
```

Capture/Compare 4 Overcapture Flag

5.173.2.2722 TIM_SR_COMIF

```
#define TIM_SR_COMIF ((uint16_t)0x0020)
```

COM interrupt Flag

5.173.2.2723 TIM_SR_TIF

```
#define TIM_SR_TIF ((uint16_t)0x0040)
```

Trigger interrupt Flag

5.173.2.2724 TIM_SR_UIF

```
#define TIM_SR_UIF ((uint16_t)0x0001)
```

Update interrupt Flag

5.173.2.2725 USART_BRR_DIV_Fraction

```
#define USART_BRR_DIV_Fraction ((uint16_t)0x000F)
```

Fraction of USARTDIV

5.173.2.2726 USART_BRR_DIV_Mantissa

```
#define USART_BRR_DIV_Mantissa ((uint16_t)0xFFFF)
```

Mantissa of USARTDIV

5.173.2.2727 USART_CR1_IDLEIE

```
#define USART_CR1_IDLEIE ((uint16_t)0x0010)
```

IDLE Interrupt Enable

5.173.2.2728 USART_CR1_M

```
#define USART_CR1_M ((uint16_t)0x1000)
```

Word length

5.173.2.2729 USART_CR1_OVER8

```
#define USART_CR1_OVER8 ((uint16_t)0x8000)
```

USART Oversampling by 8 enable

5.173.2.2730 USART_CR1_PCE

```
#define USART_CR1_PCE ((uint16_t)0x0400)
```

Parity Control Enable

5.173.2.2731 USART_CR1_PEIE

```
#define USART_CR1_PEIE ((uint16_t)0x0100)
```

PE Interrupt Enable

5.173.2.2732 USART_CR1_PS

```
#define USART_CR1_PS ((uint16_t)0x0200)
```

Parity Selection

5.173.2.2733 USART_CR1_RE

```
#define USART_CR1_RE ((uint16_t)0x0004)
```

Receiver Enable

5.173.2.2734 USART_CR1_RWU

```
#define USART_CR1_RWU ((uint16_t)0x0002)
```

Receiver wakeup

5.173.2.2735 USART_CR1_RXNEIE

```
#define USART_CR1_RXNEIE ((uint16_t)0x0020)
```

RXNE Interrupt Enable

5.173.2.2736 USART_CR1_SBK

```
#define USART_CR1_SBK ((uint16_t)0x0001)
```

Send Break

5.173.2.2737 USART_CR1_TCIE

```
#define USART_CR1_TCIE ((uint16_t)0x0040)
```

Transmission Complete Interrupt Enable

5.173.2.2738 USART_CR1_TE

```
#define USART_CR1_TE ((uint16_t)0x0008)
```

Transmitter Enable

5.173.2.2739 USART_CR1_TXEIE

```
#define USART_CR1_TXEIE ((uint16_t)0x0080)
```

PE Interrupt Enable

5.173.2.2740 USART_CR1_UE

```
#define USART_CR1_UE ((uint16_t)0x2000)
```

USART Enable

5.173.2.2741 USART_CR1_WAKE

```
#define USART_CR1_WAKE ((uint16_t)0x0800)
```

Wakeup method

5.173.2.2742 USART_CR2_ADD

```
#define USART_CR2_ADD ((uint16_t)0x000F)
```

Address of the USART node

5.173.2.2743 USART_CR2_CLKEN

```
#define USART_CR2_CLKEN ((uint16_t)0x0800)
```

Clock Enable

5.173.2.2744 USART_CR2_CPHA

```
#define USART_CR2_CPHA ((uint16_t)0x0200)
```

Clock Phase

5.173.2.2745 USART_CR2_CPOL

```
#define USART_CR2_CPOL ((uint16_t)0x0400)
```

Clock Polarity

5.173.2.2746 USART_CR2_LBCL

```
#define USART_CR2_LBCL ((uint16_t)0x0100)
```

Last Bit Clock pulse

5.173.2.2747 USART_CR2_LBDIE

```
#define USART_CR2_LBDIE ((uint16_t)0x0040)
```

LIN Break Detection Interrupt Enable

5.173.2.2748 USART_CR2_LBDL

```
#define USART_CR2_LBDL ((uint16_t)0x0020)
```

LIN Break Detection Length

5.173.2.2749 USART_CR2_LINEN

```
#define USART_CR2_LINEN ((uint16_t)0x4000)
```

LIN mode enable

5.173.2.2750 USART_CR2_STOP

```
#define USART_CR2_STOP ((uint16_t)0x3000)
```

STOP[1:0] bits (STOP bits)

5.173.2.2751 USART_CR2_STOP_0

```
#define USART_CR2_STOP_0 ((uint16_t)0x1000)
```

Bit 0

5.173.2.2752 USART_CR2_STOP_1

```
#define USART_CR2_STOP_1 ((uint16_t)0x2000)
```

Bit 1

5.173.2.2753 USART_CR3_CTSE

```
#define USART_CR3_CTSE ((uint16_t)0x0200)
```

CTS Enable

5.173.2.2754 USART_CR3_CTSIE

```
#define USART_CR3_CTSIE ((uint16_t)0x0400)
```

CTS Interrupt Enable

5.173.2.2755 USART_CR3_DMAR

```
#define USART_CR3_DMAR ((uint16_t)0x0040)
```

DMA Enable Receiver

5.173.2.2756 USART_CR3_DMAT

```
#define USART_CR3_DMAT ((uint16_t)0x0080)
```

DMA Enable Transmitter

5.173.2.2757 USART_CR3_EIE

```
#define USART_CR3_EIE ((uint16_t)0x0001)
```

Error Interrupt Enable

5.173.2.2758 USART_CR3_HDSEL

```
#define USART_CR3_HDSEL ((uint16_t)0x0008)
```

Half-Duplex Selection

5.173.2.2759 USART_CR3_IREN

```
#define USART_CR3_IREN ((uint16_t)0x0002)
```

IrDA mode Enable

5.173.2.2760 USART_CR3_IRLP

```
#define USART_CR3_IRLP ((uint16_t)0x0004)
```

IrDA Low-Power

5.173.2.2761 USART_CR3_NACK

```
#define USART_CR3_NACK ((uint16_t)0x0010)
```

Smartcard NACK enable

5.173.2.2762 USART_CR3_ONEBIT

```
#define USART_CR3_ONEBIT ((uint16_t)0x0800)
```

USART One bit method enable

5.173.2.2763 USART_CR3_RTSE

```
#define USART_CR3_RTSE ((uint16_t)0x0100)
```

RTS Enable

5.173.2.2764 USART_CR3_SCEN

```
#define USART_CR3_SCEN ((uint16_t)0x0020)
```

Smartcard mode enable

5.173.2.2765 USART_DR_DR

```
#define USART_DR_DR ((uint16_t)0x01FF)
```

Data value

5.173.2.2766 USART_GTPR_GT

```
#define USART_GTPR_GT ((uint16_t)0xFF00)
```

Guard time value

5.173.2.2767 USART_GTPR_PSC

```
#define USART_GTPR_PSC ((uint16_t)0x00FF)
```

PSC[7:0] bits (Prescaler value)

5.173.2.2768 USART_GTPR_PSC_0

```
#define USART_GTPR_PSC_0 ((uint16_t)0x0001)
```

Bit 0

5.173.2.2769 USART_GTPR_PSC_1

```
#define USART_GTPR_PSC_1 ((uint16_t)0x0002)
```

Bit 1

5.173.2.2770 USART_GTPR_PSC_2

```
#define USART_GTPR_PSC_2 ((uint16_t)0x0004)
```

Bit 2

5.173.2.2771 USART_GTPR_PSC_3

```
#define USART_GTPR_PSC_3 ((uint16_t)0x0008)
```

Bit 3

5.173.2.2772 USART_GTPR_PSC_4

```
#define USART_GTPR_PSC_4 ((uint16_t)0x0010)
```

Bit 4

5.173.2.2773 USART_GTPR_PSC_5

```
#define USART_GTPR_PSC_5 ((uint16_t)0x0020)
```

Bit 5

5.173.2.2774 USART_GTPR_PSC_6

```
#define USART_GTPR_PSC_6 ((uint16_t)0x0040)
```

Bit 6

5.173.2.2775 USART_GTPR_PSC_7

```
#define USART_GTPR_PSC_7 ((uint16_t)0x0080)
```

Bit 7

5.173.2.2776 USART_SR_CTS

```
#define USART_SR_CTS ((uint16_t)0x0200)
```

CTS Flag

5.173.2.2777 USART_SR_FE

```
#define USART_SR_FE ((uint16_t)0x0002)
```

Framing Error

5.173.2.2778 USART_SR_IDLE

```
#define USART_SR_IDLE ((uint16_t)0x0010)
```

IDLE line detected

5.173.2.2779 USART_SR_LBD

```
#define USART_SR_LBD ((uint16_t)0x0100)
```

LIN Break Detection Flag

5.173.2.2780 USART_SR_NE

```
#define USART_SR_NE ((uint16_t)0x0004)
```

Noise Error Flag

5.173.2.2781 USART_SR_ORE

```
#define USART_SR_ORE ((uint16_t)0x0008)
```

OverRun Error

5.173.2.2782 USART_SR_PE

```
#define USART_SR_PE ((uint16_t)0x0001)
```

Parity Error

5.173.2.2783 USART_SR_RXNE

```
#define USART_SR_RXNE ((uint16_t)0x0020)
```

Read Data Register Not Empty

5.173.2.2784 USART_SR_TC

```
#define USART_SR_TC ((uint16_t)0x0040)
```

Transmission Complete

5.173.2.2785 USART_SR_TXE

```
#define USART_SR_TXE ((uint16_t)0x0080)
```

Transmit Data Register Empty

5.173.2.2786 WWDG_CFR_EWI

```
#define WWDG_CFR_EWI ((uint16_t)0x0200)
```

Early Wakeup Interrupt

5.173.2.2787 WWDG_CFR_W

```
#define WWDG_CFR_W ((uint16_t)0x007F)
```

W[6:0] bits (7-bit window value)

5.173.2.2788 WWDG_CFR_W_0

```
#define WWDG_CFR_W_0 ((uint16_t)0x0001)
```

Bit 0

5.173.2.2789 WWDG_CFR_W_1

```
#define WWDG_CFR_W_1 ((uint16_t)0x0002)
```

Bit 1

5.173.2.2790 WWDG_CFR_W_2

```
#define WWDG_CFR_W_2 ((uint16_t)0x0004)
```

Bit 2

5.173.2.2791 WWDG_CFR_W_3

```
#define WWDG_CFR_W_3 ((uint16_t)0x0008)
```

Bit 3

5.173.2.2792 WWDG_CFR_W_4

```
#define WWDG_CFR_W_4 ((uint16_t)0x0010)
```

Bit 4

5.173.2.2793 WWDG_CFR_W_5

```
#define WWDG_CFR_W_5 ((uint16_t)0x0020)
```

Bit 5

5.173.2.2794 WWDG_CFR_W_6

```
#define WWDG_CFR_W_6 ((uint16_t)0x0040)
```

Bit 6

5.173.2.2795 WWDG_CFR_WDGTB

```
#define WWDG_CFR_WDGTB ((uint16_t)0x0180)
```

WDGTB[1:0] bits (Timer Base)

5.173.2.2796 WWDG_CFR_WDGTB_0

```
#define WWDG_CFR_WDGTB_0 ((uint16_t)0x0080)
```

Bit 0

5.173.2.2797 WWDG_CFR_WDGTB_1

```
#define WWDG_CFR_WDGTB_1 ((uint16_t)0x0100)
```

Bit 1

5.173.2.2798 WWDG_CR_T

```
#define WWDG_CR_T ((uint8_t)0x7F)
```

T[6:0] bits (7-Bit counter (MSB to LSB))

5.173.2.2799 WWDG_CR_T_0

```
#define WWDG_CR_T_0 ((uint8_t)0x01)
```

Bit 0

5.173.2.2800 WWDG_CR_T_1

```
#define WWDG_CR_T_1 ((uint8_t)0x02)
```

Bit 1

5.173.2.2801 WWDG_CR_T_2

```
#define WWDG_CR_T_2 ((uint8_t)0x04)
```

Bit 2

5.173.2.2802 WWDG_CR_T_3

```
#define WWDG_CR_T_3 ((uint8_t)0x08)
```

Bit 3

5.173.2.2803 WWDG_CR_T_4

```
#define WWDG_CR_T_4 ((uint8_t)0x10)
```

Bit 4

5.173.2.2804 WWDG_CR_T_5

```
#define WWDG_CR_T_5 ((uint8_t)0x20)
```

Bit 5

5.173.2.2805 WWDG_CR_T_6

```
#define WWDG_CR_T_6 ((uint8_t)0x40)
```

Bit 6

5.173.2.2806 WWDG_CR_WDGA

```
#define WWDG_CR_WDGA ((uint8_t)0x80)
```

Activation bit

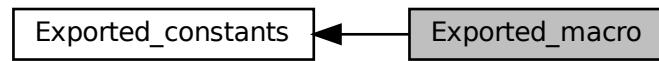
5.173.2.2807 WWDG_SR_EWIF

```
#define WWDG_SR_EWIF ((uint8_t)0x01)
```

Early Wakeup Interrupt Flag

5.174 Exported_macro

Collaboration diagram for Exported_macro:



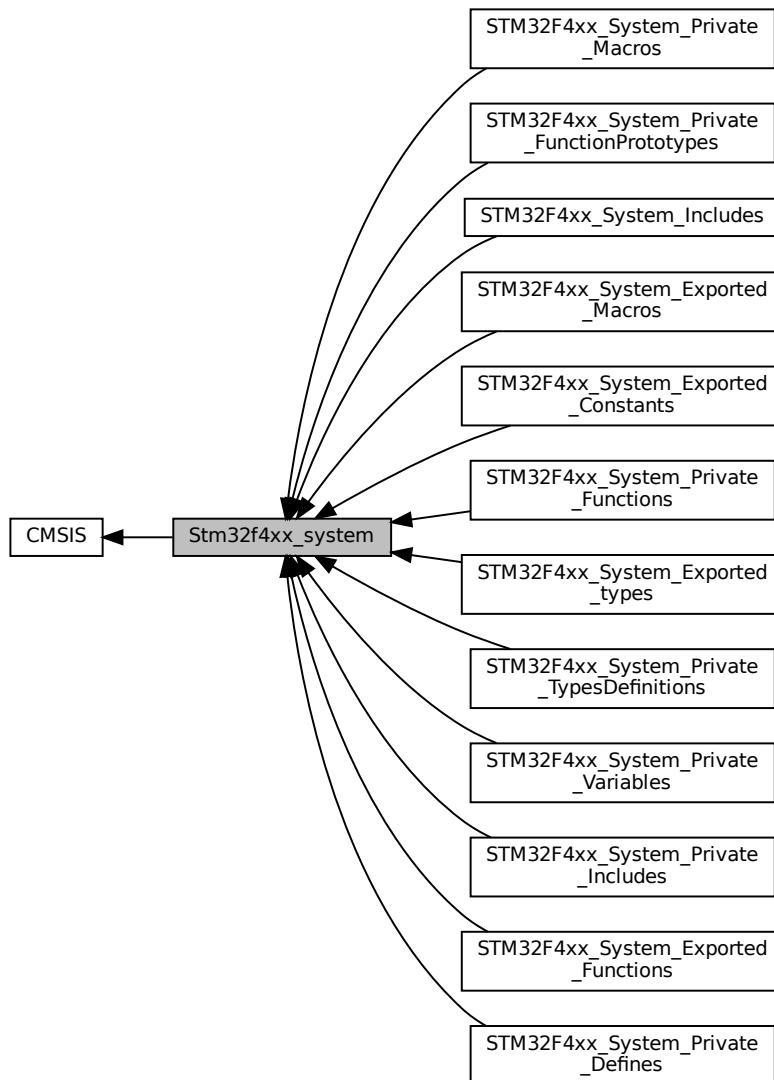
Macros

- #define **SET_BIT**(REG, BIT) ((REG) |= (BIT))
- #define **CLEAR_BIT**(REG, BIT) ((REG) &= ~(BIT))
- #define **READ_BIT**(REG, BIT) ((REG) & (BIT))
- #define **CLEAR_REG**(REG) ((REG) = (0x0))
- #define **WRITE_REG**(REG, VAL) ((REG) = (VAL))
- #define **READ_REG**(REG) ((REG))
- #define **MODIFY_REG**(REG, CLEARMASK, SETMASK) WRITE_REG((REG), (((READ_REG(REG)) & (~(CLEARMASK))) | (SETMASK)))

5.174.1 Detailed Description

5.175 Stm32f4xx_system

Collaboration diagram for Stm32f4xx_system:



Modules

- [STM32F4xx_System_Includes](#)
Define to prevent recursive inclusion.
- [STM32F4xx_System_Exported_types](#)
- [STM32F4xx_System_Exported_Constants](#)
- [STM32F4xx_System_Exported_MACROS](#)
- [STM32F4xx_System_Exported_Functions](#)

- [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](#)

5.175.1 Detailed Description

5.176 STM32F4xx_System_Includes

Define to prevent recursive inclusion.

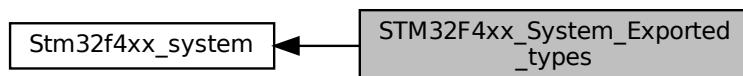
Collaboration diagram for STM32F4xx_System_Includes:



Define to prevent recursive inclusion.

5.177 STM32F4xx_System_Exported_types

Collaboration diagram for STM32F4xx_System_Exported_types:



Variables

- `uint32_t SystemCoreClock`

5.177.1 Detailed Description

5.177.2 Variable Documentation

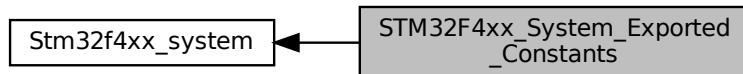
5.177.2.1 SystemCoreClock

```
uint32_t SystemCoreClock [extern]
```

System Clock Frequency (Core Clock)

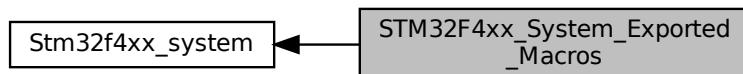
5.178 STM32F4xx_System_Exported_Constants

Collaboration diagram for STM32F4xx_System_Exported_Constants:



5.179 STM32F4xx_System_Exported_Macros

Collaboration diagram for STM32F4xx_System_Exported_Macros:



5.180 STM32F4xx_System_Exported_Functions

Collaboration diagram for STM32F4xx_System_Exported_Functions:



Functions

- void [SystemInit](#) (void)
Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- 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.

5.180.1 Detailed Description

5.180.2 Function Documentation

5.180.2.1 [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.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.h](#) file (default value 25 MHz), user has to ensure that HSE_VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

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

Parameters

None	<input type="button" value=""/>
------	---------------------------------

Return values

None	
------	--

5.180.2.2 SystemInit()

```
void SystemInit (
    void )
```

Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.

Parameters

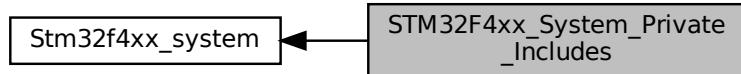
None	
------	--

Return values

None	
------	--

5.181 STM32F4xx_System_Private_Includes

Collaboration diagram for STM32F4xx_System_Private_Includes:



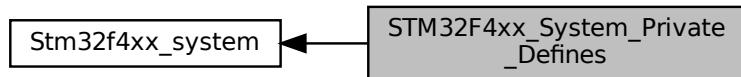
5.182 STM32F4xx_System_Private_TypesDefinitions

Collaboration diagram for STM32F4xx_System_Private_TypesDefinitions:



5.183 STM32F4xx_System_Private_Defines

Collaboration diagram for STM32F4xx_System_Private_Defines:



Macros

- `#define VECT_TAB_OFFSET 0x00`
- `#define PLL_Q 7`

5.183.1 Detailed Description

5.183.2 Macro Definition Documentation

5.183.2.1 VECT_TAB_OFFSET

```
#define VECT_TAB_OFFSET 0x00
```

< Uncomment the following line if you need to use external SRAM or SDRAM mounted on STM324xG_EVAL/STM324x7I_EVAL/STM324x9I_EVAL boards as data memory

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

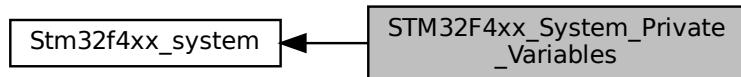
5.184 STM32F4xx_System_Private_Macros

Collaboration diagram for STM32F4xx_System_Private_Macros:



5.185 STM32F4xx_System_Private_Variables

Collaboration diagram for STM32F4xx_System_Private_Variables:



Variables

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

5.185.1 Detailed Description

5.186 STM32F4xx_System_Private_FunctionPrototypes

Collaboration diagram for STM32F4xx_System_Private_FunctionPrototypes:



5.186.1 Detailed Description

5.187 STM32F4xx_System_Private_Functions

Collaboration diagram for STM32F4xx_System_Private_Functions:



Functions

- void **SystemInit** (void)
Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void **SystemInit_Old** (void)
- 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.

5.187.1 Detailed Description

5.187.2 Function Documentation

5.187.2.1 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.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.h](#) file (default value 25 MHz), user has to ensure that HSE_← VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

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

Parameters

None	<input type="button" value=""/>
------	---------------------------------

Return values

None	<input type="button" value=""/>
------	---------------------------------

5.187.2.2 SystemInit()

```
void SystemInit (
    void )
```

Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.

Parameters

None	<input type="button" value=""/>
------	---------------------------------

Return values

None	<input type="button" value=""/>
------	---------------------------------

Chapter 6

Class Documentation

6.1 ADC_Common_TypeDef Struct Reference

Public Attributes

- `__IO uint32_t CSR`
- `__IO uint32_t CCR`
- `__IO uint32_t CDR`

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

- CMSIS/Inc/stm32f4xx.h

6.2 ADC_TypeDef Struct Reference

Analog to Digital Converter

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t SR`
- `__IO uint32_t CR1`
- `__IO uint32_t CR2`
- `__IO uint32_t SMPR1`
- `__IO uint32_t SMPR2`
- `__IO uint32_t JOFR1`
- `__IO uint32_t JOFR2`
- `__IO uint32_t JOFR3`
- `__IO uint32_t JOFR4`
- `__IO uint32_t HTR`
- `__IO uint32_t LTR`
- `__IO uint32_t SQR1`
- `__IO uint32_t SQR2`
- `__IO uint32_t SQR3`
- `__IO uint32_t JSQR`
- `__IO uint32_t JDR1`
- `__IO uint32_t JDR2`
- `__IO uint32_t JDR3`
- `__IO uint32_t JDR4`
- `__IO uint32_t DR`

6.2.1 Detailed Description

Analog to Digital Converter

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

- CMSIS/Inc/stm32f4xx.h

6.3 APSR_Type Union Reference

Union type to access the Application Program Status Register (APSR).

```
#include <core_cm4.h>
```

Public Attributes

- struct {
 uint32_t APSR_Type::_reserved0:16
 uint32_t APSR_Type::GE:4
 uint32_t APSR_Type::_reserved1:7
 uint32_t APSR_Type::Q:1
 uint32_t APSR_Type::V:1
 uint32_t APSR_Type::C:1
 uint32_t APSR_Type::Z:1
 uint32_t APSR_Type::N:1
} b
- uint32_t w

6.3.1 Detailed Description

Union type to access the Application Program Status Register (APSR).

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

- CMSIS/Inc/core_cm4.h

6.4 CAN_FIFOMailBox_TypeDef Struct Reference

Controller Area Network FIFOMailBox.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t RIR`
- `__IO uint32_t RDTR`
- `__IO uint32_t RDLR`
- `__IO uint32_t RDHR`

6.4.1 Detailed Description

Controller Area Network FIFOMailBox.

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

- CMSIS/Inc/stm32f4xx.h

6.5 CAN_FilterRegister_TypeDef Struct Reference

Controller Area Network FilterRegister.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t FR1`
- `__IO uint32_t FR2`

6.5.1 Detailed Description

Controller Area Network FilterRegister.

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

- CMSIS/Inc/stm32f4xx.h

6.6 CAN_TxMailBox_TypeDef Struct Reference

Controller Area Network TxMailBox.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t TIR`
- `__IO uint32_t TDTR`
- `__IO uint32_t TDLR`
- `__IO uint32_t TDHR`

6.6.1 Detailed Description

Controller Area Network TxMailBox.

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

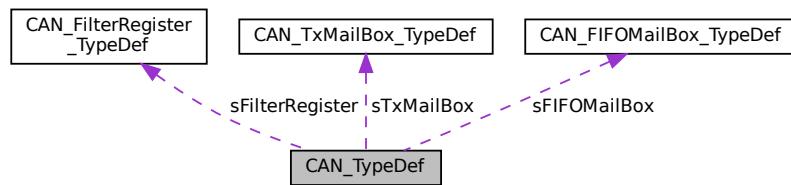
- CMSIS/Inc/stm32f4xx.h

6.7 CAN_TypeDef Struct Reference

Controller Area Network.

```
#include <stm32f4xx.h>
```

Collaboration diagram for CAN_TypeDef:



Public Attributes

- `__IO uint32_t MCR`
- `__IO uint32_t MSR`
- `__IO uint32_t TSR`
- `__IO uint32_t RFOR`
- `__IO uint32_t RF1R`
- `__IO uint32_t IER`
- `__IO uint32_t ESR`
- `__IO uint32_t BTR`
- `uint32_t RESERVED0 [88]`
- `CAN_TxMailBox_TypeDef sTxMailBox [3]`
- `CAN_FIFOMailBox_TypeDef sFIFOMailBox [2]`
- `uint32_t RESERVED1 [12]`
- `__IO uint32_t FMR`
- `__IO uint32_t FM1R`
- `uint32_t RESERVED2`
- `__IO uint32_t FS1R`
- `uint32_t RESERVED3`
- `__IO uint32_t FFA1R`
- `uint32_t RESERVED4`
- `__IO uint32_t FA1R`
- `uint32_t RESERVED5 [8]`
- `CAN_FilterRegister_TypeDef sFilterRegister [28]`

6.7.1 Detailed Description

Controller Area Network.

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

- CMSIS/Inc/stm32f4xx.h

6.8 CONTROL_Type Union Reference

Union type to access the Control Registers (CONTROL).

```
#include <core_cm4.h>
```

Public Attributes

- struct {
 uint32_t CONTROL_Type::nPRIV:1
 uint32_t CONTROL_Type::SPSEL:1
 uint32_t CONTROL_Type::FPCA:1
 uint32_t CONTROL_Type::_reserved0:29
} **b**
- uint32_t **w**

6.8.1 Detailed Description

Union type to access the Control Registers (CONTROL).

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

- CMSIS/Inc/core_cm4.h

6.9 CoreDebug_Type Struct Reference

Structure type to access the Core Debug Register (CoreDebug).

```
#include <core_cm4.h>
```

Public Attributes

- **_IO** uint32_t **DHCSR**
- **_IO** uint32_t **DCRSR**
- **_IO** uint32_t **DCRDR**
- **_IO** uint32_t **DEMCR**

6.9.1 Detailed Description

Structure type to access the Core Debug Register (CoreDebug).

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

- CMSIS/Inc/core_cm4.h

6.10 CRC_TypeDef Struct Reference

CRC calculation unit.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t DR`
- `__IO uint8_t IDR`
- `uint8_t RESERVED0`
- `uint16_t RESERVED1`
- `__IO uint32_t CR`
- `uint32 CR`
- `uint32 SR`
- `uint32 DR`
- `uint32 IDR`
- `uint32 POL`
- `uint32 RESERVED`
- `uint32 INIT`

6.10.1 Detailed Description

CRC calculation unit.

6.10.2 Member Data Documentation

6.10.2.1 CR

```
uint32 CRC_TypeDef::CR
```

CRC Control Register

6.10.2.2 DR

`uint32_t CRC_TypeDef::DR`

CRC Data Register

6.10.2.3 IDR

`uint32_t CRC_TypeDef::IDR`

CRC Independent Data Register

6.10.2.4 INIT

`uint32_t CRC_TypeDef::INIT`

CRC Initial Value Register

6.10.2.5 POL

`uint32_t CRC_TypeDef::POL`

CRC Polynomial Register

6.10.2.6 RESERVED

`uint32_t CRC_TypeDef::RESERVED`

Reserved

6.10.2.7 SR

`uint32_t CRC_TypeDef::SR`

CRC Status Register

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

- CMSIS/Inc/[stm32f4xx.h](#)
- drivers/Inc/[stm32f401xx.h](#)

6.11 CRYP_TypeDef Struct Reference

Crypto Processor.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t SR`
- `__IO uint32_t DR`
- `__IO uint32_t DOUT`
- `__IO uint32_t DMACR`
- `__IO uint32_t IMSCR`
- `__IO uint32_t RISR`
- `__IO uint32_t MISR`
- `__IO uint32_t K0LR`
- `__IO uint32_t K0RR`
- `__IO uint32_t K1LR`
- `__IO uint32_t K1RR`
- `__IO uint32_t K2LR`
- `__IO uint32_t K2RR`
- `__IO uint32_t K3LR`
- `__IO uint32_t K3RR`
- `__IO uint32_t IV0LR`
- `__IO uint32_t IV0RR`
- `__IO uint32_t IV1LR`
- `__IO uint32_t IV1RR`
- `__IO uint32_t CSGCMCCM0R`
- `__IO uint32_t CSGCMCCM1R`
- `__IO uint32_t CSGCMCCM2R`
- `__IO uint32_t CSGCMCCM3R`
- `__IO uint32_t CSGCMCCM4R`
- `__IO uint32_t CSGCMCCM5R`
- `__IO uint32_t CSGCMCCM6R`
- `__IO uint32_t CSGCMCCM7R`
- `__IO uint32_t CSGCM0R`
- `__IO uint32_t CSGCM1R`
- `__IO uint32_t CSGCM2R`
- `__IO uint32_t CSGCM3R`
- `__IO uint32_t CSGCM4R`
- `__IO uint32_t CSGCM5R`
- `__IO uint32_t CSGCM6R`
- `__IO uint32_t CSGCM7R`

6.11.1 Detailed Description

Crypto Processor.

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

- CMSIS/Inc/stm32f4xx.h

6.12 DAC_TypeDef Struct Reference

Digital to Analog Converter.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t SWTRIGR`
- `__IO uint32_t DHR12R1`
- `__IO uint32_t DHR12L1`
- `__IO uint32_t DHR8R1`
- `__IO uint32_t DHR12R2`
- `__IO uint32_t DHR12L2`
- `__IO uint32_t DHR8R2`
- `__IO uint32_t DHR12RD`
- `__IO uint32_t DHR12LD`
- `__IO uint32_t DHR8RD`
- `__IO uint32_t DOR1`
- `__IO uint32_t DOR2`
- `__IO uint32_t SR`

6.12.1 Detailed Description

Digital to Analog Converter.

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

- CMSIS/Inc/stm32f4xx.h

6.13 DBGMCU_TypeDef Struct Reference

Debug MCU.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t IDCODE`
- `__IO uint32_t CR`
- `__IO uint32_t APB1FZ`
- `__IO uint32_t APB2FZ`

6.13.1 Detailed Description

Debug MCU.

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

- CMSIS/Inc/stm32f4xx.h

6.14 DCMI_TypeDef Struct Reference

DCMI.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t SR`
- `__IO uint32_t RISR`
- `__IO uint32_t IER`
- `__IO uint32_t MISR`
- `__IO uint32_t ICR`
- `__IO uint32_t ESCR`
- `__IO uint32_t ESUR`
- `__IO uint32_t CWSTRTR`
- `__IO uint32_t CWSIZER`
- `__IO uint32_t DR`

6.14.1 Detailed Description

DCMI.

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

- CMSIS/Inc/stm32f4xx.h

6.15 DMA2D_TypeDef Struct Reference

DMA2D Controller.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t ISR`
- `__IO uint32_t IFCR`
- `__IO uint32_t FGMAR`
- `__IO uint32_t FGOR`
- `__IO uint32_t BGMAR`
- `__IO uint32_t BGOR`
- `__IO uint32_t FGPCCR`
- `__IO uint32_t FGCOLR`
- `__IO uint32_t BGPFCCR`
- `__IO uint32_t BGCOLR`
- `__IO uint32_t FGCMAR`

- `__IO uint32_t BGCMAR`
- `__IO uint32_t OPFCCR`
- `__IO uint32_t OCOLR`
- `__IO uint32_t OMAR`
- `__IO uint32_t OOR`
- `__IO uint32_t NLR`
- `__IO uint32_t LWR`
- `__IO uint32_t AMTCR`
- `uint32_t RESERVED [236]`
- `__IO uint32_t FGCLUT [256]`
- `__IO uint32_t BGCLUT [256]`

6.15.1 Detailed Description

DMA2D Controller.

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

- CMSIS/Inc/stm32f4xx.h

6.16 DMA_Stream_TypeDef Struct Reference

DMA Controller.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t NDTR`
- `__IO uint32_t PAR`
- `__IO uint32_t M0AR`
- `__IO uint32_t M1AR`
- `__IO uint32_t FCR`

6.16.1 Detailed Description

DMA Controller.

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

- CMSIS/Inc/stm32f4xx.h

6.17 DMA_TypeDef Struct Reference

Public Attributes

- `__IO uint32_t LISR`
- `__IO uint32_t HISR`
- `__IO uint32_t LIFCR`
- `__IO uint32_t HIFCR`

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

- CMSIS/Inc/stm32f4xx.h

6.18 DWT_TypeDef Struct Reference

Structure type to access the Data Watchpoint and Trace Register (DWT).

```
#include <core_cm4.h>
```

Public Attributes

- `__IO uint32_t CTRL`
- `__IO uint32_t CYCCNT`
- `__IO uint32_t CPICNT`
- `__IO uint32_t EXCCNT`
- `__IO uint32_t SLEEPCNT`
- `__IO uint32_t LSUCNT`
- `__IO uint32_t FOLDCNT`
- `__I uint32_t PCSR`
- `__IO uint32_t COMP0`
- `__IO uint32_t MASK0`
- `__IO uint32_t FUNCTION0`
- `uint32_t RESERVED0 [1]`
- `__IO uint32_t COMP1`
- `__IO uint32_t MASK1`
- `__IO uint32_t FUNCTION1`
- `uint32_t RESERVED1 [1]`
- `__IO uint32_t COMP2`
- `__IO uint32_t MASK2`
- `__IO uint32_t FUNCTION2`
- `uint32_t RESERVED2 [1]`
- `__IO uint32_t COMP3`
- `__IO uint32_t MASK3`
- `__IO uint32_t FUNCTION3`

6.18.1 Detailed Description

Structure type to access the Data Watchpoint and Trace Register (DWT).

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

- CMSIS/Inc/core_cm4.h

6.19 ETH_TypeDef Struct Reference

Ethernet MAC.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t MACCR`
- `__IO uint32_t MACFFR`
- `__IO uint32_t MACHTHR`
- `__IO uint32_t MACHTLR`
- `__IO uint32_t MACMIIAR`
- `__IO uint32_t MACMIIDR`
- `__IO uint32_t MACFCR`
- `__IO uint32_t MACVLANTR`
- `uint32_t RESERVED0 [2]`
- `__IO uint32_t MACRWUFFR`
- `__IO uint32_t MACPMTCSR`
- `uint32_t RESERVED1 [2]`
- `__IO uint32_t MACSR`
- `__IO uint32_t MACIMR`
- `__IO uint32_t MACA0HR`
- `__IO uint32_t MACA0LR`
- `__IO uint32_t MACA1HR`
- `__IO uint32_t MACA1LR`
- `__IO uint32_t MACA2HR`
- `__IO uint32_t MACA2LR`
- `__IO uint32_t MACA3HR`
- `__IO uint32_t MACA3LR`
- `uint32_t RESERVED2 [40]`
- `__IO uint32_t MMCCR`
- `__IO uint32_t MMCRIR`
- `__IO uint32_t MMCTIR`
- `__IO uint32_t MMCRIMR`
- `__IO uint32_t MMCTIMR`
- `uint32_t RESERVED3 [14]`
- `__IO uint32_t MMCTGFSCCR`
- `__IO uint32_t MMCTGFMSCCR`
- `uint32_t RESERVED4 [5]`
- `__IO uint32_t MMCTGFCSR`
- `uint32_t RESERVED5 [10]`
- `__IO uint32_t MMCRFCECR`

- `__IO uint32_t MMCRAECSR`
- `uint32_t RESERVED6 [10]`
- `__IO uint32_t MMCRGUFCR`
- `uint32_t RESERVED7 [334]`
- `__IO uint32_t PTPTSCR`
- `__IO uint32_t PTSSIR`
- `__IO uint32_t PTPTSHR`
- `__IO uint32_t PTPTSLR`
- `__IO uint32_t PTPTSHUR`
- `__IO uint32_t PTPTSLUR`
- `__IO uint32_t PTPTSAR`
- `__IO uint32_t PTPTTHR`
- `__IO uint32_t PTPTTLR`
- `__IO uint32_t RESERVED8`
- `__IO uint32_t PTPTSSR`
- `uint32_t RESERVED9 [565]`
- `__IO uint32_t DMABMR`
- `__IO uint32_t DMATPDR`
- `__IO uint32_t DMARPDR`
- `__IO uint32_t DMARDLAR`
- `__IO uint32_t DMATDLAR`
- `__IO uint32_t DMASTR`
- `__IO uint32_t DMAOMR`
- `__IO uint32_t DMAIER`
- `__IO uint32_t DMAMFBOCR`
- `__IO uint32_t DMARSWTR`
- `uint32_t RESERVED10 [8]`
- `__IO uint32_t DMACHTDR`
- `__IO uint32_t DMACHRDR`
- `__IO uint32_t DMACHTBAR`
- `__IO uint32_t DMACHRBAR`

6.19.1 Detailed Description

Ethernet MAC.

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

- CMSIS/Inc/stm32f4xx.h

6.20 EXTI_TypeDef Struct Reference

External Interrupt/Event Controller.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t IMR`
- `__IO uint32_t EMR`
- `__IO uint32_t RTSR`
- `__IO uint32_t FTSR`
- `__IO uint32_t SWIER`
- `__IO uint32_t PR`
- `vuint32_t IMR`
- `vuint32_t EMR`
- `vuint32_t RTSR`
- `vuint32_t FTSR`
- `vuint32_t SWIER`
- `vuint32_t PR`

6.20.1 Detailed Description

External Interrupt/Event Controller.

6.20.2 Member Data Documentation

6.20.2.1 EMR

`vuint32_t EXTI_TypeDef::EMR`

Event Mask Register

6.20.2.2 FTSR

`vuint32_t EXTI_TypeDef::FTSR`

Falling Trigger Selection Register

6.20.2.3 IMR

`vuint32_t EXTI_TypeDef::IMR`

Interrupt Mask Register

6.20.2.4 PR

`vuint32_t EXTI_TypeDef::PR`

Pending Register

6.20.2.5 RTSR

```
vuint32_t EXTI_TypeDef::RTSR
```

Rising Trigger Selection Register

6.20.2.6 SWIER

```
vuint32_t EXTI_TypeDef::SWIER
```

Software Interrupt Event Register

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

- CMSIS/Inc/[stm32f4xx.h](#)
- drivers/Inc/[stm32f401xx.h](#)

6.21 FLASH_TypeDef Struct Reference

FLASH Registers.

```
#include <stm32f4xx.h>
```

Public Attributes

- [__IO uint32_t ACR](#)
- [__IO uint32_t KEYR](#)
- [__IO uint32_t OPTKEYR](#)
- [__IO uint32_t SR](#)
- [__IO uint32_t CR](#)
- [__IO uint32_t OPTCR](#)
- [__IO uint32_t OPTCR1](#)

6.21.1 Detailed Description

FLASH Registers.

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

- CMSIS/Inc/[stm32f4xx.h](#)

6.22 GPIO_PinConfig_t Struct Reference

Configuration structure for GPIO pins.

```
#include <stm32f401xx_gpio_driver.h>
```

Public Attributes

- `uint8 GPIO_MODE`
- `uint8 GPIO_PinSpeed`
- `uint16 GPIO_PinNumber`
- `uint32 GPIO_PinPuPdControl`
- `uint32 GPIO_PinOPType`
- `uint32 GPIO_PinAltFunMode`

6.22.1 Detailed Description

Configuration structure for GPIO pins.

This structure defines the configuration parameters for a GPIO pin, including mode, speed, pull-up/pull-down configuration, output type, and alternate function mode.

6.22.2 Member Data Documentation

6.22.2.1 GPIO_MODE

`uint8 GPIO_PinConfig_t::GPIO_MODE`

Specifies the operating mode for the selected pins. This parameter can be a value of [GPIO Pin Modes](#)

6.22.2.2 GPIO_PinAltFunMode

`uint32 GPIO_PinConfig_t::GPIO_PinAltFunMode`

Specifies the alternate function mode of the GPIO pin.

6.22.2.3 GPIO_PinNumber

`uint16 GPIO_PinConfig_t::GPIO_PinNumber`

Specifies the GPIO pins to be configured. This parameter can be a value of [GPIO Pin Numbers](#)

6.22.2.4 GPIO_PinOPType

`uint32 GPIO_PinConfig_t::GPIO_PinOPType`

Specifies the output type of the GPIO pin.

6.22.2.5 GPIO_PinPuPdControl

```
uint32_t GPIO_PinConfig_t::GPIO_PinPuPdControl
```

Specifies the pull-up/pull-down configuration for the GPIO pin.

6.22.2.6 GPIO_PinSpeed

```
uint8_t GPIO_PinConfig_t::GPIO_PinSpeed
```

Specifies the speed for the selected pins. This parameter can be a value of [GPIO Pin Speeds](#)

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

- drivers/Inc/stm32f401xx_gpio_driver.h

6.23 GPIO_TypeDef Struct Reference

General Purpose I/O.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t MODER`
- `__IO uint32_t OTYPER`
- `__IO uint32_t OSPEEDR`
- `__IO uint32_t PUPDR`
- `__IO uint32_t IDR`
- `__IO uint32_t ODR`
- `__IO uint16_t BSRRRL`
- `__IO uint16_t BSRRRH`
- `__IO uint32_t LCKR`
- `__IO uint32_t AFR [2]`
- `vuint32_t MODER`
- `vuint32_t OTYPER`
- `vuint32_t OSPEEDR`
- `vuint32_t PUPDR`
- `vuint32_t IDR`
- `vuint32_t ODR`
- `vuint32_t BSRR`
- `vuint32_t LCKR`
- `vuint32_t AFR [2]`

6.23.1 Detailed Description

General Purpose I/O.

6.23.2 Member Data Documentation

6.23.2.1 AFR

```
vuint32_t GPIO_TypeDef::AFR[2]
```

Alternate Function Registers

6.23.2.2 BSRR

```
vuint32_t GPIO_TypeDef::BSRR
```

Bit Set/Reset Register

6.23.2.3 IDR

```
vuint32_t GPIO_TypeDef::IDR
```

Input Data Register

6.23.2.4 LCKR

```
vuint32_t GPIO_TypeDef::LCKR
```

Configuration Lock Register

6.23.2.5 MODER

```
vuint32_t GPIO_TypeDef::MODER
```

Mode Register

6.23.2.6 ODR

```
vuint32_t GPIO_TypeDef::ODR
```

Output Data Register

6.23.2.7 OSPEEDR

```
vuint32_t GPIO_TypeDef::OSPEEDR
```

Output Speed Register

6.23.2.8 OTYPER

```
vuint32_t GPIO_TypeDef::OTYPER
```

Output Type Register

6.23.2.9 PUPDR

```
vuint32_t GPIO_TypeDef::PUPDR
```

Pull-up/Pull-down Register

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

- CMSIS/Inc/[stm32f4xx.h](#)
- drivers/Inc/[stm32f401xx.h](#)

6.24 HASH_DIGEST_TypeDef Struct Reference

HASH_DIGEST.

```
#include <stm32f4xx.h>
```

Public Attributes

- [__IO](#) uint32_t [HR](#) [8]

6.24.1 Detailed Description

HASH_DIGEST.

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

- CMSIS/Inc/[stm32f4xx.h](#)

6.25 HASH_TypeDef Struct Reference

HASH.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t DIN`
- `__IO uint32_t STR`
- `__IO uint32_t HR [5]`
- `__IO uint32_t IMR`
- `__IO uint32_t SR`
- `uint32_t RESERVED [52]`
- `__IO uint32_t CSR [54]`

6.25.1 Detailed Description

HASH.

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

- CMSIS/Inc/stm32f4xx.h

6.26 I2C_TypeDef Struct Reference

Inter-integrated Circuit Interface.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint16_t CR1`
- `uint16_t RESERVED0`
- `__IO uint16_t CR2`
- `uint16_t RESERVED1`
- `__IO uint16_t OAR1`
- `uint16_t RESERVED2`
- `__IO uint16_t OAR2`
- `uint16_t RESERVED3`
- `__IO uint16_t DR`
- `uint16_t RESERVED4`
- `__IO uint16_t SR1`
- `uint16_t RESERVED5`
- `__IO uint16_t SR2`
- `uint16_t RESERVED6`
- `__IO uint16_t CCR`
- `uint16_t RESERVED7`
- `__IO uint16_t TRISE`
- `uint16_t RESERVED8`
- `__IO uint16_t FLTR`
- `uint16_t RESERVED9`
- `uint32 CR1`
- `uint32 CR2`
- `uint32 OAR1`
- `uint32 OAR2`
- `uint32 DR`
- `uint32 SR1`
- `uint32 SR2`
- `uint32 CCR`
- `uint32 TRISE`

6.26.1 Detailed Description

Inter-integrated Circuit Interface.

6.26.2 Member Data Documentation

6.26.2.1 CCR

```
uint32 I2C_TypeDef::CCR
```

I2C Clock Control Register

6.26.2.2 CR1

```
uint32 I2C_TypeDef::CR1
```

I2C Control Register 1

6.26.2.3 CR2

```
uint32 I2C_TypeDef::CR2
```

I2C Control Register 2

6.26.2.4 DR

```
uint32 I2C_TypeDef::DR
```

I2C Data Register

6.26.2.5 OAR1

```
uint32 I2C_TypeDef::OAR1
```

I2C Own Address Register 1

6.26.2.6 OAR2

```
uint32 I2C_TypeDef::OAR2
```

I2C Own Address Register 2

6.26.2.7 SR1

```
uint32_t I2C_TypeDef::SR1
```

I2C Status Register 1

6.26.2.8 SR2

```
uint32_t I2C_TypeDef::SR2
```

I2C Status Register 2

6.26.2.9 TRISE

```
uint32_t I2C_TypeDef::TRISE
```

I2C TRISE Register

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

- CMSIS/Inc/[stm32f4xx.h](#)
- drivers/Inc/[stm32f401xx.h](#)

6.27 IPSR_Type Union Reference

Union type to access the Interrupt Program Status Register (IPSR).

```
#include <core_cm4.h>
```

Public Attributes

- struct {
 uint32_t IPSR_Type::ISR:9
 uint32_t IPSR_Type::_reserved0:23
} **b**
- uint32_t **w**

6.27.1 Detailed Description

Union type to access the Interrupt Program Status Register (IPSR).

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

- CMSIS/Inc/[core_cm4.h](#)

6.28 ITM_Type Struct Reference

Structure type to access the Instrumentation Trace Macrocell Register (ITM).

```
#include <core_cm4.h>
```

Public Attributes

- union {
 - __O uint8_t ITM_Type::u8
 - __O uint16_t ITM_Type::u16
 - __O uint32_t ITM_Type::u32}
- PORT [32]
- uint32_t RESERVED0 [864]
- __IO uint32_t TER
- uint32_t RESERVED1 [15]
- __IO uint32_t TPR
- uint32_t RESERVED2 [15]
- __IO uint32_t TCR
- uint32_t RESERVED3 [29]
- __O uint32_t IWR
- __I uint32_t IRR
- __IO uint32_t IMCR
- uint32_t RESERVED4 [43]
- __O uint32_t LAR
- __I uint32_t LSR
- uint32_t RESERVED5 [6]
- __I uint32_t PID4
- __I uint32_t PID5
- __I uint32_t PID6
- __I uint32_t PID7
- __I uint32_t PID0
- __I uint32_t PID1
- __I uint32_t PID2
- __I uint32_t PID3
- __I uint32_t CID0
- __I uint32_t CID1
- __I uint32_t CID2
- __I uint32_t CID3

6.28.1 Detailed Description

Structure type to access the Instrumentation Trace Macrocell Register (ITM).

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

- CMSIS/Inc/core_cm4.h

6.29 IWDG_TypeDef Struct Reference

Independent WATCHDOG.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t KR`
- `__IO uint32_t PR`
- `__IO uint32_t RLR`
- `__IO uint32_t SR`

6.29.1 Detailed Description

Independent WATCHDOG.

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

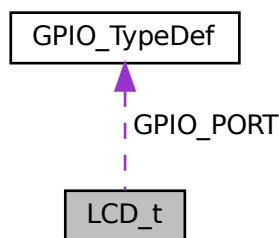
- CMSIS/Inc/[stm32f4xx.h](#)

6.30 LCD_t Struct Reference

Structure for configuring the LCD.

```
#include <lcd_driver.h>
```

Collaboration diagram for LCD_t:



Public Attributes

- LCD_MODE_t Mode
- LCD_ROWS_t Rows
- uint8 Display_Mode
- uint8 Entry_Mode
- GPIO_TypeDef * GPIO_PORT
- uint16 RS_PIN
- uint16 EN_PIN
- uint16 D0_PIN
- uint16 D1_PIN
- uint16 D2_PIN
- uint16 D3_PIN
- uint16 D4_PIN
- uint16 D5_PIN
- uint16 D6_PIN
- uint16 D7_PIN

6.30.1 Detailed Description

Structure for configuring the LCD.

6.30.2 Member Data Documentation

6.30.2.1 D0_PIN

`uint16 LCD_t::D0_PIN`

GPIO pin for Data pin 0. (see [GPIO Pin Numbers](#))

6.30.2.2 D1_PIN

`uint16 LCD_t::D1_PIN`

GPIO pin for Data pin 1. (see [GPIO Pin Numbers](#))

6.30.2.3 D2_PIN

`uint16 LCD_t::D2_PIN`

GPIO pin for Data pin 2. (see [GPIO Pin Numbers](#))

6.30.2.4 D3_PIN

```
uint16 LCD_t::D3_PIN
```

GPIO pin for Data pin 3. (see [GPIO Pin Numbers](#))

6.30.2.5 D4_PIN

```
uint16 LCD_t::D4_PIN
```

GPIO pin for Data pin 4. (see [GPIO Pin Numbers](#))

6.30.2.6 D5_PIN

```
uint16 LCD_t::D5_PIN
```

GPIO pin for Data pin 5. (see [GPIO Pin Numbers](#))

6.30.2.7 D6_PIN

```
uint16 LCD_t::D6_PIN
```

GPIO pin for Data pin 6. (see [GPIO Pin Numbers](#))

6.30.2.8 D7_PIN

```
uint16 LCD_t::D7_PIN
```

GPIO pin for Data pin 7. (see [GPIO Pin Numbers](#))

6.30.2.9 Display_Mode

```
uint8 LCD_t::Display_Mode
```

Display mode configuration (see [LCD Commands](#)).

6.30.2.10 EN_PIN

```
uint16 LCD_t::EN_PIN
```

GPIO pin for Enable (EN). (see [GPIO Pin Numbers](#))

6.30.2.11 Entry_Mode

```
uint8 LCD_t::Entry_Mode
```

Entry mode configuration (see [LCD Commands](#)).

6.30.2.12 GPIO_PORT

```
GPIO_TypeDef* LCD_t::GPIO_PORT
```

GPIO port used for LCD control.

6.30.2.13 Mode

```
LCD_MODE_t LCD_t::Mode
```

LCD mode (8-bit or 4-bit).

6.30.2.14 Rows

```
LCD_ROWS_t LCD_t::Rows
```

Number of rows on the LCD.

6.30.2.15 RS_PIN

```
uint16 LCD_t::RS_PIN
```

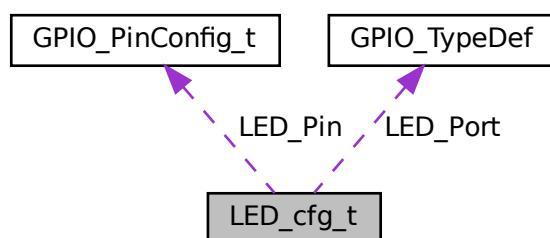
GPIO pin for Register Select (RS). (see [GPIO Pin Numbers](#))

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

- [bsp/Inc/lcd_driver.h](#)

6.31 LED_cfg_t Struct Reference

Collaboration diagram for LED_cfg_t:



Public Attributes

- `GPIO_TypeDef * LED_Port`
- `GPIO_PinConfig_t LED_Pin`
- `LED_Mode_t LED_Mode`

6.31.1 Member Data Documentation

6.31.1.1 LED_Mode

`LED_Mode_t LED_cfg_t::LED_Mode`

LED mode configuration.

6.31.1.2 LED_Pin

`GPIO_PinConfig_t LED_cfg_t::LED_Pin`

GPIO pin configuration for the LED.

6.31.1.3 LED_Port

`GPIO_TypeDef* LED_cfg_t::LED_Port`

GPIO port for the LED.

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

- `bsp/Inc/led_driver.h`

6.32 LTDC_Layer_TypeDef Struct Reference

LCD-TFT Display layer x Controller.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t WHPCR`
- `__IO uint32_t WVPCR`
- `__IO uint32_t CKCR`
- `__IO uint32_t PFCR`
- `__IO uint32_t CACR`
- `__IO uint32_t DCCR`
- `__IO uint32_t BFCR`
- `uint32_t RESERVED0 [2]`
- `__IO uint32_t CFBAR`
- `__IO uint32_t CFBLR`
- `__IO uint32_t CFBLNR`
- `uint32_t RESERVED1 [3]`
- `__IO uint32_t CLUTWR`

6.32.1 Detailed Description

LCD-TFT Display layer x Controller.

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

- CMSIS/Inc/stm32f4xx.h

6.33 LTDC_TypeDef Struct Reference

LCD-TFT Display Controller.

```
#include <stm32f4xx.h>
```

Public Attributes

- `uint32_t RESERVED0 [2]`
- `__IO uint32_t SSCR`
- `__IO uint32_t BPCR`
- `__IO uint32_t AWCR`
- `__IO uint32_t TWCR`
- `__IO uint32_t GCR`
- `uint32_t RESERVED1 [2]`
- `__IO uint32_t SRCR`
- `uint32_t RESERVED2 [1]`
- `__IO uint32_t BCCR`
- `uint32_t RESERVED3 [1]`
- `__IO uint32_t IER`
- `__IO uint32_t ISR`
- `__IO uint32_t ICR`
- `__IO uint32_t LIPCR`
- `__IO uint32_t CPSR`
- `__IO uint32_t CDSR`

6.33.1 Detailed Description

LCD-TFT Display Controller.

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

- CMSIS/Inc/stm32f4xx.h

6.34 NVIC_Type Struct Reference

Structure type to access the Nested Vectored Interrupt Controller (NVIC).

```
#include <core_cm4.h>
```

Public Attributes

- `__IO uint32_t ISER [8]`
- `uint32_t RESERVED0 [24]`
- `__IO uint32_t ICER [8]`
- `uint32_t RSERVED1 [24]`
- `__IO uint32_t ISPR [8]`
- `uint32_t RESERVED2 [24]`
- `__IO uint32_t ICPR [8]`
- `uint32_t RESERVED3 [24]`
- `__IO uint32_t IABR [8]`
- `uint32_t RESERVED4 [56]`
- `__IO uint8_t IP [240]`
- `uint32_t RESERVED5 [644]`
- `__O uint32_t STIR`

6.34.1 Detailed Description

Structure type to access the Nested Vectored Interrupt Controller (NVIC).

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

- CMSIS/Inc/core_cm4.h

6.35 NVIC_TypeDef Struct Reference

Public Attributes

- `vuint32_t ISER [3]`
- `uint32 RESERVED0 [29]`
- `vuint32_t ICER [3]`
- `uint32 RESERVED1 [29]`
- `vuint32_t ISPR [3]`
- `uint32 RESERVED2 [29]`
- `vuint32_t ICPR [3]`
- `uint32 RESERVED3 [29]`
- `vuint32_t IABR [3]`
- `uint32 RESERVED4 [61]`
- `vuint8_t IP [80]`
- `uint32 RESERVED5 [684]`
- `vuint32_t STIR`

6.35.1 Member Data Documentation

6.35.1.1 IABR

```
vuint32_t NVIC_TypeDef::IABR[3]
```

Interrupt Active Bit Registers

6.35.1.2 ICER

```
vuint32_t NVIC_TypeDef::ICER[3]
```

Interrupt Clear-Enable Registers

6.35.1.3 ICPR

```
vuint32_t NVIC_TypeDef::ICPR[3]
```

Interrupt Clear-Pending Registers

6.35.1.4 IP

```
vuint8_t NVIC_TypeDef::IP[80]
```

Interrupt Priority Registers

6.35.1.5 ISER

```
vuint32_t NVIC_TypeDef::ISER[3]
```

Interrupt Set-Enable Registers

6.35.1.6 ISPR

```
vuint32_t NVIC_TypeDef::ISPR[3]
```

Interrupt Set-Pending Registers

6.35.1.7 RESERVED0

```
uint32_t NVIC_TypeDef::RESERVED0[29]
```

Reserved

6.35.1.8 RESERVED1

```
uint32_t NVIC_TypeDef::RESERVED1[29]
```

Reserved

6.35.1.9 RESERVED2

```
uint32_t NVIC_TypeDef::RESERVED2[29]
```

Reserved

6.35.1.10 RESERVED3

```
uint32_t NVIC_TypeDef::RESERVED3[29]
```

Reserved

6.35.1.11 RESERVED4

```
uint32_t NVIC_TypeDef::RESERVED4[61]
```

Reserved

6.35.1.12 RESERVED5

```
vuint32_t NVIC_TypeDef::RESERVED5[684]
```

Reserved

6.35.1.13 STIR

```
vuint32_t NVIC_TypeDef::STIR
```

Software Trigger Interrupt Register

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

- drivers/Inc/stm32f401xx.h

6.36 PWR_TypeDef Struct Reference

Power Control.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t CSR`

6.36.1 Detailed Description

Power Control.

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

- CMSIS/Inc/stm32f4xx.h

6.37 RCC_TypeDef Struct Reference

Reset and Clock Control.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t PLLCFGR`
- `__IO uint32_t CFGR`
- `__IO uint32_t CIR`
- `__IO uint32_t AHB1RSTR`
- `__IO uint32_t AHB2RSTR`
- `__IO uint32_t AHB3RSTR`
- `uint32_t RESERVED0`
- `__IO uint32_t APB1RSTR`
- `__IO uint32_t APB2RSTR`
- `uint32_t RESERVED1 [2]`
- `__IO uint32_t AHB1ENR`
- `__IO uint32_t AHB2ENR`
- `__IO uint32_t AHB3ENR`
- `uint32_t RESERVED2`
- `__IO uint32_t APB1ENR`
- `__IO uint32_t APB2ENR`
- `uint32_t RESERVED3 [2]`
- `__IO uint32_t AHB1LPENR`
- `__IO uint32_t AHB2LPENR`
- `__IO uint32_t AHB3LPENR`
- `uint32_t RESERVED4`
- `__IO uint32_t APB1LPENR`
- `__IO uint32_t APB2LPENR`
- `uint32_t RESERVED5 [2]`
- `__IO uint32_t BDCR`
- `__IO uint32_t CSR`
- `uint32_t RESERVED6 [2]`
- `__IO uint32_t SSCGR`

- `__IO uint32_t PLLI2SCFGR`
- `__IO uint32_t PLLSAICFGR`
- `__IO uint32_t DCKCFGR`
- `__IO uint32_t CKGATENR`
- `__IO uint32_t DCKCFGR2`
- `vuint32_t CR`
- `vuint32_t PLLCFG`
- `vuint32_t CFG`
- `vuint32_t CIR`
- `vuint32_t AHB1RSTR`
- `vuint32_t AHB2RSTR`
- `uint32 RESERVED0 [2]`
- `vuint32_t APB1RSTR`
- `vuint32_t APB2RSTR`
- `uint32 RESERVED1 [2]`
- `vuint32_t AHB1ENR`
- `vuint32_t AHB2ENR`
- `uint32 RESERVED2 [2]`
- `vuint32_t APB1ENR`
- `vuint32_t APB2ENR`
- `uint32 RESERVED3 [2]`
- `vuint32_t AHB1LPENR`
- `vuint32_t AHB2LPENR`
- `uint32 RESERVED4 [2]`
- `vuint32_t APB1LPENR`
- `vuint32_t APB2LPENR`
- `uint32 RESERVED5 [2]`
- `vuint32_t BDCR`
- `vuint32_t CSR`
- `uint32 RESERVED6 [2]`
- `vuint32_t SSCGR`
- `vuint32_t PLLI2SCFGR`
- `uint32 RESERVED7`
- `vuint32_t DCKCFGR`

6.37.1 Detailed Description

Reset and Clock Control.

6.37.2 Member Data Documentation

6.37.2.1 AHB1ENR

`vuint32_t RCC_TypeDef::AHB1ENR`

AHB1 Peripheral Clock Enable Register

6.37.2.2 AHB1LPENR

```
vuint32_t RCC_TypeDef::AHB1LPENR
```

AHB1 Peripheral Low Power Enable Register

6.37.2.3 AHB1RSTR

```
vuint32_t RCC_TypeDef::AHB1RSTR
```

AHB1 Peripheral Reset Register

6.37.2.4 AHB2ENR

```
vuint32_t RCC_TypeDef::AHB2ENR
```

AHB2 Peripheral Clock Enable Register

6.37.2.5 AHB2LPENR

```
vuint32_t RCC_TypeDef::AHB2LPENR
```

AHB2 Peripheral Low Power Enable Register

6.37.2.6 AHB2RSTR

```
vuint32_t RCC_TypeDef::AHB2RSTR
```

AHB2 Peripheral Reset Register

6.37.2.7 APB1ENR

```
vuint32_t RCC_TypeDef::APB1ENR
```

APB1 Peripheral Clock Enable Register

6.37.2.8 APB1LPENR

```
vuint32_t RCC_TypeDef::APB1LPENR
```

APB1 Peripheral Low Power Enable Register

6.37.2.9 APB1RSTR

```
vuint32_t RCC_TypeDef::APB1RSTR
```

APB1 Peripheral Reset Register

6.37.2.10 APB2ENR

```
vuint32_t RCC_TypeDef::APB2ENR
```

APB2 Peripheral Clock Enable Register

6.37.2.11 APB2LPENR

```
vuint32_t RCC_TypeDef::APB2LPENR
```

APB2 Peripheral Low Power Enable Register

6.37.2.12 APB2RSTR

```
vuint32_t RCC_TypeDef::APB2RSTR
```

APB2 Peripheral Reset Register

6.37.2.13 BDCR

```
vuint32_t RCC_TypeDef::BDCR
```

Backup Domain Control Register

6.37.2.14 CFGR

```
vuint32_t RCC_TypeDef::CFGR
```

Clock Configuration Register

6.37.2.15 CIR

```
vuint32_t RCC_TypeDef::CIR
```

Clock Interrupt Register

6.37.2.16 CR

```
vuint32_t RCC_TypeDef::CR
```

Clock Control Register

6.37.2.17 CSR

```
vuint32_t RCC_TypeDef::CSR
```

Clock Control and Status Register

6.37.2.18 DCKCFGR

```
vuint32_t RCC_TypeDef::DCKCFGR
```

Dedicated Clocks Configuration Register

6.37.2.19 PLLCFGGR

```
vuint32_t RCC_TypeDef::PLLCFGR
```

PLL Configuration Register

6.37.2.20 PLLI2SCFGR

```
vuint32_t RCC_TypeDef::PLLI2SCFGR
```

PLLI2S Configuration Register

6.37.2.21 RESERVED0

```
uint32_t RCC_TypeDef::RESERVED0[2]
```

Reserved

6.37.2.22 RESERVED1

```
uint32_t RCC_TypeDef::RESERVED1[2]
```

Reserved

6.37.2.23 RESERVED2

```
uint32_t RCC_TypeDef::RESERVED2[2]
```

Reserved

6.37.2.24 RESERVED3

```
uint32_t RCC_TypeDef::RESERVED3[2]
```

Reserved

6.37.2.25 RESERVED4

```
uint32_t RCC_TypeDef::RESERVED4[2]
```

Reserved

6.37.2.26 RESERVED5

```
uint32_t RCC_TypeDef::RESERVED5[2]
```

Reserved

6.37.2.27 RESERVED6

```
uint32_t RCC_TypeDef::RESERVED6[2]
```

Reserved

6.37.2.28 RESERVED7

```
uint32_t RCC_TypeDef::RESERVED7
```

Reserved

6.37.2.29 SSCGR

```
vuint32_t RCC_TypeDef::SSCGR
```

Spread Spectrum Clock Generation Register

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

- CMSIS/Inc/[stm32f4xx.h](#)
- drivers/Inc/[stm32f401xx.h](#)

6.38 RNG_TypeDef Struct Reference

RNG.

```
#include <stm32f4xx.h>
```

Public Attributes

- [__IO uint32_t CR](#)
- [__IO uint32_t SR](#)
- [__IO uint32_t DR](#)

6.38.1 Detailed Description

RNG.

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

- CMSIS/Inc/[stm32f4xx.h](#)

6.39 RTC_TypeDef Struct Reference

Real-Time Clock.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t TR`
- `__IO uint32_t DR`
- `__IO uint32_t CR`
- `__IO uint32_t ISR`
- `__IO uint32_t PRER`
- `__IO uint32_t WUTR`
- `__IO uint32_t CALIBR`
- `__IO uint32_t ALRMAR`
- `__IO uint32_t ALRMBR`
- `__IO uint32_t WPR`
- `__IO uint32_t SSR`
- `__IO uint32_t SHIFTR`
- `__IO uint32_t TSTR`
- `__IO uint32_t TSDR`
- `__IO uint32_t TSSSR`
- `__IO uint32_t CALR`
- `__IO uint32_t TAFCR`
- `__IO uint32_t ALRMASSR`
- `__IO uint32_t ALRMBSSR`
- `uint32_t RESERVED7`
- `__IO uint32_t BKP0R`
- `__IO uint32_t BKP1R`
- `__IO uint32_t BKP2R`
- `__IO uint32_t BKP3R`
- `__IO uint32_t BKP4R`
- `__IO uint32_t BKP5R`
- `__IO uint32_t BKP6R`
- `__IO uint32_t BKP7R`
- `__IO uint32_t BKP8R`
- `__IO uint32_t BKP9R`
- `__IO uint32_t BKP10R`
- `__IO uint32_t BKP11R`
- `__IO uint32_t BKP12R`
- `__IO uint32_t BKP13R`
- `__IO uint32_t BKP14R`
- `__IO uint32_t BKP15R`
- `__IO uint32_t BKP16R`
- `__IO uint32_t BKP17R`
- `__IO uint32_t BKP18R`
- `__IO uint32_t BKP19R`

6.39.1 Detailed Description

Real-Time Clock.

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

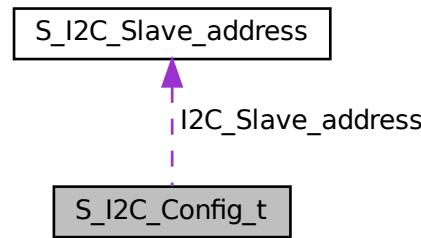
- CMSIS/Inc/stm32f4xx.h

6.40 S_I2C_Config_t Struct Reference

Structure for I2C configuration.

```
#include <stm32f401xx_i2c_driver.h>
```

Collaboration diagram for S_I2C_Config_t:



Public Attributes

- uint32_t I2C_Speed
- uint32_t I2C_stretchmode
- uint32_t I2C_Mode
- struct S_I2C_Slave_address I2C_Slave_address
- uint32_t I2C_Ack_Conrtol
- uint32_t General_Call_Address_Detection
- void(* P_Slave_Event_CallBack)(Slave_State state)

6.40.1 Detailed Description

Structure for I2C configuration.

6.40.2 Member Data Documentation

6.40.2.1 General_Call_Address_Detection

```
uint32_t S_I2C_Config_t::General_Call_Address_Detection
```

General call address detection.

6.40.2.2 I2C_Ack_Conrrol

```
uint32_t S_I2C_Config_t::I2C_Ack_Conrrol
```

Acknowledge control.

6.40.2.3 I2C_Mode

```
uint32_t S_I2C_Config_t::I2C_Mode
```

I2C or SMBus mode.

6.40.2.4 I2C_Slave_address

```
struct S_I2C_Slave_address S_I2C_Config_t::I2C_Slave_address
```

Slave address configuration.

6.40.2.5 I2C_Speed

```
uint32_t S_I2C_Config_t::I2C_Speed
```

I2C clock speed.

6.40.2.6 I2C_stretchmode

```
uint32_t S_I2C_Config_t::I2C_stretchmode
```

Clock stretching mode.

6.40.2.7 P_Slave_Event_CallBack

```
void(* S_I2C_Config_t::P_Slave_Event_CallBack) (Slave_State state)
```

Callback function for slave events.

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

- drivers/Inc/stm32f401xx_i2c_driver.h

6.41 S_I2C_Slave_address Struct Reference

Structure for I2C slave address configuration.

```
#include <stm32f401xx_i2c_driver.h>
```

Public Attributes

- uint16_t [Enable_Dual_Address](#)
- uint16_t [PrimaryAddress](#)
- uint16_t [SecondaryAddress](#)
- uint16_t [I2C_Addressing_Mode](#)

6.41.1 Detailed Description

Structure for I2C slave address configuration.

6.41.2 Member Data Documentation

6.41.2.1 Enable_Dual_Address

```
uint16_t S_I2C_Slave_address::Enable_Dual_Address
```

Enable or disable dual addressing mode.

6.41.2.2 I2C_Addressing_Mode

```
uint16_t S_I2C_Slave_address::I2C_Addressing_Mode
```

Addressing mode (7-bit or 10-bit).

6.41.2.3 PrimaryAddress

```
uint16_t S_I2C_Slave_address::PrimaryAddress
```

Primary 7-bit or 10-bit address of the I2C slave.

6.41.2.4 SecondaryAddress

```
uint16_t S_I2C_Slave_address::SecondaryAddress
```

Secondary 7-bit or 10-bit address of the I2C slave.

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

- drivers/Inc/[stm32f401xx_i2c_driver.h](#)

6.42 S_IRQ_SRC Struct Reference

Structure to identify the source of SPI interrupts.

```
#include <stm32f401xx_spi_driver.h>
```

Public Attributes

- uint8_t **TXE**:1
- uint8_t **RXE**:1
- uint8_t **ERRI**:1
- uint8_t **Reserved**:5

6.42.1 Detailed Description

Structure to identify the source of SPI interrupts.

This structure is used to indicate which interrupt event has occurred in the SPI peripheral.

6.42.2 Member Data Documentation

6.42.2.1 ERRI

```
uint8_t S_IRQ_SRC::ERRI
```

Error interrupt

6.42.2.2 Reserved

```
uint8_t S_IRQ_SRC::Reserved
```

Reserved bits

6.42.2.3 RXE

```
uint8_t S_IRQ_SRC::RXE
```

RX buffer is empty

6.42.2.4 TXE

```
uint8_t S_IRQ_SRC::TXE
```

TX buffer is empty

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

- drivers/Inc/stm32f401xx_spi_driver.h

6.43 S_SPI_Config_t Struct Reference

Configuration structure for SPI.

```
#include <stm32f401xx_spi_driver.h>
```

Public Attributes

- uint16_t [Device_Mode](#)
- uint16_t [Communication_Mode](#)
- uint16_t [Payload_Length](#)
- uint16_t [Frame_Format](#)
- uint16_t [Clock_Polarity](#)
- uint16_t [Clock_Phase](#)
- uint16_t [NSS](#)
- uint16_t [SPI_Prescaler](#)
- uint16_t [IRQ_Enable](#)
- void(* [P IRQ_CallBack](#))(struct [S_IRQ_SRC](#) IRQ)

6.43.1 Detailed Description

Configuration structure for SPI.

This structure is used to configure the SPI peripheral. It contains settings for device mode, communication mode, payload length, frame format, clock polarity, clock phase, NSS configuration, SPI prescaler, and interrupt enable.

6.43.2 Member Data Documentation

6.43.2.1 Clock_Phase

```
uint16_t S_SPI_Config_t::Clock_Phase
```

Specifies clock phase (sampling on leading or trailing edge). [SPI Clock Phase Definitions](#)

6.43.2.2 Clock_Polarity

```
uint16_t S_SPI_Config_t::Clock_Polarity
```

Specifies clock polarity (idle low or high). [SPI Clock Polarity Definitions](#)

6.43.2.3 Communication_Mode

```
uint16_t S_SPI_Config_t::Communication_Mode
```

Specifies communication mode (bidirectional or unidirectional). [Communication_Mode_Define](#)

6.43.2.4 Device_Mode

```
uint16_t S_SPI_Config_t::Device_Mode
```

Specifies if SPI operates in master mode or slave mode. [Device_Mode_Define](#)

6.43.2.5 Frame_Format

```
uint16_t S_SPI_Config_t::Frame_Format
```

Specifies frame format (MSB or LSB first). [SPI Frame Format Definitions](#)

6.43.2.6 IRQ_Enable

```
uint16_t S_SPI_Config_t::IRQ_Enable
```

Enables or disables SPI interrupts. [SPI IRQ Enable Definitions](#)

6.43.2.7 NSS

```
uint16_t S_SPI_Config_t::NSS
```

Specifies NSS management (hardware or software). [SPI NSS Definitions](#)

6.43.2.8 P_IRQ_CallBack

```
void(* S_SPI_Config_t::P_IRQ_CallBack) (struct S_IRQ_SRC IRQ)
```

Callback function for interrupt handling

6.43.2.9 Payload_Length

```
uint16_t S_SPI_Config_t::Payload_Length
```

Specifies payload length (8 or 16 bits). [SPI Payload Length Definitions](#)

6.43.2.10 SPI_Prescaler

```
uint16_t S_SPI_Config_t::SPI_Prescaler
```

Specifies SPI frequency. [SPI Prescaler Definitions](#)

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

- drivers/Inc/[stm32f401xx_spi_driver.h](#)

6.44 SAI_Block_TypeDef Struct Reference

Public Attributes

- [__IO uint32_t CR1](#)
- [__IO uint32_t CR2](#)
- [__IO uint32_t FRCR](#)
- [__IO uint32_t SLOTR](#)
- [__IO uint32_t IMR](#)
- [__IO uint32_t SR](#)
- [__IO uint32_t CLRFR](#)
- [__IO uint32_t DR](#)

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

- CMSIS/Inc/[stm32f4xx.h](#)

6.45 SAI_TypeDef Struct Reference

Serial Audio Interface.

```
#include <stm32f4xx.h>
```

Public Attributes

- [__IO uint32_t GCR](#)

6.45.1 Detailed Description

Serial Audio Interface.

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

- CMSIS/Inc/[stm32f4xx.h](#)

6.46 SCB_Type Struct Reference

Structure type to access the System Control Block (SCB).

```
#include <core_cm4.h>
```

Public Attributes

- `__I uint32_t CPUID`
- `__IO uint32_t ICSR`
- `__IO uint32_t VTOR`
- `__IO uint32_t AIRCR`
- `__IO uint32_t SCR`
- `__IO uint32_t CCR`
- `__IO uint8_t SHP [12]`
- `__IO uint32_t SHCSR`
- `__IO uint32_t CFSR`
- `__IO uint32_t HFSR`
- `__IO uint32_t DFSR`
- `__IO uint32_t MMFAR`
- `__IO uint32_t BFAR`
- `__IO uint32_t AFSR`
- `__I uint32_t PFR [2]`
- `__I uint32_t DFR`
- `__I uint32_t ADR`
- `__I uint32_t MMFR [4]`
- `__I uint32_t ISAR [5]`
- `uint32_t RESERVED0 [5]`
- `__IO uint32_t CPACR`

6.46.1 Detailed Description

Structure type to access the System Control Block (SCB).

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

- CMSIS/Inc/core_cm4.h

6.47 SCB_TypeDef Struct Reference

Public Attributes

- `vuint32_t CPUID`
- `vuint32_t ICSR`
- `vuint32_t VTOR`
- `vuint32_t AIRCR`
- `vuint32_t SCR`
- `vuint32_t CCR`
- `vuint8_t SHP [12]`
- `vuint32_t SHCSR`
- `vuint32_t CFSR`
- `vuint32_t HFSR`
- `uint32 RESERVED`
- `vuint32_t MMAR`
- `vuint32_t BFAR`

6.47.1 Member Data Documentation

6.47.1.1 AIRCR

```
vuint32_t SCB_TypeDef::AIRCR
```

Application Interrupt and Reset Control Register

6.47.1.2 BFAR

```
vuint32_t SCB_TypeDef::BFAR
```

Bus Fault Address Register

6.47.1.3 CCR

```
vuint32_t SCB_TypeDef::CCR
```

Configuration and Control Register

6.47.1.4 CFSR

```
vuint32_t SCB_TypeDef::CFSR
```

Configurable Fault Status Register

6.47.1.5 CPUID

```
vuint32_t SCB_TypeDef::CPUID
```

CPU ID Base Register

6.47.1.6 HFSR

```
vuint32_t SCB_TypeDef::HFSR
```

Hard Fault Status Register

6.47.1.7 ICSR

```
vuint32_t SCB_TypeDef::ICSR
```

Interrupt Control and State Register

6.47.1.8 MMAR

`vuint32_t` SCB_TypeDef::MMAR

Memory Management Fault Address Register

6.47.1.9 RESERVED

`uint32` SCB_TypeDef::RESERVED

Reserved

6.47.1.10 SCR

`vuint32_t` SCB_TypeDef::SCR

System Control Register

6.47.1.11 SHCSR

`vuint32_t` SCB_TypeDef::SHCSR

System Handler Control and State Register

6.47.1.12 SHP

`vuint8_t` SCB_TypeDef::SHP[12]

System Handler Priority Registers

6.47.1.13 VTOR

`vuint32_t` SCB_TypeDef::VTOR

Vector Table Offset Register

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

- drivers/Inc/[stm32f401xx.h](#)

6.48 SCnSCB_Type Struct Reference

Structure type to access the System Control and ID Register not in the SCB.

```
#include <core_cm4.h>
```

Public Attributes

- `uint32_t RESERVED0 [1]`
- `__I uint32_t ICTR`
- `__IO uint32_t ACTLR`

6.48.1 Detailed Description

Structure type to access the System Control and ID Register not in the SCB.

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

- CMSIS/Inc/[core_cm4.h](#)

6.49 SDIO_TypeDef Struct Reference

SD host Interface.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t POWER`
- `__IO uint32_t CLKCR`
- `__IO uint32_t ARG`
- `__IO uint32_t CMD`
- `__I uint32_t RESPCMD`
- `__I uint32_t RESP1`
- `__I uint32_t RESP2`
- `__I uint32_t RESP3`
- `__I uint32_t RESP4`
- `__IO uint32_t DTIMER`
- `__IO uint32_t DLEN`
- `__IO uint32_t DCTRL`
- `__I uint32_t DCOUNT`
- `__I uint32_t STA`
- `__IO uint32_t ICR`
- `__IO uint32_t MASK`
- `uint32_t RESERVED0 [2]`
- `__I uint32_t FIFO_CNT`
- `uint32_t RESERVED1 [13]`
- `__IO uint32_t FIFO`

6.49.1 Detailed Description

SD host Interface.

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

- CMSIS/Inc/[stm32f4xx.h](#)

6.50 SPI_TypeDef Struct Reference

Serial Peripheral Interface.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint16_t CR1`
- `uint16_t RESERVED0`
- `__IO uint16_t CR2`
- `uint16_t RESERVED1`
- `__IO uint16_t SR`
- `uint16_t RESERVED2`
- `__IO uint16_t DR`
- `uint16_t RESERVED3`
- `__IO uint16_t CRCPR`
- `uint16_t RESERVED4`
- `__IO uint16_t RXCRR`
- `uint16_t RESERVED5`
- `__IO uint16_t TXCRR`
- `uint16_t RESERVED6`
- `__IO uint16_t I2SCFGR`
- `uint16_t RESERVED7`
- `__IO uint16_t I2SPR`
- `uint16_t RESERVED8`
- `uint32 CR1`
- `uint32 CR2`
- `uint32 SR`
- `uint32 DR`
- `uint32 CRCPR`
- `uint32 RXCRR`
- `uint32 TXCRR`
- `uint32 I2SCFGR`
- `uint32 I2SPR`

6.50.1 Detailed Description

Serial Peripheral Interface.

6.50.2 Member Data Documentation

6.50.2.1 CR1

```
uint32 SPI_TypeDef::CR1
```

SPI Control Register 1

6.50.2.2 CR2

```
uint32 SPI_TypeDef::CR2
```

SPI Control Register 1

6.50.2.3 CRCPR

```
uint32 SPI_TypeDef::CRCPR
```

SPI CRC Polynomial Register

6.50.2.4 DR

```
uint32 SPI_TypeDef::DR
```

SPI Data Register

6.50.2.5 I2SCFGR

```
uint32 SPI_TypeDef::I2SCFGR
```

SPI I2S Configuration Register

6.50.2.6 I2SPR

```
uint32 SPI_TypeDef::I2SPR
```

SPI I2S Prescaler Register

6.50.2.7 RXCRCR

```
uint32 SPI_TypeDef::RXCRCR
```

SPI RX CRC Register

6.50.2.8 SR

```
uint32 SPI_TypeDef::SR
```

SPI Status Register

6.50.2.9 TXCRCR

```
uint32 SPI_TypeDef::TXCRCR
```

SPI TX CRC Register

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

- CMSIS/Inc/[stm32f4xx.h](#)
- drivers/Inc/[stm32f401xx.h](#)

6.51 STK_config_t Struct Reference

SysTick Configuration Structure Definition.

```
#include <stm32f401xx_systick_driver.h>
```

Public Attributes

- `uint8 running_mode`
- `uint8 clock_config`
- `uint8 interrupt_config`
- `uint32 reload_value`
- `void(* Callback_Function)(void)`

6.51.1 Detailed Description

SysTick Configuration Structure Definition.

This structure contains the configuration options for the SysTick timer, including the running mode, clock source, interrupt configuration, and reload value. A callback function can also be specified to handle the SysTick interrupt.

6.51.2 Member Data Documentation

6.51.2.1 Callback_Function

```
void(* STK_config_t::Callback_Function) (void)
```

Pointer to the callback function for SysTick interrupt

6.51.2.2 clock_config

```
uint8 STK_config_t::clock_config
```

[SysTick Clock Configuration](#) - Clock source for the SysTick timer

6.51.2.3 interrupt_config

```
uint8 STK_config_t::interrupt_config
```

SysTick Interrupt Configuration - Interrupt enable/disable configuration

6.51.2.4 reload_value

```
uint32 STK_config_t::reload_value
```

Reload value to define the SysTick timer's duration

6.51.2.5 running_mode

```
uint8 STK_config_t::running_mode
```

SysTick Running Mode Configuration - Running mode of the SysTick timer

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

- drivers/Inc/stm32f401xx_systick_driver.h

6.52 STK_TypeDef Struct Reference

Public Attributes

- `vuint32_t CTRL`
- `vuint32_t LOAD`
- `vuint32_t VAL`
- `vuint32_t CALIB`

6.52.1 Member Data Documentation

6.52.1.1 CALIB

```
vuint32_t STK_TypeDef::CALIB
```

Calibration Value Register

6.52.1.2 CTRL

```
vuint32_t STK_TypeDef::CTRL
```

Control and Status Register

6.52.1.3 LOAD

`vuint32_t` `STK_TypeDef::LOAD`

Reload Value Register

6.52.1.4 VAL

`vuint32_t` `STK_TypeDef::VAL`

Current Value Register

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

- `drivers/Inc/stm32f401xx.h`

6.53 SYSCFG_RegDef_t Struct Reference

Public Attributes

- `vuint32_t MEMRMP`
- `vuint32_t PMC`
- `vuint32_t EXTICR [4]`
- `uint32 RESERVED1 [2]`
- `vuint32_t CMPCR`
- `uint32 RESERVED2 [2]`
- `vuint32_t CFGR`

6.53.1 Member Data Documentation

6.53.1.1 CFGR

`vuint32_t` `SYSCFG_RegDef_t::CFGR`

SYSCFG Configuration Register

6.53.1.2 CMPCR

`vuint32_t` `SYSCFG_RegDef_t::CMPCR`

SYSCFG Compensation Cell Control Register

6.53.1.3 EXTICR

```
vuint32_t SYSCFG_RegDef_t::EXTICR[4]
```

SYSCFG External Interrupt Configuration Registers

6.53.1.4 MEMRMP

```
vuint32_t SYSCFG_RegDef_t::MEMRMP
```

SYSCFG Memory Remap Register

6.53.1.5 PMC

```
vuint32_t SYSCFG_RegDef_t::PMC
```

SYSCFG Peripheral Mode Configuration Register

6.53.1.6 RESERVED1

```
uint32_t SYSCFG_RegDef_t::RESERVED1[2]
```

Reserved

6.53.1.7 RESERVED2

```
uint32_t SYSCFG_RegDef_t::RESERVED2[2]
```

Reserved

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

- drivers/Inc/stm32f401xx.h

6.54 SYSCFG_TypeDef Struct Reference

System configuration controller.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t MEMRMP`
- `__IO uint32_t PMC`
- `__IO uint32_t EXTICR [4]`
- `uint32_t RESERVED [2]`
- `__IO uint32_t CMPCR`

6.54.1 Detailed Description

System configuration controller.

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

- CMSIS/Inc/[stm32f4xx.h](#)

6.55 SysTick_Type Struct Reference

Structure type to access the System Timer (SysTick).

```
#include <core_cm4.h>
```

Public Attributes

- [__IO uint32_t CTRL](#)
- [__IO uint32_t LOAD](#)
- [__IO uint32_t VAL](#)
- [__I uint32_t CALIB](#)

6.55.1 Detailed Description

Structure type to access the System Timer (SysTick).

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

- CMSIS/Inc/[core_cm4.h](#)

6.56 TIM1_TypeDef Struct Reference

Public Attributes

- [uint32 CR1](#)
- [uint32 CR2](#)
- [uint32 SMCR](#)
- [uint32 DIER](#)
- [uint32 SR](#)
- [uint32 EGR](#)
- [uint32 CCMR1](#)
- [uint32 CCMR2](#)
- [uint32 CCER](#)
- [uint32 CNT](#)
- [uint32 PSC](#)
- [uint32 ARR](#)
- [uint32 RCR](#)
- [uint32 CCR1](#)
- [uint32 CCR2](#)
- [uint32 CCR3](#)
- [uint32 CCR4](#)
- [uint32 BDTR](#)
- [uint32 DCR](#)
- [uint32 DMAR](#)
- [uint32 OR](#)

6.56.1 Member Data Documentation

6.56.1.1 ARR

```
uint32 TIM1_TypeDef::ARR
```

TIM1 Auto-Reload Register

6.56.1.2 BDTR

```
uint32 TIM1_TypeDef::BDTR
```

TIM1 Break and Dead-Time Register

6.56.1.3 CCER

```
uint32 TIM1_TypeDef::CCER
```

TIM1 Capture/Compare Enable Register

6.56.1.4 CCMR1

```
uint32 TIM1_TypeDef::CCMR1
```

TIM1 Capture/Compare Mode Register 1

6.56.1.5 CCMR2

```
uint32 TIM1_TypeDef::CCMR2
```

TIM1 Capture/Compare Mode Register 2

6.56.1.6 CCR1

```
uint32 TIM1_TypeDef::CCR1
```

TIM1 Capture/Compare Register 1

6.56.1.7 CCR2

```
uint32 TIM1_TypeDef::CCR2
```

TIM1 Capture/Compare Register 2

6.56.1.8 CCR3

```
uint32 TIM1_TypeDef::CCR3
```

TIM1 Capture/Compare Register 3

6.56.1.9 CCR4

```
uint32 TIM1_TypeDef::CCR4
```

TIM1 Capture/Compare Register 4

6.56.1.10 CNT

```
uint32 TIM1_TypeDef::CNT
```

TIM1 Counter Register

6.56.1.11 CR1

```
uint32 TIM1_TypeDef::CR1
```

TIM1 Control Register 1

6.56.1.12 CR2

```
uint32 TIM1_TypeDef::CR2
```

TIM1 Control Register 2

6.56.1.13 DCR

```
uint32 TIM1_TypeDef::DCR
```

TIM1 DMA Control Register

6.56.1.14 DIER

```
uint32 TIM1_TypeDef::DIER
```

TIM1 DMA/Interrupt Enable Register

6.56.1.15 DMAR

```
uint32 TIM1_TypeDef::DMAR
```

TIM1 DMA Address Register

6.56.1.16 EGR

```
uint32 TIM1_TypeDef::EGR
```

TIM1 Event Generation Register

6.56.1.17 OR

```
uint32 TIM1_TypeDef::OR
```

TIM1 Option Register

6.56.1.18 PSC

```
uint32 TIM1_TypeDef::PSC
```

TIM1 Prescaler Register

6.56.1.19 RCR

```
uint32 TIM1_TypeDef::RCR
```

TIM1 Repetition Counter Register

6.56.1.20 SMCR

```
uint32 TIM1_TypeDef::SMCR
```

TIM1 Slave Mode Control Register

6.56.1.21 SR

```
uint32 TIM1_TypeDef::SR
```

TIM1 Status Register

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

- drivers/Inc/stm32f401xx.h

6.57 TIM_TypeDef Struct Reference

TIM.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint16_t CR1`
- `uint16_t RESERVED0`
- `__IO uint16_t CR2`
- `uint16_t RESERVED1`
- `__IO uint16_t SMCR`
- `uint16_t RESERVED2`
- `__IO uint16_t DIER`
- `uint16_t RESERVED3`
- `__IO uint16_t SR`
- `uint16_t RESERVED4`
- `__IO uint16_t EGR`
- `uint16_t RESERVED5`
- `__IO uint16_t CCMR1`
- `uint16_t RESERVED6`
- `__IO uint16_t CCMR2`
- `uint16_t RESERVED7`
- `__IO uint16_t CCER`
- `uint16_t RESERVED8`
- `__IO uint32_t CNT`
- `__IO uint16_t PSC`
- `uint16_t RESERVED9`
- `__IO uint32_t ARR`
- `__IO uint16_t RCR`
- `uint16_t RESERVED10`
- `__IO uint32_t CCR1`
- `__IO uint32_t CCR2`
- `__IO uint32_t CCR3`
- `__IO uint32_t CCR4`
- `__IO uint16_t BDTR`
- `uint16_t RESERVED11`
- `__IO uint16_t DCR`
- `uint16_t RESERVED12`
- `__IO uint16_t DMAR`
- `uint16_t RESERVED13`
- `__IO uint16_t OR`
- `uint16_t RESERVED14`

6.57.1 Detailed Description

TIM.

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

- CMSIS/Inc/stm32f4xx.h

6.58 TPI_Type Struct Reference

Structure type to access the Trace Port Interface Register (TPI).

```
#include <core_cm4.h>
```

Public Attributes

- `__IO uint32_t SSPSR`
- `__IO uint32_t CSPSR`
- `uint32_t RESERVED0 [2]`
- `__IO uint32_t ACPR`
- `uint32_t RESERVED1 [55]`
- `__IO uint32_t SPPR`
- `uint32_t RESERVED2 [131]`
- `__I uint32_t FFSR`
- `__IO uint32_t FFCR`
- `__I uint32_t FSCR`
- `uint32_t RESERVED3 [759]`
- `__I uint32_t TRIGGER`
- `__I uint32_t FIFO0`
- `__I uint32_t ITATBCTR2`
- `uint32_t RESERVED4 [1]`
- `__I uint32_t ITATBCTR0`
- `__I uint32_t FIFO1`
- `__IO uint32_t ITCTRL`
- `uint32_t RESERVED5 [39]`
- `__IO uint32_t CLAIMSET`
- `__IO uint32_t CLAIMCLR`
- `uint32_t RESERVED7 [8]`
- `__I uint32_t DEVID`
- `__I uint32_t DEVTYPE`

6.58.1 Detailed Description

Structure type to access the Trace Port Interface Register (TPI).

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

- CMSIS/Inc/core_cm4.h

6.59 USART_cfg_t Struct Reference

Configuration structure for USART (Universal Synchronous Asynchronous Receiver Transmitter) peripheral.

```
#include <stm32f401xx_usart_driver.h>
```

Public Attributes

- `uint8 USART_Mode`
- `uint32 BaudRate`
- `uint8 Payload_Length`
- `uint32 Parity`
- `uint32 StopBits`
- `uint8 HwFlowCtl`
- `uint8 IRQ_Enable`
- `void(* P_IRQ_Callback)(void)`

6.59.1 Detailed Description

Configuration structure for USART (Universal Synchronous Asynchronous Receiver Transmitter) peripheral.

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

- drivers/Inc/stm32f401xx_usart_driver.h

6.60 USART_TypeDef Struct Reference

Universal Synchronous Asynchronous Receiver Transmitter.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint16_t SR`
- `uint16_t RESERVED0`
- `__IO uint16_t DR`
- `uint16_t RESERVED1`
- `__IO uint16_t BRR`
- `uint16_t RESERVED2`
- `__IO uint16_t CR1`
- `uint16_t RESERVED3`
- `__IO uint16_t CR2`
- `uint16_t RESERVED4`
- `__IO uint16_t CR3`
- `uint16_t RESERVED5`
- `__IO uint16_t GTPR`
- `uint16_t RESERVED6`
- `uint32 SR`
- `uint32 DR`
- `uint32 BRR`
- `uint32 CR1`
- `uint32 CR2`
- `uint32 CR3`
- `uint32 GTPR`

6.60.1 Detailed Description

Universal Synchronous Asynchronous Receiver Transmitter.

6.60.2 Member Data Documentation

6.60.2.1 BRR

```
uint32 USART_TypeDef::BRR
```

USART Baud Rate Register

6.60.2.2 CR1

```
uint32 USART_TypeDef::CR1
```

USART Control Register 1

6.60.2.3 CR2

```
uint32 USART_TypeDef::CR2
```

USART Control Register 2

6.60.2.4 CR3

```
uint32 USART_TypeDef::CR3
```

USART Control Register 3

6.60.2.5 DR

```
uint32 USART_TypeDef::DR
```

USART Data Register

6.60.2.6 GTPR

```
uint32 USART_TypeDef::GTPR
```

USART Guard Time and Prescaler Register

6.60.2.7 SR

```
uint32 USART_TypeDef::SR
```

USART Status Register

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

- CMSIS/Inc/[stm32f4xx.h](#)
- drivers/Inc/[stm32f401xx.h](#)

6.61 WWDG_TypeDef Struct Reference

Window WATCHDOG.

```
#include <stm32f4xx.h>
```

Public Attributes

- `__IO uint32_t CR`
- `__IO uint32_t CFR`
- `__IO uint32_t SR`

6.61.1 Detailed Description

Window WATCHDOG.

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

- CMSIS/Inc/stm32f4xx.h

6.62 xPSR_Type Union Reference

Union type to access the Special-Purpose Program Status Registers (xPSR).

```
#include <core_cm4.h>
```

Public Attributes

- struct {
 `uint32_t xPSR_Type::ISR:9`
 `uint32_t xPSR_Type::_reserved0:7`
 `uint32_t xPSR_Type::GE:4`
 `uint32_t xPSR_Type::_reserved1:4`
 `uint32_t xPSR_Type::T:1`
 `uint32_t xPSR_Type::IT:2`
 `uint32_t xPSR_Type::Q:1`
 `uint32_t xPSR_Type::V:1`
 `uint32_t xPSR_Type::C:1`
 `uint32_t xPSR_Type::Z:1`
 `uint32_t xPSR_Type::N:1`
} **b**
- `uint32_t w`

6.62.1 Detailed Description

Union type to access the Special-Purpose Program Status Registers (xPSR).

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

- CMSIS/Inc/core_cm4.h

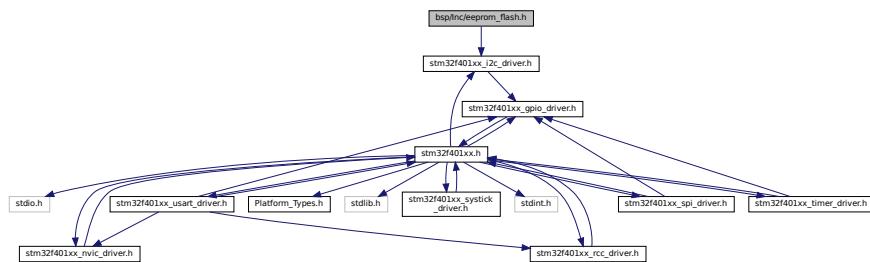
Chapter 7

File Documentation

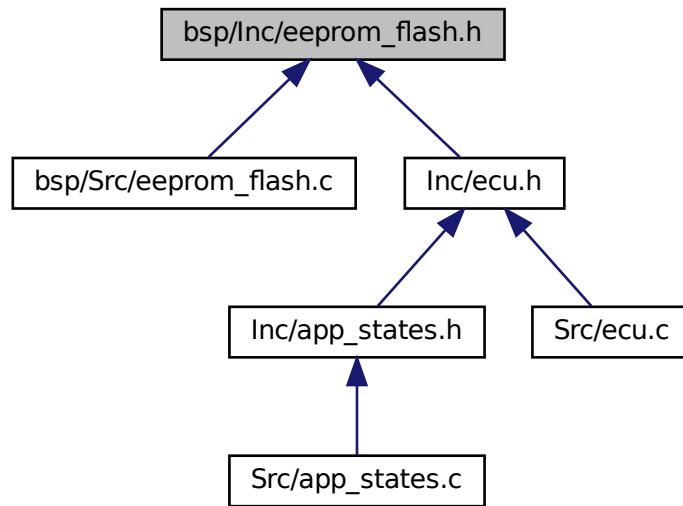
7.1 bsp/Inc/eeprom_flash.h File Reference

Header file for EEPROM flash memory operations.

```
#include "stm32f401xx_i2c_driver.h"  
Include dependency graph for eeprom_flash.h:
```



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



Macros

- `#define EEPROM_Slave_address 0x50`
AT24C1024B EEPROM slave address for I2C communication.

Functions

- `void EEPROM_Init (void)`
Initializes the EEPROM for communication.
- `unsigned char EEPROM_Write_NBytes (unsigned int Memory_address, unsigned char *bytes, uint8_t Data_Length)`
Writes multiple bytes to the EEPROM.
- `unsigned char EEPROM_Read_byte (unsigned int Memory_address, uint8_t *dataout, uint8_t datalen)`
Reads a byte from the EEPROM.

7.1.1 Detailed Description

Header file for EEPROM flash memory operations.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file provides the function prototypes and constants for initializing and managing EEPROM flash memory using the STM32F401xx MCU. It includes functions for writing and reading data from the EEPROM and defines the EEPROM slave address used in I2C communication.

Copyright

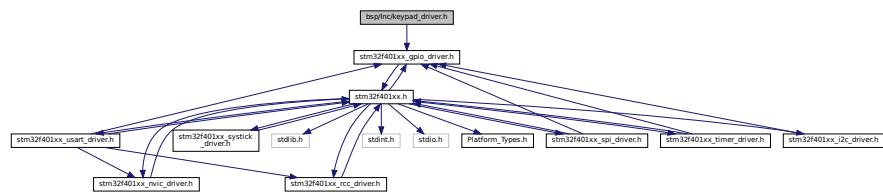
Copyright (c) 2024 Mohamed Ali Haoufa. All rights reserved.

The EEPROM is configured as an I2C slave device with the address 0x50. The provided functions allow for initialization, writing, and reading data from the AT24C1024B EEPROM. Ensure the correct I2C driver is included and configured for communication with the EEPROM.

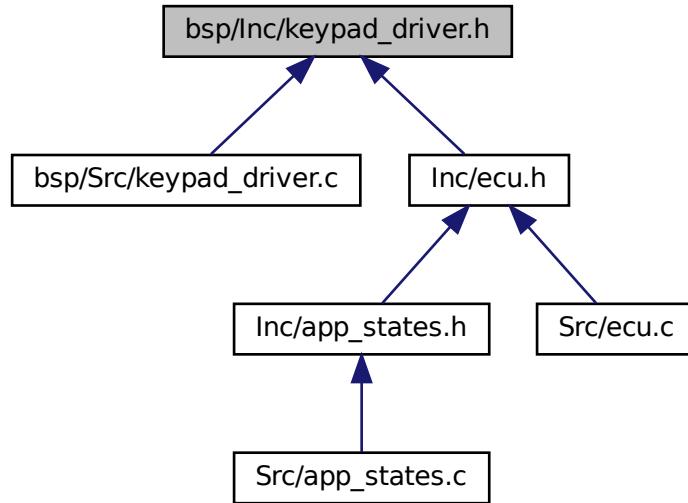
7.2 bsp/Inc/keypad_driver.h File Reference

Header file for the STM32F401xx microcontroller keypad driver.

```
#include "stm32f401xx_gpio_driver.h"  
Include dependency graph for keypad_driver.h:
```



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



Macros

- `#define KEYPAD_PORT GPIOE`
Defines the GPIO port and pins for the keypad.
- `#define KEYPAD_ROWS 4`
- `#define ROW0 GPIO_PIN_0`
- `#define ROW1 GPIO_PIN_1`
- `#define ROW2 GPIO_PIN_3`
- `#define ROW3 GPIO_PIN_4`
- `#define KEYPAD_COLS 3`
- `#define COL0 GPIO_PIN_5`
- `#define COL1 GPIO_PIN_6`
- `#define COL2 GPIO_PIN_7`

Functions

- `void keypad_init (void)`
Initializes the keypad.
- `uint8 keypad_Get_Pressed_Key (void)`
Checks for a pressed key and returns its value.

7.2.1 Detailed Description

Header file for the STM32F401xx microcontroller keypad driver.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This header file provides the necessary definitions, macros, and function prototypes for interfacing with a keypad connected to the STM32F401xx microcontroller. It includes definitions for keypad rows and columns, as well as APIs for initializing the keypad and reading the pressed key.

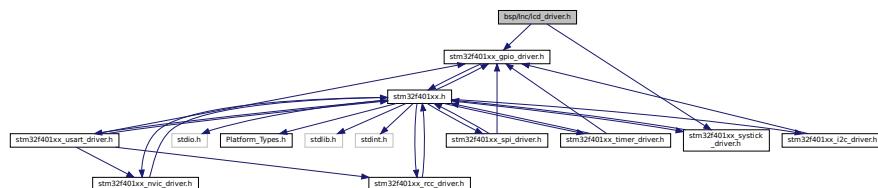
Copyright

Copyright (c) 2024 Mohamed Ali Haoufa. All rights reserved.

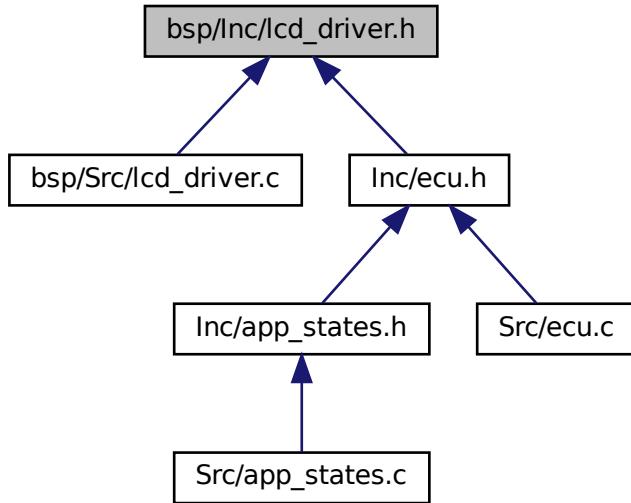
7.3 bsp/Inc/lcd_driver.h File Reference

Header file for the STM32F401xx microcontroller LCD driver.

```
#include "stm32f401xx_gpio_driver.h"
#include "stm32f401xx_systick_driver.h"
Include dependency graph for lcd_driver.h:
```



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



Classes

- struct [LCD_t](#)

Structure for configuring the LCD.

Macros

- #define [LCD_CLEAR_DISPLAY](#) (0x01)
Command to clear the LCD display.
- #define [LCD_RETURN_HOME](#) (0x02)
Command to return the cursor to the home position.
- #define [LCD_ENTRY_MODE_DEC_SHIFT_OFF](#) (0x04)
Command to set the entry mode to decrement and shift off.
- #define [LCD_ENTRY_MODE_DEC_SHIFT_ON](#) (0x05)
Command to set the entry mode to decrement and shift on.
- #define [LCD_ENTRY_MODE_INC_SHIFT_OFF](#) (0x06)
Command to set the entry mode to increment and shift off.
- #define [LCD_ENTRY_MODE_INC_SHIFT_ON](#) (0x07)
Command to set the entry mode to increment and shift on.
- #define [LCD_CURSOR_MOVE_SHIFT_LEFT](#) (0x10)
Command to move the cursor left.
- #define [LCD_CURSOR_MOVE_SHIFT_RIGHT](#) (0x14)
Command to move the cursor right.
- #define [LCD_DISPLAY_SHIFT_LEFT](#) (0x18)
Command to shift the display left.
- #define [LCD_DISPLAY_SHIFT_RIGHT](#) (0x1C)

- `#define LCD_DISPLAY_ON_UNDERLINE_OFF_CURSOR_OFF (0x0C)`

Command to shift the display right.
- `#define LCD_DISPLAY_ON_UNDERLINE_OFF_CURSOR_ON (0x0D)`

Command to turn on the display with underline off and cursor off.
- `#define LCD_DISPLAY_ON_UNDERLINE_ON_CURSOR_OFF (0x0E)`

Command to turn on the display with underline off and cursor on.
- `#define LCD_DISPLAY_ON_UNDERLINE_ON_CURSOR_ON (0x0F)`

Command to turn on the display with underline on and cursor off.
- `#define LCD_DISPLAY_OFF_CURSOR_OFF (0x08)`

Command to turn off the display with cursor off.
- `#define LCD_8BIT_MODE_2_LINE (0x38)`

Command to set the display in 8-bit mode with 2 lines.
- `#define LCD_4BIT_MODE_2_LINE (0x28)`

Command to set the display in 4-bit mode with 2 lines.
- `#define LCD_FIRST_ROW (0x80)`

Command to set the cursor position to the first row.
- `#define LCD_SECOND_ROW (0xC0)`

Command to set the cursor position to the second row.
- `#define LCD_THIRD_ROW (0x94)`

Command to set the cursor position to the third row.
- `#define LCD_FOURTH_ROW (0xD4)`

Command to set the cursor position to the fourth row.

Enumerations

- enum `LCD_MODE_t { LCD_8BIT , LCD_4BIT }`

Enum for specifying the LCD mode.
- enum `LCD_ROWS_t { LCD_2ROWS , LCD_4ROWS }`

Enum for specifying the number of rows on the LCD.

Functions

- void `LCD_Init (LCD_t *LCD_cfg)`

Initializes the LCD with user-defined configurations.
- void `LCD_Send_Command (LCD_t *LCD_cfg, uint8 command)`

Sends a command to the LCD to be executed.
- void `LCD_Send_Char (LCD_t *LCD_cfg, uint8 Char)`

Sends a character to the LCD to be displayed.
- void `LCD_Send_Char_Pos (LCD_t *LCD_cfg, uint8 Char, uint8 row, uint8 column)`

Sends a character to the LCD to be displayed at a specific location.
- void `LCD_Send_String (LCD_t *LCD_cfg, uint8 *string)`

Sends a string to the LCD to be displayed.
- void `LCD_Send_String_Pos (LCD_t *LCD_cfg, uint8 *string, uint8 row, uint8 column)`

Sends a string to the LCD to be displayed at a specific location.
- void `LCD_Send_Enable_Signal (LCD_t *LCD_cfg)`

Sends an enable signal to the LCD.
- void `LCD_Set_Cursor (LCD_t *LCD_cfg, uint8 row, uint8 column)`

Sets the location of the cursor on the LCD.

7.3.1 Detailed Description

Header file for the STM32F401xx microcontroller LCD driver.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This header file provides the necessary definitions, macros, and function prototypes for interfacing with an LCD connected to the STM32F401xx microcontroller. It includes configurations for LCD modes, rows, and various LCD commands and functions.

Copyright

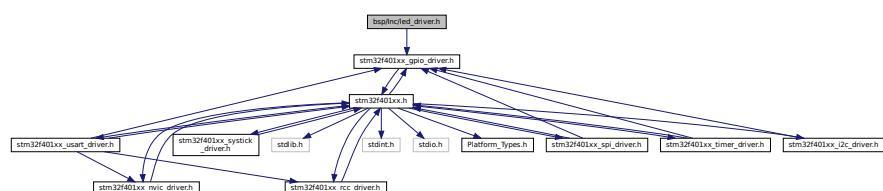
Copyright (c) 2024 Mohamed Ali Haoufa. All rights reserved.

7.4 bsp/Inc/led_driver.h File Reference

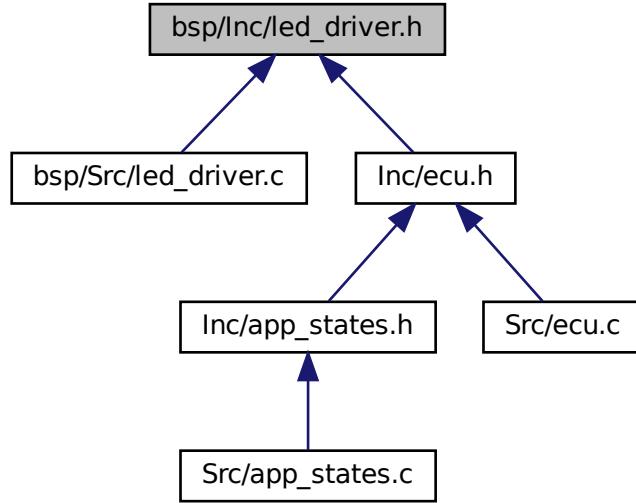
Header file for the STM32F401xx microcontroller LED driver.

```
#include "stm32f401xx_gpio_driver.h"
```

Include dependency graph for led_driver.h:



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



Classes

- struct `LED_cfg_t`

Enumerations

- enum `LED_Mode_t` { `LED_Active_High` , `LED_Active_Low` , `LED_Mode_max` }

Functions

- void `LED_Init` (const `LED_cfg_t` *led_cfg)
Initializes the LED with the specified configuration.
- void `LED_TurnOn` (const `LED_cfg_t` *led_cfg)
Turns on the LED.
- void `LED_TurnOff` (const `LED_cfg_t` *led_cfg)
Turns off the LED.
- void `LED_Toggle` (const `LED_cfg_t` *led_cfg)
Toggles the state of the LED.

7.4.1 Detailed Description

Header file for the STM32F401xx microcontroller LED driver.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file provides the necessary definitions, macros, and function prototypes for controlling LEDs connected to the STM32F401xx microcontroller. It includes configurations for LED modes and APIs for initializing and manipulating LEDs.

Copyright

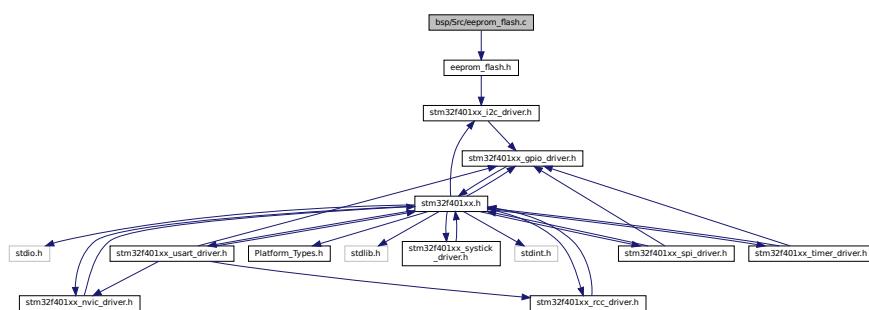
Copyright (c) 2024 Mohamed Ali Haoufa. All rights reserved.

7.5 bsp/Src/eeprom_flash.c File Reference

Implementation of EEPROM flash memory operations for STM32F401xx MCU.

```
#include "eeprom_flash.h"
```

Include dependency graph for eeprom_flash.c:



Functions

- void [EEPROM_Init](#) (void)

Initializes the EEPROM for I2C communication.
- unsigned char [EEPROM_Write_NBytes](#) (unsigned int Memory_address, unsigned char *bytes, uint8_t Data_Length)

Writes a block of data to the EEPROM.
- unsigned char [EEPROM_Read_byte](#) (unsigned int Memory_address, uint8_t *dataout, uint8_t datalen)

Reads a byte from the EEPROM.

Variables

- `uint8_t buffer [256]`
Buffer used for I2C communication with AT24C1024B EEPROM.

7.5.1 Detailed Description

Implementation of EEPROM flash memory operations for STM32F401xx MCU.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file contains the implementation of functions to initialize and manage EEPROM flash memory using I2C communication with the STM32F401xx MCU. It includes functions for writing and reading data from the EEPROM.

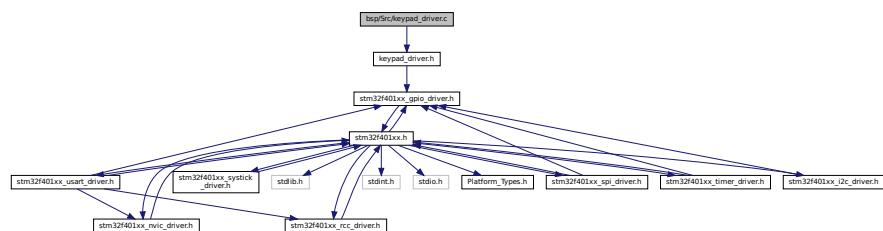
Copyright

Copyright (c) 2024 Mohamed Ali Haoufa. All rights reserved.

7.6 bsp/Src/keypad_driver.c File Reference

Source file for the keypad driver, implementing functionality to initialize and read input from a keypad connected to the STM32F401xx MCU.

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



Functions

- `void keypad_init (void)`
Initializes the keypad by configuring the GPIO pins.
- `uint8 keypad_Get_Pressed_Key (void)`
Reads the pressed key from the keypad.

7.6.1 Detailed Description

Source file for the keypad driver, implementing functionality to initialize and read input from a keypad connected to the STM32F401xx MCU.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file contains the implementation of functions for initializing the keypad and reading the pressed key values. It includes definitions and configurations for keypad rows and columns, as well as utility functions to interface with the hardware.

Copyright

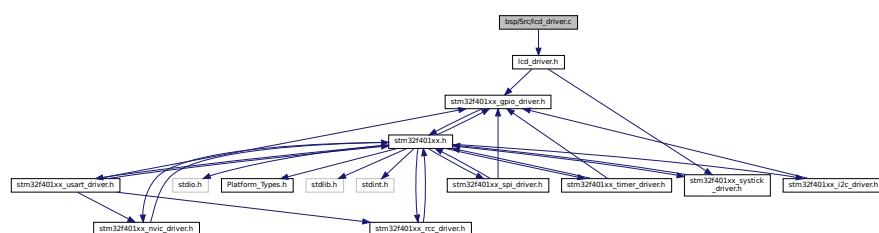
Copyright (c) 2024 Mohamed Ali Haoufa. All rights reserved.

7.7 bsp/Src/lcd_driver.c File Reference

Contains functions for controlling an LCD using an STM32F401xx MCU.

```
#include "lcd_driver.h"
```

Include dependency graph for lcd_driver.c:



Functions

- void `LCD_Init (LCD_t *LCD_cfg)`
Initializes the LCD based on user-defined configurations.
- void `LCD_Send_Command (LCD_t *LCD_cfg, uint8 command)`
Sends a command to the LCD.
- void `LCD_Send_Char (LCD_t *LCD_cfg, uint8 Char)`
Sends a character to the LCD to be displayed.
- void `LCD_Send_Char_Pos (LCD_t *LCD_cfg, uint8 Char, uint8 row, uint8 column)`
Sends a character to the LCD to be displayed at a specific location.
- void `LCD_Send_String (LCD_t *LCD_cfg, uint8 *string)`
Sends a string to the LCD to be displayed.
- void `LCD_Send_String_Pos (LCD_t *LCD_cfg, uint8 *string, uint8 row, uint8 column)`
Sends a string to the LCD to be displayed at a specific location.
- void `LCD_Send_Enable_Signal (LCD_t *LCD_cfg)`
Sends an enable signal to the LCD.
- void `LCD_Set_Cursor (LCD_t *LCD_cfg, uint8 row, uint8 column)`
Sets the location of the cursor on the LCD.

7.7.1 Detailed Description

Contains functions for controlling an LCD using an STM32F401xx MCU.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file implements the LCD driver functions including initialization, sending commands, characters, and strings to the LCD, and controlling the cursor position. It uses GPIOs of the STM32F401xx microcontroller for communication with the LCD.

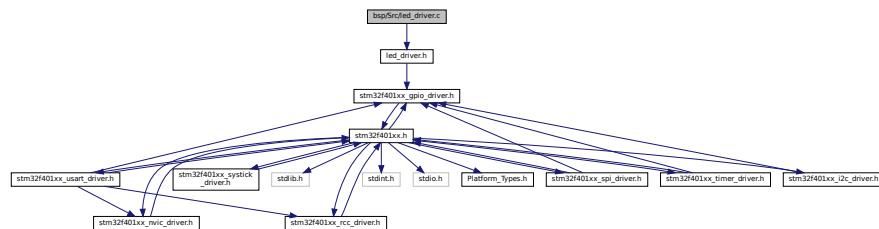
Copyright

Copyright (c) 2024

7.8 bsp/Src/led_driver.c File Reference

LED Driver Functions for STM32F401xx MCU.

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



Functions

- void `LED_Init` (const `LED_cfg_t` *led_cfg)
Initializes the LED according to the provided configuration.
- void `LED_TurnOn` (const `LED_cfg_t` *led_cfg)
Turns on the LED.
- void `LED_TurnOff` (const `LED_cfg_t` *led_cfg)
Turns off the LED.
- void `LED_Toggle` (const `LED_cfg_t` *led_cfg)
Toggles the status of the LED.

7.8.1 Detailed Description

LED Driver Functions for STM32F401xx MCU.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file provides functions to initialize and control LEDs connected to the STM32F401xx microcontroller. It includes functions to initialize the LED, turn it on or off, and toggle its state.

Copyright

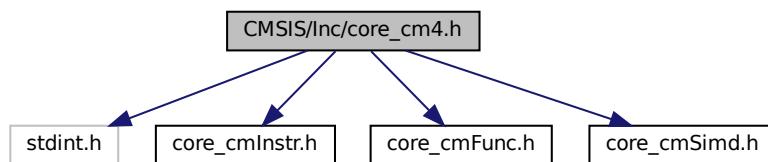
Copyright (c) 2024

7.9 CMSIS/Inc/core_cm4.h File Reference

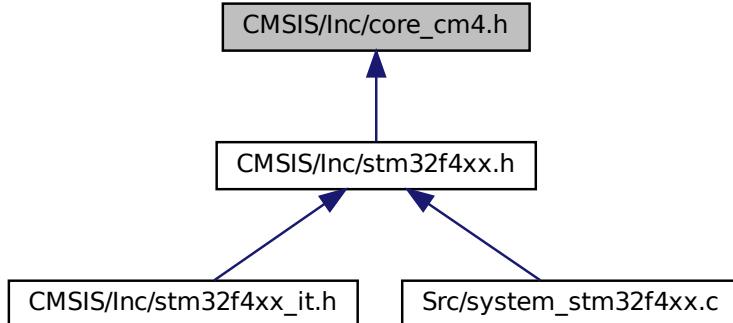
CMSIS Cortex-M4 Core Peripheral Access Layer Header File.

```
#include <stdint.h>
#include <core_cmInstr.h>
#include <core_cmFunc.h>
#include <core_cmSimd.h>
```

Include dependency graph for core_cm4.h:



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



Classes

- union [APSR_Type](#)
Union type to access the Application Program Status Register (APSR).
- union [IPSR_Type](#)
Union type to access the Interrupt Program Status Register (IPSR).
- union [xPSR_Type](#)
Union type to access the Special-Purpose Program Status Registers (xPSR).
- union [CONTROL_Type](#)
Union type to access the Control Registers (CONTROL).
- struct [NVIC_Type](#)

- **SCB_Type**
Structure type to access the Nested Vectored Interrupt Controller (NVIC).
 - **SCnSCB_Type**
Structure type to access the System Control Block (SCB).
 - **SysTick_Type**
Structure type to access the System Timer (SysTick).
 - **ITM_Type**
Structure type to access the Instrumentation Trace Macrocell Register (ITM).
 - **DWT_Type**
Structure type to access the Data Watchpoint and Trace Register (DWT).
 - **TPI_Type**
Structure type to access the Trace Port Interface Register (TPI).
 - **CoreDebug_Type**
Structure type to access the Core Debug Register (CoreDebug).
-
- `#define __CM4_CMSIS_VERSION_MAIN (0x04)`
 - `#define __CM4_CMSIS_VERSION_SUB (0x00)`
 - `#define __CM4_CMSIS_VERSION`
 - `#define __CORTEX_M (0x04)`
 - `#define __CORE_CM4_H_DEPENDANT`
 - `#define __I volatile const`
 - `#define __O volatile`
 - `#define __IO volatile`
 - `#define APSR_N_Pos 31`
 - `#define APSR_N_Msk (1UL << APSR_N_Pos)`
 - `#define APSR_Z_Pos 30`
 - `#define APSR_Z_Msk (1UL << APSR_Z_Pos)`
 - `#define APSR_C_Pos 29`
 - `#define APSR_C_Msk (1UL << APSR_C_Pos)`
 - `#define APSR_V_Pos 28`
 - `#define APSR_V_Msk (1UL << APSR_V_Pos)`
 - `#define APSR_Q_Pos 27`
 - `#define APSR_Q_Msk (1UL << APSR_Q_Pos)`
 - `#define APSR_GE_Pos 16`
 - `#define APSR_GE_Msk (0xFUL << APSR_GE_Pos)`
 - `#define IPSR_ISR_Pos 0`
 - `#define IPSR_ISR_Msk (0x1FFUL /*<< IPSR_ISR_Pos*/)`
 - `#define xPSR_N_Pos 31`
 - `#define xPSR_N_Msk (1UL << xPSR_N_Pos)`
 - `#define xPSR_Z_Pos 30`
 - `#define xPSR_Z_Msk (1UL << xPSR_Z_Pos)`
 - `#define xPSR_C_Pos 29`
 - `#define xPSR_C_Msk (1UL << xPSR_C_Pos)`
 - `#define xPSR_V_Pos 28`
 - `#define xPSR_V_Msk (1UL << xPSR_V_Pos)`
 - `#define xPSR_Q_Pos 27`
 - `#define xPSR_Q_Msk (1UL << xPSR_Q_Pos)`
 - `#define xPSR_IT_Pos 25`
 - `#define xPSR_IT_Msk (3UL << xPSR_IT_Pos)`
 - `#define xPSR_T_Pos 24`
 - `#define xPSR_T_Msk (1UL << xPSR_T_Pos)`
 - `#define xPSR_GE_Pos 16`

- #define xPSR_GE_Msk (0xFUL << xPSR_GE_Pos)
- #define xPSR_ISR_Pos 0
- #define xPSR_ISR_Msk (0x1FFUL /*<< xPSR_ISR_Pos*/)
- #define CONTROL_FPCA_Pos 2
- #define CONTROL_FPCA_Msk (1UL << CONTROL_FPCA_Pos)
- #define CONTROL_SPSEL_Pos 1
- #define CONTROL_SPSEL_Msk (1UL << CONTROL_SPSEL_Pos)
- #define CONTROL_nPRIV_Pos 0
- #define CONTROL_nPRIV_Msk (1UL /*<< CONTROL_nPRIV_Pos*/)
- #define NVIC_STIR_INTID_Pos 0
- #define NVIC_STIR_INTID_Msk (0x1FFUL /*<< NVIC_STIR_INTID_Pos*/)
- #define SCB_CPUID_IMPLEMENTER_Pos 24
- #define SCB_CPUID_IMPLEMENTER_Msk (0xFFUL << SCB_CPUID_IMPLEMENTER_Pos)
- #define SCB_CPUID_VARIANT_Pos 20
- #define SCB_CPUID_VARIANT_Msk (0xFUL << SCB_CPUID_VARIANT_Pos)
- #define SCB_CPUID_ARCHITECTURE_Pos 16
- #define SCB_CPUID_ARCHITECTURE_Msk (0xFUL << SCB_CPUID_ARCHITECTURE_Pos)
- #define SCB_CPUID_PARTNO_Pos 4
- #define SCB_CPUID_PARTNO_Msk (0xFFFFUL << SCB_CPUID_PARTNO_Pos)
- #define SCB_CPUID_REVISION_Pos 0
- #define SCB_CPUID_REVISION_Msk (0xFUL /*<< SCB_CPUID_REVISION_Pos*/)
- #define SCB_ICSR_NMIPENDSET_Pos 31
- #define SCB_ICSR_NMIPENDSET_Msk (1UL << SCB_ICSR_NMIPENDSET_Pos)
- #define SCB_ICSR_PENDSVSET_Pos 28
- #define SCB_ICSR_PENDSVSET_Msk (1UL << SCB_ICSR_PENDSVSET_Pos)
- #define SCB_ICSR_PENDSVCLR_Pos 27
- #define SCB_ICSR_PENDSVCLR_Msk (1UL << SCB_ICSR_PENDSVCLR_Pos)
- #define SCB_ICSR_PENDSTSET_Pos 26
- #define SCB_ICSR_PENDSTSET_Msk (1UL << SCB_ICSR_PENDSTSET_Pos)
- #define SCB_ICSR_PENDSTCLR_Pos 25
- #define SCB_ICSR_PENDSTCLR_Msk (1UL << SCB_ICSR_PENDSTCLR_Pos)
- #define SCB_ICSR_ISRPREEMPT_Pos 23
- #define SCB_ICSR_ISRPREEMPT_Msk (1UL << SCB_ICSR_ISRPREEMPT_Pos)
- #define SCB_ICSR_ISRPENDING_Pos 22
- #define SCB_ICSR_ISRPENDING_Msk (1UL << SCB_ICSR_ISRPENDING_Pos)
- #define SCB_ICSR_VECTPENDING_Pos 12
- #define SCB_ICSR_VECTPENDING_Msk (0x1FFUL << SCB_ICSR_VECTPENDING_Pos)
- #define SCB_ICSR_RETTOBASE_Pos 11
- #define SCB_ICSR_RETTOBASE_Msk (1UL << SCB_ICSR_RETTOBASE_Pos)
- #define SCB_ICSR_VECTACTIVE_Pos 0
- #define SCB_ICSR_VECTACTIVE_Msk (0x1FFUL /*<< SCB_ICSR_VECTACTIVE_Pos*/)
- #define SCB_VTOR_TBLOFF_Pos 7
- #define SCB_VTOR_TBLOFF_Msk (0x1FFFFFFUL << SCB_VTOR_TBLOFF_Pos)
- #define SCB_AIRCR_VECTKEY_Pos 16
- #define SCB_AIRCR_VECTKEY_Msk (0xFFFFUL << SCB_AIRCR_VECTKEY_Pos)
- #define SCB_AIRCR_VECTKEYSTAT_Pos 16
- #define SCB_AIRCR_VECTKEYSTAT_Msk (0xFFFFUL << SCB_AIRCR_VECTKEYSTAT_Pos)
- #define SCB_AIRCR_ENDIANESS_Pos 15
- #define SCB_AIRCR_ENDIANESS_Msk (1UL << SCB_AIRCR_ENDIANESS_Pos)
- #define SCB_AIRCR_PRIGROUP_Pos 8
- #define SCB_AIRCR_PRIGROUP_Msk (7UL << SCB_AIRCR_PRIGROUP_Pos)
- #define SCB_AIRCR_SYSRESETREQ_Pos 2
- #define SCB_AIRCR_SYSRESETREQ_Msk (1UL << SCB_AIRCR_SYSRESETREQ_Pos)
- #define SCB_AIRCR_VECTCLRACTIVE_Pos 1
- #define SCB_AIRCR_VECTCLRACTIVE_Msk (1UL << SCB_AIRCR_VECTCLRACTIVE_Pos)

- #define SCB_AIRCR_VECTRESET_Pos 0
- #define SCB_AIRCR_VECTRESET_Msk (1UL /*<< SCB_AIRCR_VECTRESET_Pos*/)
- #define SCB_SCR_SEVONPEND_Pos 4
- #define SCB_SCR_SEVONPEND_Msk (1UL << SCB_SCR_SEVONPEND_Pos)
- #define SCB_SCR_SLEEPDEEP_Pos 2
- #define SCB_SCR_SLEEPDEEP_Msk (1UL << SCB_SCR_SLEEPDEEP_Pos)
- #define SCB_SCR_SLEEPONEXIT_Pos 1
- #define SCB_SCR_SLEEPONEXIT_Msk (1UL << SCB_SCR_SLEEPONEXIT_Pos)
- #define SCB_CCR_STKALIGN_Pos 9
- #define SCB_CCR_STKALIGN_Msk (1UL << SCB_CCR_STKALIGN_Pos)
- #define SCB_CCR_BFHFNIGN_Pos 8
- #define SCB_CCR_BFHFNIGN_Msk (1UL << SCB_CCR_BFHFNIGN_Pos)
- #define SCB_CCR_DIV_0_TRP_Pos 4
- #define SCB_CCR_DIV_0_TRP_Msk (1UL << SCB_CCR_DIV_0_TRP_Pos)
- #define SCB_CCR_UNALIGN_TRP_Pos 3
- #define SCB_CCR_UNALIGN_TRP_Msk (1UL << SCB_CCR_UNALIGN_TRP_Pos)
- #define SCB_CCR_USERSETMPEND_Pos 1
- #define SCB_CCR_USERSETMPEND_Msk (1UL << SCB_CCR_USERSETMPEND_Pos)
- #define SCB_CCR_NONBASETHRDENA_Pos 0
- #define SCB_CCR_NONBASETHRDENA_Msk (1UL /*<< SCB_CCR_NONBASETHRDENA_Pos*/)
- #define SCB_SHCSR_USGFAULTENA_Pos 18
- #define SCB_SHCSR_USGFAULTENA_Msk (1UL << SCB_SHCSR_USGFAULTENA_Pos)
- #define SCB_SHCSR_BUSFAULTENA_Pos 17
- #define SCB_SHCSR_BUSFAULTENA_Msk (1UL << SCB_SHCSR_BUSFAULTENA_Pos)
- #define SCB_SHCSR_MEMFAULTENA_Pos 16
- #define SCB_SHCSR_MEMFAULTENA_Msk (1UL << SCB_SHCSR_MEMFAULTENA_Pos)
- #define SCB_SHCSR_SVCALLPENDED_Pos 15
- #define SCB_SHCSR_SVCALLPENDED_Msk (1UL << SCB_SHCSR_SVCALLPENDED_Pos)
- #define SCB_SHCSR_BUSFAULTPENDED_Pos 14
- #define SCB_SHCSR_BUSFAULTPENDED_Msk (1UL << SCB_SHCSR_BUSFAULTPENDED_Pos)
- #define SCB_SHCSR_MEMFAULTPENDED_Pos 13
- #define SCB_SHCSR_MEMFAULTPENDED_Msk (1UL << SCB_SHCSR_MEMFAULTPENDED_Pos)
- #define SCB_SHCSR_USGFAULTPENDED_Pos 12
- #define SCB_SHCSR_USGFAULTPENDED_Msk (1UL << SCB_SHCSR_USGFAULTPENDED_Pos)
- #define SCB_SHCSR_SYSTICKACT_Pos 11
- #define SCB_SHCSR_SYSTICKACT_Msk (1UL << SCB_SHCSR_SYSTICKACT_Pos)
- #define SCB_SHCSR_PENDSVACT_Pos 10
- #define SCB_SHCSR_PENDSVACT_Msk (1UL << SCB_SHCSR_PENDSVACT_Pos)
- #define SCB_SHCSR_MONITORACT_Pos 8
- #define SCB_SHCSR_MONITORACT_Msk (1UL << SCB_SHCSR_MONITORACT_Pos)
- #define SCB_SHCSR_SVCALLACT_Pos 7
- #define SCB_SHCSR_SVCALLACT_Msk (1UL << SCB_SHCSR_SVCALLACT_Pos)
- #define SCB_SHCSR_USGFAULTACT_Pos 3
- #define SCB_SHCSR_USGFAULTACT_Msk (1UL << SCB_SHCSR_USGFAULTACT_Pos)
- #define SCB_SHCSR_BUSFAULTACT_Pos 1
- #define SCB_SHCSR_BUSFAULTACT_Msk (1UL << SCB_SHCSR_BUSFAULTACT_Pos)
- #define SCB_SHCSR_MEMFAULTACT_Pos 0
- #define SCB_SHCSR_MEMFAULTACT_Msk (1UL /*<< SCB_SHCSR_MEMFAULTACT_Pos*/)
- #define SCB_CFSR_USGFAULTSR_Pos 16
- #define SCB_CFSR_USGFAULTSR_Msk (0xFFFFUL << SCB_CFSR_USGFAULTSR_Pos)
- #define SCB_CFSR_BUSFAULTSR_Pos 8
- #define SCB_CFSR_BUSFAULTSR_Msk (0xFFUL << SCB_CFSR_BUSFAULTSR_Pos)
- #define SCB_CFSR_MEMFAULTSR_Pos 0
- #define SCB_CFSR_MEMFAULTSR_Msk (0xFFUL /*<< SCB_CFSR_MEMFAULTSR_Pos*/)
- #define SCB_HFSR_DEBUGEVT_Pos 31

- #define SCB_HFSR_DEBUGEV_T_Msk (1UL << SCB_HFSR_DEBUGEV_T_Pos)
- #define SCB_HFSR_FORCED_Pos 30
- #define SCB_HFSR_FORCED_Msk (1UL << SCB_HFSR_FORCED_Pos)
- #define SCB_HFSR_VECTTBL_Pos 1
- #define SCB_HFSR_VECTTBL_Msk (1UL << SCB_HFSR_VECTTBL_Pos)
- #define SCB_DFSR_EXTERNAL_Pos 4
- #define SCB_DFSR_EXTERNAL_Msk (1UL << SCB_DFSR_EXTERNAL_Pos)
- #define SCB_DFSR_VCATCH_Pos 3
- #define SCB_DFSR_VCATCH_Msk (1UL << SCB_DFSR_VCATCH_Pos)
- #define SCB_DFSR_DWTTRAP_Pos 2
- #define SCB_DFSR_DWTTRAP_Msk (1UL << SCB_DFSR_DWTTRAP_Pos)
- #define SCB_DFSR_BKPT_Pos 1
- #define SCB_DFSR_BKPT_Msk (1UL << SCB_DFSR_BKPT_Pos)
- #define SCB_DFSR_HALTED_Pos 0
- #define SCB_DFSR_HALTED_Msk (1UL /*<< SCB_DFSR_HALTED_Pos*/)
- #define SCnSCB_ICTR_INTLINESNUM_Pos 0
- #define SCnSCB_ICTR_INTLINESNUM_Msk (0xFUL /*<< SCnSCB_ICTR_INTLINESNUM_Pos*/)
- #define SCnSCB_ACTLR_DISOOP_F_Pos 9
- #define SCnSCB_ACTLR_DISOOP_F_Msk (1UL << SCnSCB_ACTLR_DISOOP_F_Pos)
- #define SCnSCB_ACTLR_DISFPCA_Pos 8
- #define SCnSCB_ACTLR_DISFPCA_Msk (1UL << SCnSCB_ACTLR_DISFPCA_Pos)
- #define SCnSCB_ACTLR_DISFOLD_Pos 2
- #define SCnSCB_ACTLR_DISFOLD_Msk (1UL << SCnSCB_ACTLR_DISFOLD_Pos)
- #define SCnSCB_ACTLR_DISDEFWB_UBuf_Pos 1
- #define SCnSCB_ACTLR_DISDEFWB_UBuf_Msk (1UL << SCnSCB_ACTLR_DISDEFWB_UBuf_Pos)
- #define SCnSCB_ACTLR_DISMCYCINT_Pos 0
- #define SCnSCB_ACTLR_DISMCYCINT_Msk (1UL /*<< SCnSCB_ACTLR_DISMCYCINT_Pos*/)
- #define SysTick_CTRL_COUNTFLAG_Pos 16
- #define SysTick_CTRL_COUNTFLAG_Msk (1UL << SysTick_CTRL_COUNTFLAG_Pos)
- #define SysTick_CTRL_CLKSOURCE_Pos 2
- #define SysTick_CTRL_CLKSOURCE_Msk (1UL << SysTick_CTRL_CLKSOURCE_Pos)
- #define SysTick_CTRL_TICKINT_Pos 1
- #define SysTick_CTRL_TICKINT_Msk (1UL << SysTick_CTRL_TICKINT_Pos)
- #define SysTick_CTRL_ENABLE_Pos 0
- #define SysTick_CTRL_ENABLE_Msk (1UL /*<< SysTick_CTRL_ENABLE_Pos*/)
- #define SysTick_LOAD_RELOAD_Pos 0
- #define SysTick_LOAD_RELOAD_Msk (0xFFFFFUL /*<< SysTick_LOAD_RELOAD_Pos*/)
- #define SysTick_VAL_CURRENT_Pos 0
- #define SysTick_VAL_CURRENT_Msk (0xFFFFFUL /*<< SysTick_VAL_CURRENT_Pos*/)
- #define SysTick_CALIB_NOREF_Pos 31
- #define SysTick_CALIB_NOREF_Msk (1UL << SysTick_CALIB_NOREF_Pos)
- #define SysTick_CALIB_SKEW_Pos 30
- #define SysTick_CALIB_SKEW_Msk (1UL << SysTick_CALIB_SKEW_Pos)
- #define SysTick_CALIB_TENMS_Pos 0
- #define SysTick_CALIB_TENMS_Msk (0xFFFFFUL /*<< SysTick_CALIB_TENMS_Pos*/)
- #define ITM_TPR_PRIVMASK_Pos 0
- #define ITM_TPR_PRIVMASK_Msk (0xFUL /*<< ITM_TPR_PRIVMASK_Pos*/)
- #define ITM_TCR_BUSY_Pos 23
- #define ITM_TCR_BUSY_Msk (1UL << ITM_TCR_BUSY_Pos)
- #define ITM_TCR_TraceBusID_Pos 16
- #define ITM_TCR_TraceBusID_Msk (0x7FUL << ITM_TCR_TraceBusID_Pos)
- #define ITM_TCR_GTSFREQ_Pos 10
- #define ITM_TCR_GTSFREQ_Msk (3UL << ITM_TCR_GTSFREQ_Pos)
- #define ITM_TCR_TSPrescale_Pos 8
- #define ITM_TCR_TSPrescale_Msk (3UL << ITM_TCR_TSPrescale_Pos)

- #define ITM_TCR_SWOENA_Pos 4
- #define ITM_TCR_SWOENA_Msk (1UL << ITM_TCR_SWOENA_Pos)
- #define ITM_TCR_DWTENA_Pos 3
- #define ITM_TCR_DWTENA_Msk (1UL << ITM_TCR_DWTENA_Pos)
- #define ITM_TCR_SYNCENA_Pos 2
- #define ITM_TCR_SYNCENA_Msk (1UL << ITM_TCR_SYNCENA_Pos)
- #define ITM_TCR_TSENA_Pos 1
- #define ITM_TCR_TSENA_Msk (1UL << ITM_TCR_TSENA_Pos)
- #define ITM_TCR_ITMENA_Pos 0
- #define ITM_TCR_ITMENA_Msk (1UL /*<< ITM_TCR_ITMENA_Pos*/)
- #define ITM_IWR_ATVALIDM_Pos 0
- #define ITM_IWR_ATVALIDM_Msk (1UL /*<< ITM_IWR_ATVALIDM_Pos*/)
- #define ITM_IRR_ATREADYM_Pos 0
- #define ITM_IRR_ATREADYM_Msk (1UL /*<< ITM_IRR_ATREADYM_Pos*/)
- #define ITM_IMCR_INTEGRATION_Pos 0
- #define ITM_IMCR_INTEGRATION_Msk (1UL /*<< ITM_IMCR_INTEGRATION_Pos*/)
- #define ITM_LSR_ByteAcc_Pos 2
- #define ITM_LSR_ByteAcc_Msk (1UL << ITM_LSR_ByteAcc_Pos)
- #define ITM_LSR_Access_Pos 1
- #define ITM_LSR_Access_Msk (1UL << ITM_LSR_Access_Pos)
- #define ITM_LSR_Present_Pos 0
- #define ITM_LSR_Present_Msk (1UL /*<< ITM_LSR_Present_Pos*/)
- #define DWT_CTRL_NUMCOMP_Pos 28
- #define DWT_CTRL_NUMCOMP_Msk (0xFUL << DWT_CTRL_NUMCOMP_Pos)
- #define DWT_CTRL_NOTRCPKT_Pos 27
- #define DWT_CTRL_NOTRCPKT_Msk (0x1UL << DWT_CTRL_NOTRCPKT_Pos)
- #define DWT_CTRL_NOEXTTRIG_Pos 26
- #define DWT_CTRL_NOEXTTRIG_Msk (0x1UL << DWT_CTRL_NOEXTTRIG_Pos)
- #define DWT_CTRL_NOCYCCNT_Pos 25
- #define DWT_CTRL_NOCYCCNT_Msk (0x1UL << DWT_CTRL_NOCYCCNT_Pos)
- #define DWT_CTRL_NOPRFCNT_Pos 24
- #define DWT_CTRL_NOPRFCNT_Msk (0x1UL << DWT_CTRL_NOPRFCNT_Pos)
- #define DWT_CTRL_CYCEVTENA_Pos 22
- #define DWT_CTRL_CYCEVTENA_Msk (0x1UL << DWT_CTRL_CYCEVTENA_Pos)
- #define DWT_CTRL_FOLDEVTENA_Pos 21
- #define DWT_CTRL_FOLDEVTENA_Msk (0x1UL << DWT_CTRL_FOLDEVTENA_Pos)
- #define DWT_CTRL_LSUEVTENA_Pos 20
- #define DWT_CTRL_LSUEVTENA_Msk (0x1UL << DWT_CTRL_LSUEVTENA_Pos)
- #define DWT_CTRL_SLEEPEVTENA_Pos 19
- #define DWT_CTRL_SLEEPEVTENA_Msk (0x1UL << DWT_CTRL_SLEEPEVTENA_Pos)
- #define DWT_CTRL_EXCEVTENA_Pos 18
- #define DWT_CTRL_EXCEVTENA_Msk (0x1UL << DWT_CTRL_EXCEVTENA_Pos)
- #define DWT_CTRL_CPIEVTEA_Pos 17
- #define DWT_CTRL_CPIEVTEA_Msk (0x1UL << DWT_CTRL_CPIEVTEA_Pos)
- #define DWT_CTRL_EXCTRCENA_Pos 16
- #define DWT_CTRL_EXCTRCENA_Msk (0x1UL << DWT_CTRL_EXCTRCENA_Pos)
- #define DWT_CTRL_PCSAMPLENA_Pos 12
- #define DWT_CTRL_PCSAMPLENA_Msk (0x1UL << DWT_CTRL_PCSAMPLENA_Pos)
- #define DWT_CTRL_SYNCTAP_Pos 10
- #define DWT_CTRL_SYNCTAP_Msk (0x3UL << DWT_CTRL_SYNCTAP_Pos)
- #define DWT_CTRL_CYCTAP_Pos 9
- #define DWT_CTRL_CYCTAP_Msk (0x1UL << DWT_CTRL_CYCTAP_Pos)
- #define DWT_CTRL_POSTINIT_Pos 5
- #define DWT_CTRL_POSTINIT_Msk (0xFUL << DWT_CTRL_POSTINIT_Pos)
- #define DWT_CTRL_POSTPRESET_Pos 1

- #define DWT_CTRL_POSTPRESET_Msk (0xFUL << DWT_CTRL_POSTPRESET_Pos)
- #define DWT_CTRL_CYCCNTENA_Pos 0
- #define DWT_CTRL_CYCCNTENA_Msk (0x1UL /*<< DWT_CTRL_CYCCNTENA_Pos*/)
- #define DWT_CPLICNT_CPLICNT_Pos 0
- #define DWT_CPLICNT_CPLICNT_Msk (0xFFUL /*<< DWT_CPLICNT_CPLICNT_Pos*/)
- #define DWT_EXCCNT_EXCCNT_Pos 0
- #define DWT_EXCCNT_EXCCNT_Msk (0xFFUL /*<< DWT_EXCCNT_EXCCNT_Pos*/)
- #define DWT_SLEEP_CNT_SLEEP_CNT_Pos 0
- #define DWT_SLEEP_CNT_SLEEP_CNT_Msk (0xFFUL /*<< DWT_SLEEP_CNT_SLEEP_CNT_Pos*/)
- #define DWT_LSUCNT_LSUCNT_Pos 0
- #define DWT_LSUCNT_LSUCNT_Msk (0xFFUL /*<< DWT_LSUCNT_LSUCNT_Pos*/)
- #define DWT_FOLDCNT_FOLDCNT_Pos 0
- #define DWT_FOLDCNT_FOLDCNT_Msk (0xFFUL /*<< DWT_FOLDCNT_FOLDCNT_Pos*/)
- #define DWT_MASK_MASK_Pos 0
- #define DWT_MASK_MASK_Msk (0x1FUL /*<< DWT_MASK_MASK_Pos*/)
- #define DWT_FUNCTION_MATCHED_Pos 24
- #define DWT_FUNCTION_MATCHED_Msk (0x1UL << DWT_FUNCTION_MATCHED_Pos)
- #define DWT_FUNCTION_DATAVADDR1_Pos 16
- #define DWT_FUNCTION_DATAVADDR1_Msk (0xFUL << DWT_FUNCTION_DATAVADDR1_Pos)
- #define DWT_FUNCTION_DATAVADDR0_Pos 12
- #define DWT_FUNCTION_DATAVADDR0_Msk (0xFUL << DWT_FUNCTION_DATAVADDR0_Pos)
- #define DWT_FUNCTION_DATAVSIZE_Pos 10
- #define DWT_FUNCTION_DATAVSIZE_Msk (0x3UL << DWT_FUNCTION_DATAVSIZE_Pos)
- #define DWT_FUNCTION_LNK1ENA_Pos 9
- #define DWT_FUNCTION_LNK1ENA_Msk (0x1UL << DWT_FUNCTION_LNK1ENA_Pos)
- #define DWT_FUNCTION_DATAVMATCH_Pos 8
- #define DWT_FUNCTION_DATAVMATCH_Msk (0x1UL << DWT_FUNCTION_DATAVMATCH_Pos)
- #define DWT_FUNCTION_CYCMATCH_Pos 7
- #define DWT_FUNCTION_CYCMATCH_Msk (0x1UL << DWT_FUNCTION_CYCMATCH_Pos)
- #define DWT_FUNCTION_EMITRANGE_Pos 5
- #define DWT_FUNCTION_EMITRANGE_Msk (0x1UL << DWT_FUNCTION_EMITRANGE_Pos)
- #define DWT_FUNCTION_FUNCTION_Pos 0
- #define DWT_FUNCTION_FUNCTION_Msk (0xFUL /*<< DWT_FUNCTION_FUNCTION_Pos*/)
- #define TPI_ACPR_PRESCALER_Pos 0
- #define TPI_ACPR_PRESCALER_Msk (0xFFFFUL /*<< TPI_ACPR_PRESCALER_Pos*/)
- #define TPI_SPPR_TXMODE_Pos 0
- #define TPI_SPPR_TXMODE_Msk (0x3UL /*<< TPI_SPPR_TXMODE_Pos*/)
- #define TPI_FFSR_FtNonStop_Pos 3
- #define TPI_FFSR_FtNonStop_Msk (0x1UL << TPI_FFSR_FtNonStop_Pos)
- #define TPI_FFSR_TCPresent_Pos 2
- #define TPI_FFSR_TCPresent_Msk (0x1UL << TPI_FFSR_TCPresent_Pos)
- #define TPI_FFSR_FtStopped_Pos 1
- #define TPI_FFSR_FtStopped_Msk (0x1UL << TPI_FFSR_FtStopped_Pos)
- #define TPI_FFSR_FInProg_Pos 0
- #define TPI_FFSR_FInProg_Msk (0x1UL /*<< TPI_FFSR_FInProg_Pos*/)
- #define TPI_FFCR_TrigIn_Pos 8
- #define TPI_FFCR_TrigIn_Msk (0x1UL << TPI_FFCR_TrigIn_Pos)
- #define TPI_FFCR_EnFCont_Pos 1
- #define TPI_FFCR_EnFCont_Msk (0x1UL << TPI_FFCR_EnFCont_Pos)
- #define TPI_TRIGGER_TRIGGER_Pos 0
- #define TPI_TRIGGER_TRIGGER_Msk (0x1UL /*<< TPI_TRIGGER_TRIGGER_Pos*/)
- #define TPI_FIFO0_ITM_ATVALID_Pos 29
- #define TPI_FIFO0_ITM_ATVALID_Msk (0x3UL << TPI_FIFO0_ITM_ATVALID_Pos)
- #define TPI_FIFO0_ITM_bytectcount_Pos 27
- #define TPI_FIFO0_ITM_bytectcount_Msk (0x3UL << TPI_FIFO0_ITM_bytectcount_Pos)

- #define TPI_FIFO0_ETM_ATVALID_Pos 26
- #define TPI_FIFO0_ETM_ATVALID_Msk (0x3UL << TPI_FIFO0_ETM_ATVALID_Pos)
- #define TPI_FIFO0_ETM_bytecount_Pos 24
- #define TPI_FIFO0_ETM_bytecount_Msk (0x3UL << TPI_FIFO0_ETM_bytecount_Pos)
- #define TPI_FIFO0_ETM2_Pos 16
- #define TPI_FIFO0_ETM2_Msk (0xFFUL << TPI_FIFO0_ETM2_Pos)
- #define TPI_FIFO0_ETM1_Pos 8
- #define TPI_FIFO0_ETM1_Msk (0xFFUL << TPI_FIFO0_ETM1_Pos)
- #define TPI_FIFO0_ETM0_Pos 0
- #define TPI_FIFO0_ETM0_Msk (0xFFUL /*<< TPI_FIFO0_ETM0_Pos*/)
- #define TPI_ITATBCTR2_ATREADY_Pos 0
- #define TPI_ITATBCTR2_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR2_ATREADY_Pos*/)
- #define TPI_FIFO1_ITM_ATVALID_Pos 29
- #define TPI_FIFO1_ITM_ATVALID_Msk (0x3UL << TPI_FIFO1_ITM_ATVALID_Pos)
- #define TPI_FIFO1_ITM_bytecount_Pos 27
- #define TPI_FIFO1_ITM_bytecount_Msk (0x3UL << TPI_FIFO1_ITM_bytecount_Pos)
- #define TPI_FIFO1_ETM_ATVALID_Pos 26
- #define TPI_FIFO1_ETM_ATVALID_Msk (0x3UL << TPI_FIFO1_ETM_ATVALID_Pos)
- #define TPI_FIFO1_ETM_bytecount_Pos 24
- #define TPI_FIFO1_ETM_bytecount_Msk (0x3UL << TPI_FIFO1_ETM_bytecount_Pos)
- #define TPI_FIFO1_ITM2_Pos 16
- #define TPI_FIFO1_ITM2_Msk (0xFFUL << TPI_FIFO1_ITM2_Pos)
- #define TPI_FIFO1_ITM1_Pos 8
- #define TPI_FIFO1_ITM1_Msk (0xFFUL << TPI_FIFO1_ITM1_Pos)
- #define TPI_FIFO1_ITM0_Pos 0
- #define TPI_FIFO1_ITM0_Msk (0xFFUL /*<< TPI_FIFO1_ITM0_Pos*/)
- #define TPI_ITATBCTR0_ATREADY_Pos 0
- #define TPI_ITATBCTR0_ATREADY_Msk (0x1UL /*<< TPI_ITATBCTR0_ATREADY_Pos*/)
- #define TPI_ITCTRL_Mode_Pos 0
- #define TPI_ITCTRL_Mode_Msk (0x1UL /*<< TPI_ITCTRL_Mode_Pos*/)
- #define TPI_DEVID_NRZVALID_Pos 11
- #define TPI_DEVID_NRZVALID_Msk (0x1UL << TPI_DEVID_NRZVALID_Pos)
- #define TPI_DEVID_MANCVALID_Pos 10
- #define TPI_DEVID_MANCVALID_Msk (0x1UL << TPI_DEVID_MANCVALID_Pos)
- #define TPI_DEVID_PTINVALID_Pos 9
- #define TPI_DEVID_PTINVALID_Msk (0x1UL << TPI_DEVID_PTINVALID_Pos)
- #define TPI_DEVID_MinBufSz_Pos 6
- #define TPI_DEVID_MinBufSz_Msk (0x7UL << TPI_DEVID_MinBufSz_Pos)
- #define TPI_DEVID_AsynClkIn_Pos 5
- #define TPI_DEVID_AsynClkIn_Msk (0x1UL << TPI_DEVID_AsynClkIn_Pos)
- #define TPI_DEVID_NrTraceInput_Pos 0
- #define TPI_DEVID_NrTraceInput_Msk (0xFUL /*<< TPI_DEVID_NrTraceInput_Pos*/)
- #define TPI_DEVTYPE_MajorType_Pos 4
- #define TPI_DEVTYPE_MajorType_Msk (0xFUL << TPI_DEVTYPE_MajorType_Pos)
- #define TPI_DEVTYPE_SubType_Pos 0
- #define TPI_DEVTYPE_SubType_Msk (0xFUL /*<< TPI_DEVTYPE_SubType_Pos*/)
- #define CoreDebug_DHCSR_DBGKEY_Pos 16
- #define CoreDebug_DHCSR_DBGKEY_Msk (0xFFFFFUL << CoreDebug_DHCSR_DBGKEY_Pos)
- #define CoreDebug_DHCSR_S_RESET_ST_Pos 25
- #define CoreDebug_DHCSR_S_RESET_ST_Msk (1UL << CoreDebug_DHCSR_S_RESET_ST_Pos)
- #define CoreDebug_DHCSR_S_RETIRE_ST_Pos 24
- #define CoreDebug_DHCSR_S_RETIRE_ST_Msk (1UL << CoreDebug_DHCSR_S_RETIRE_ST_Pos)
- #define CoreDebug_DHCSR_S_LOCKUP_Pos 19
- #define CoreDebug_DHCSR_S_LOCKUP_Msk (1UL << CoreDebug_DHCSR_S_LOCKUP_Pos)
- #define CoreDebug_DHCSR_S_SLEEP_Pos 18

- #define CoreDebug_DHCSR_S_SLEEP_Msk (1UL << CoreDebug_DHCSR_S_SLEEP_Pos)
- #define CoreDebug_DHCSR_S_HALT_Pos 17
- #define CoreDebug_DHCSR_S_HALT_Msk (1UL << CoreDebug_DHCSR_S_HALT_Pos)
- #define CoreDebug_DHCSR_S_REGRDY_Pos 16
- #define CoreDebug_DHCSR_S_REGRDY_Msk (1UL << CoreDebug_DHCSR_S_REGRDY_Pos)
- #define CoreDebug_DHCSR_C_SNAPSTALL_Pos 5
- #define CoreDebug_DHCSR_C_SNAPSTALL_Msk (1UL << CoreDebug_DHCSR_C_SNAPSTALL_Pos)
- #define CoreDebug_DHCSR_C_MASKINTS_Pos 3
- #define CoreDebug_DHCSR_C_MASKINTS_Msk (1UL << CoreDebug_DHCSR_C_MASKINTS_Pos)
- #define CoreDebug_DHCSR_C_STEP_Pos 2
- #define CoreDebug_DHCSR_C_STEP_Msk (1UL << CoreDebug_DHCSR_C_STEP_Pos)
- #define CoreDebug_DHCSR_C_HALT_Pos 1
- #define CoreDebug_DHCSR_C_HALT_Msk (1UL << CoreDebug_DHCSR_C_HALT_Pos)
- #define CoreDebug_DHCSR_C_DEBUGEN_Pos 0
- #define CoreDebug_DHCSR_C_DEBUGEN_Msk (1UL /*<< CoreDebug_DHCSR_C_DEBUGEN_Pos*/)
- #define CoreDebug_DCRSR_REGWnR_Pos 16
- #define CoreDebug_DCRSR_REGWnR_Msk (1UL << CoreDebug_DCRSR_REGWnR_Pos)
- #define CoreDebug_DCRSR_REGSEL_Pos 0
- #define CoreDebug_DCRSR_REGSEL_Msk (0x1FUL /*<< CoreDebug_DCRSR_REGSEL_Pos*/)
- #define CoreDebug_DEMCR_TRCENA_Pos 24
- #define CoreDebug_DEMCR_TRCENA_Msk (1UL << CoreDebug_DEMCR_TRCENA_Pos)
- #define CoreDebug_DEMCR_MON_REQ_Pos 19
- #define CoreDebug_DEMCR_MON_REQ_Msk (1UL << CoreDebug_DEMCR_MON_REQ_Pos)
- #define CoreDebug_DEMCR_MON_STEP_Pos 18
- #define CoreDebug_DEMCR_MON_STEP_Msk (1UL << CoreDebug_DEMCR_MON_STEP_Pos)
- #define CoreDebug_DEMCR_MON_PEND_Pos 17
- #define CoreDebug_DEMCR_MON_PEND_Msk (1UL << CoreDebug_DEMCR_MON_PEND_Pos)
- #define CoreDebug_DEMCR_MON_EN_Pos 16
- #define CoreDebug_DEMCR_MON_EN_Msk (1UL << CoreDebug_DEMCR_MON_EN_Pos)
- #define CoreDebug_DEMCR_VC_HARDERR_Pos 10
- #define CoreDebug_DEMCR_VC_HARDERR_Msk (1UL << CoreDebug_DEMCR_VC_HARDERR_Pos)
- #define CoreDebug_DEMCR_VC_INTERR_Pos 9
- #define CoreDebug_DEMCR_VC_INTERR_Msk (1UL << CoreDebug_DEMCR_VC_INTERR_Pos)
- #define CoreDebug_DEMCR_VC_BUSERR_Pos 8
- #define CoreDebug_DEMCR_VC_BUSERR_Msk (1UL << CoreDebug_DEMCR_VC_BUSERR_Pos)
- #define CoreDebug_DEMCR_VC_STATERR_Pos 7
- #define CoreDebug_DEMCR_VC_STATERR_Msk (1UL << CoreDebug_DEMCR_VC_STATERR_Pos)
- #define CoreDebug_DEMCR_VC_CHKERR_Pos 6
- #define CoreDebug_DEMCR_VC_CHKERR_Msk (1UL << CoreDebug_DEMCR_VC_CHKERR_Pos)
- #define CoreDebug_DEMCR_VC_NOCPERR_Pos 5
- #define CoreDebug_DEMCR_VC_NOCPERR_Msk (1UL << CoreDebug_DEMCR_VC_NOCPERR_Pos)
- #define CoreDebug_DEMCR_VC_MMERR_Pos 4
- #define CoreDebug_DEMCR_VC_MMERR_Msk (1UL << CoreDebug_DEMCR_VC_MMERR_Pos)
- #define CoreDebug_DEMCR_VC_CORERESET_Pos 0
- #define CoreDebug_DEMCR_VC_CORERESET_Msk (1UL /*<< CoreDebug_DEMCR_VC_CORERESET_Pos*/)
- #define SCS_BASE (0xE000E000UL)
- #define ITM_BASE (0xE0000000UL)
- #define DWT_BASE (0xE0001000UL)
- #define TPI_BASE (0xE0040000UL)
- #define CoreDebug_BASE (0xE000EDF0UL)
- #define SysTick_BASE (SCS_BASE + 0x0010UL)
- #define NVIC_BASE (SCS_BASE + 0x0100UL)
- #define SCB_BASE (SCS_BASE + 0x0D00UL)
- #define SCnSCB ((SCnSCB_Type *) SCS_BASE)
- #define SCB ((SCB_Type *) SCB_BASE)

- #define SysTick ((SysTick_Type *) SysTick_BASE)
- #define NVIC ((NVIC_Type *) NVIC_BASE)
- #define ITM ((ITM_Type *) ITM_BASE)
- #define DWT ((DWT_Type *) DWT_BASE)
- #define TPI ((TPI_Type *) TPI_BASE)
- #define CoreDebug ((CoreDebug_Type *) CoreDebug_BASE)
- #define ITM_RXBUFFER_EMPTY 0x5AA55AA5
- volatile int32_t ITM_RxBuffer
- __STATIC_INLINE void NVIC_SetPriorityGrouping (uint32_t PriorityGroup)
Set Priority Grouping.
- __STATIC_INLINE uint32_t NVIC_GetPriorityGrouping (void)
Get Priority Grouping.
- __STATIC_INLINE void NVIC_EnableIRQ (IRQn_Type IRQn)
Enable External Interrupt.
- __STATIC_INLINE void NVIC_DisableIRQ (IRQn_Type IRQn)
Disable External Interrupt.
- __STATIC_INLINE uint32_t NVIC_GetPendingIRQ (IRQn_Type IRQn)
Get Pending Interrupt.
- __STATIC_INLINE void NVIC_SetPendingIRQ (IRQn_Type IRQn)
Set Pending Interrupt.
- __STATIC_INLINE void NVIC_ClearPendingIRQ (IRQn_Type IRQn)
Clear Pending Interrupt.
- __STATIC_INLINE uint32_t NVIC_GetActive (IRQn_Type IRQn)
Get Active Interrupt.
- __STATIC_INLINE void NVIC_SetPriority (IRQn_Type IRQn, uint32_t priority)
Set Interrupt Priority.
- __STATIC_INLINE uint32_t NVIC_GetPriority (IRQn_Type IRQn)
Get Interrupt Priority.
- __STATIC_INLINE uint32_t NVIC_EncodePriority (uint32_t PriorityGroup, uint32_t PreemptPriority, uint32_t SubPriority)
Encode Priority.
- __STATIC_INLINE void NVIC_DecodePriority (uint32_t Priority, uint32_t PriorityGroup, uint32_t *pPreempt←Priority, uint32_t *pSubPriority)
Decode Priority.
- __STATIC_INLINE void NVIC_SystemReset (void)
System Reset.
- __STATIC_INLINE uint32_t SysTick_Config (uint32_t ticks)
System Tick Configuration.
- __STATIC_INLINE uint32_t ITM_SendChar (uint32_t ch)
ITM Send Character.
- __STATIC_INLINE int32_t ITM_ReceiveChar (void)
ITM Receive Character.
- __STATIC_INLINE int32_t ITM_CheckChar (void)
ITM Check Character.

7.9.1 Detailed Description

CMSIS Cortex-M4 Core Peripheral Access Layer Header File.

Version

V4.10

Date

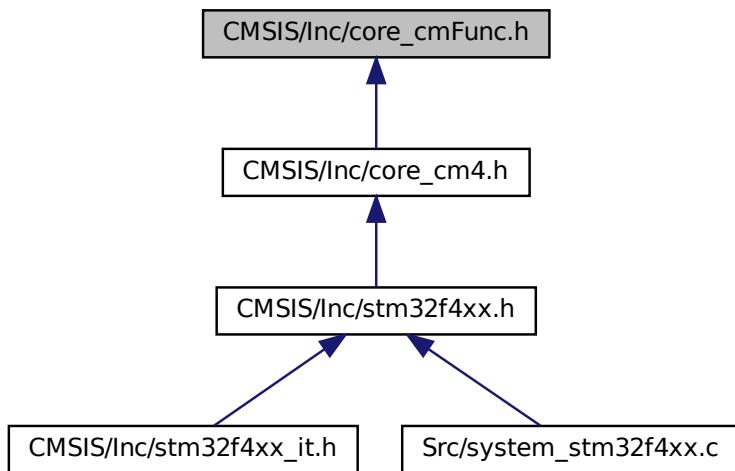
18. March 2015

Note

7.10 CMSIS/Inc/core_cmFunc.h File Reference

CMSIS Cortex-M Core Function Access Header File.

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



7.10.1 Detailed Description

CMSIS Cortex-M Core Function Access Header File.

Version

V4.10

Date

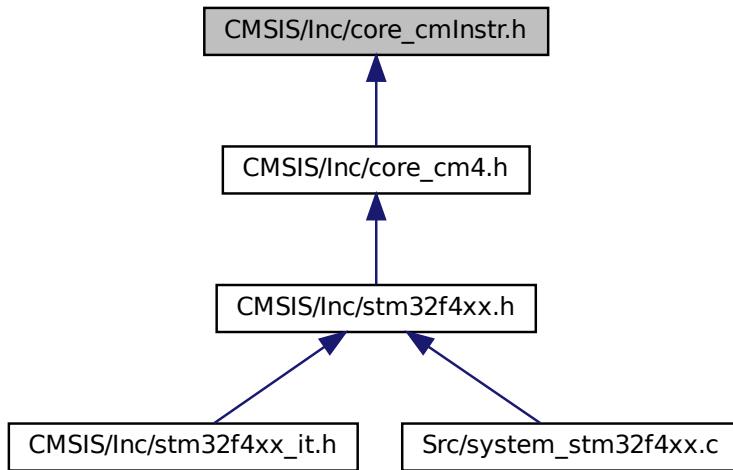
18. March 2015

Note

7.11 CMSIS/Inc/core_cmlInstr.h File Reference

CMSIS Cortex-M Core Instruction Access Header File.

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



7.11.1 Detailed Description

CMSIS Cortex-M Core Instruction Access Header File.

Version

V4.10

Date

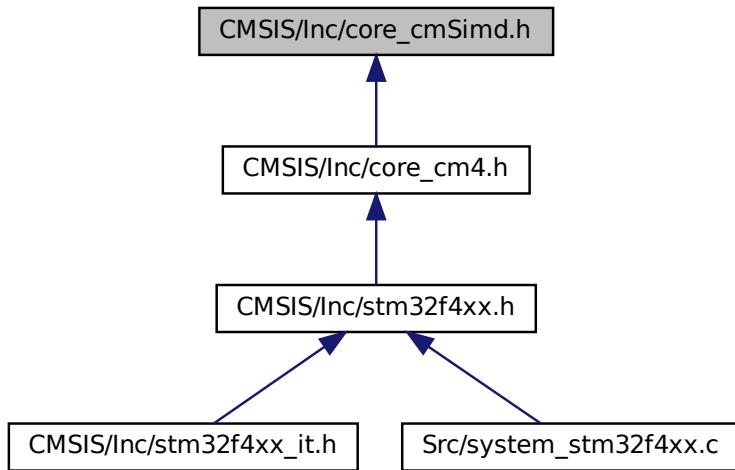
18. March 2015

Note

7.12 CMSIS/Inc/core_cmSimd.h File Reference

CMSIS Cortex-M SIMD Header File.

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



7.12.1 Detailed Description

CMSIS Cortex-M SIMD Header File.

Version

V4.10

Date

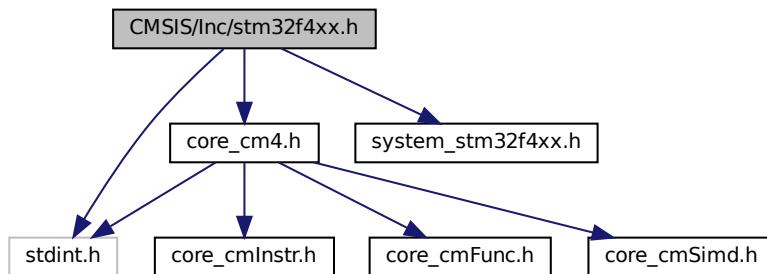
18. March 2015

Note

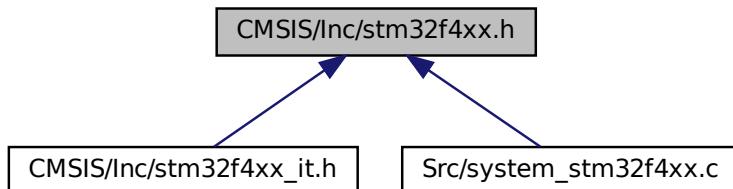
7.13 CMSIS/Inc/stm32f4xx.h File Reference

CMSIS Cortex-M4 Device Peripheral Access Layer Header File. This file contains all the peripheral register's definitions, bits definitions and memory mapping for STM32F4xx devices.

```
#include "core_cm4.h"
#include "system_stm32f4xx.h"
#include <stdint.h>
Include dependency graph for stm32f4xx.h:
```



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



Classes

- struct [ADC_TypeDef](#)
Analog to Digital Converter
- struct [ADC_Common_TypeDef](#)
- struct [CAN_TxMailBox_TypeDef](#)
Controller Area Network TxMailBox.
- struct [CAN_FIFOMailBox_TypeDef](#)
Controller Area Network FIFO MailBox.
- struct [CAN_FilterRegister_TypeDef](#)

- struct [CAN_TypeDef](#)
Controller Area Network FilterRegister.
- struct [CRC_TypeDef](#)
CRC calculation unit.
- struct [DAC_TypeDef](#)
Digital to Analog Converter.
- struct [DBGMCU_TypeDef](#)
Debug MCU.
- struct [DCMI_TypeDef](#)
DCMI.
- struct [DMA_Stream_TypeDef](#)
DMA Controller.
- struct [DMA_TypeDef](#)
- struct [DMA2D_TypeDef](#)
DMA2D Controller.
- struct [ETH_TypeDef](#)
Ethernet MAC.
- struct [EXTI_TypeDef](#)
External Interrupt/Event Controller.
- struct [FLASH_TypeDef](#)
FLASH Registers.
- struct [GPIO_TypeDef](#)
General Purpose I/O.
- struct [SYSCFG_TypeDef](#)
System configuration controller.
- struct [I2C_TypeDef](#)
Inter-integrated Circuit Interface.
- struct [IWDG_TypeDef](#)
Independent WATCHDOG.
- struct [LTDC_TypeDef](#)
LCD-TFT Display Controller.
- struct [LTDC_Layer_TypeDef](#)
LCD-TFT Display layer x Controller.
- struct [PWR_TypeDef](#)
Power Control.
- struct [RCC_TypeDef](#)
Reset and Clock Control.
- struct [RTC_TypeDef](#)
Real-Time Clock.
- struct [SAI_TypeDef](#)
Serial Audio Interface.
- struct [SAI_Block_TypeDef](#)
- struct [SDIO_TypeDef](#)
SD host Interface.
- struct [SPI_TypeDef](#)
Serial Peripheral Interface.
- struct [TIM_TypeDef](#)
TIM.
- struct [USART_TypeDef](#)
Universal Synchronous Asynchronous Receiver Transmitter.

- struct [WWDG_TypeDef](#)
Window WATCHDOG.
- struct [CRYP_TypeDef](#)
Crypto Processor.
- struct [HASH_TypeDef](#)
HASH.
- struct [HASH_DIGEST_TypeDef](#)
HASH_DIGEST.
- struct [RNG_TypeDef](#)
RNG.

Macros

- #define **STM32F401xx**
- #define **HSE_VALUE** ((uint32_t)25000000)
Comment the line below if you will not use the peripherals drivers. In this case, these drivers will not be included and the application code will be based on direct access to peripherals registers.
- #define **HSE_STARTUP_TIMEOUT** ((uint16_t)0x05000)
In the following line adjust the External High Speed oscillator (HSE) Startup Timeout value.
- #define **HSI_VALUE** ((uint32_t)16000000)
- #define **_STM32F4XX_STDPERIPH_VERSION_MAIN** (0x01)
STM32F4XX Standard Peripherals Library version number V1.8.0.
- #define **_STM32F4XX_STDPERIPH_VERSION_SUB1** (0x08)
- #define **_STM32F4XX_STDPERIPH_VERSION_SUB2** (0x00)
- #define **_STM32F4XX_STDPERIPH_VERSION_RC** (0x00)
- #define **_STM32F4XX_STDPERIPH_VERSION**
- #define **_CM4_REV** 0x0001
Configuration of the Cortex-M4 Processor and Core Peripherals.
- #define **_MPU_PRESENT** 1
- #define **_NVIC_PRIO_BITS** 4
- #define **_Vendor_SysTickConfig** 0
- #define **_FPU_PRESENT** 1
- #define **IS_FUNCTIONAL_STATE**(STATE) (((STATE) == **DISABLE**) || ((STATE) == **ENABLE**))
- #define **FLASH_BASE** ((uint32_t)0x08000000)
- #define **CCMDATARAM_BASE** ((uint32_t)0x10000000)
- #define **SRAM1_BASE** ((uint32_t)0x20000000)
- #define **PERIPH_BASE** ((uint32_t)0x40000000)
- #define **BKPSRAM_BASE** ((uint32_t)0x40024000)
- #define **CCMDATARAM_BB_BASE** ((uint32_t)0x12000000)
- #define **SRAM1_BB_BASE** ((uint32_t)0x22000000)
- #define **PERIPH_BB_BASE** ((uint32_t)0x42000000)
- #define **BKPSRAM_BB_BASE** ((uint32_t)0x42480000)
- #define **SRAM_BASE** **SRAM1_BASE**
- #define **SRAM_BB_BASE** **SRAM1_BB_BASE**
- #define **APB1PERIPH_BASE** **PERIPH_BASE**
- #define **APB2PERIPH_BASE** (**PERIPH_BASE** + 0x00010000)
- #define **AHB1PERIPH_BASE** (**PERIPH_BASE** + 0x00020000)
- #define **AHB2PERIPH_BASE** (**PERIPH_BASE** + 0x10000000)
- #define **TIM2_BASE** (**APB1PERIPH_BASE** + 0x0000)
- #define **TIM3_BASE** (**APB1PERIPH_BASE** + 0x0400)
- #define **TIM4_BASE** (**APB1PERIPH_BASE** + 0x0800)
- #define **TIM5_BASE** (**APB1PERIPH_BASE** + 0x0C00)

- #define **TIM6_BASE** (APB1PERIPH_BASE + 0x1000)
- #define **TIM7_BASE** (APB1PERIPH_BASE + 0x1400)
- #define **TIM12_BASE** (APB1PERIPH_BASE + 0x1800)
- #define **TIM13_BASE** (APB1PERIPH_BASE + 0x1C00)
- #define **TIM14_BASE** (APB1PERIPH_BASE + 0x2000)
- #define **RTC_BASE** (APB1PERIPH_BASE + 0x2800)
- #define **WWDG_BASE** (APB1PERIPH_BASE + 0x2C00)
- #define **IWDG_BASE** (APB1PERIPH_BASE + 0x3000)
- #define **I2S2ext_BASE** (APB1PERIPH_BASE + 0x3400)
- #define **SPI2_BASE** (APB1PERIPH_BASE + 0x3800)
- #define **SPI3_BASE** (APB1PERIPH_BASE + 0x3C00)
- #define **I2S3ext_BASE** (APB1PERIPH_BASE + 0x4000)
- #define **USART2_BASE** (APB1PERIPH_BASE + 0x4400)
- #define **USART3_BASE** (APB1PERIPH_BASE + 0x4800)
- #define **UART4_BASE** (APB1PERIPH_BASE + 0x4C00)
- #define **UART5_BASE** (APB1PERIPH_BASE + 0x5000)
- #define **I2C1_BASE** (APB1PERIPH_BASE + 0x5400)
- #define **I2C2_BASE** (APB1PERIPH_BASE + 0x5800)
- #define **I2C3_BASE** (APB1PERIPH_BASE + 0x5C00)
- #define **CAN1_BASE** (APB1PERIPH_BASE + 0x6400)
- #define **CAN2_BASE** (APB1PERIPH_BASE + 0x6800)
- #define **PWR_BASE** (APB1PERIPH_BASE + 0x7000)
- #define **DAC_BASE** (APB1PERIPH_BASE + 0x7400)
- #define **UART7_BASE** (APB1PERIPH_BASE + 0x7800)
- #define **UART8_BASE** (APB1PERIPH_BASE + 0x7C00)
- #define **TIM1_BASE** (APB2PERIPH_BASE + 0x0000)
- #define **TIM8_BASE** (APB2PERIPH_BASE + 0x0400)
- #define **USART1_BASE** (APB2PERIPH_BASE + 0x1000)
- #define **USART6_BASE** (APB2PERIPH_BASE + 0x1400)
- #define **UART9_BASE** (APB2PERIPH_BASE + 0x1800U)
- #define **UART10_BASE** (APB2PERIPH_BASE + 0x1C00U)
- #define **ADC1_BASE** (APB2PERIPH_BASE + 0x2000)
- #define **ADC2_BASE** (APB2PERIPH_BASE + 0x2100)
- #define **ADC3_BASE** (APB2PERIPH_BASE + 0x2200)
- #define **ADC_BASE** (APB2PERIPH_BASE + 0x2300)
- #define **SDIO_BASE** (APB2PERIPH_BASE + 0x2C00)
- #define **SPI1_BASE** (APB2PERIPH_BASE + 0x3000)
- #define **SPI4_BASE** (APB2PERIPH_BASE + 0x3400)
- #define **SYSCFG_BASE** (APB2PERIPH_BASE + 0x3800)
- #define **EXTI_BASE** (APB2PERIPH_BASE + 0x3C00)
- #define **TIM9_BASE** (APB2PERIPH_BASE + 0x4000)
- #define **TIM10_BASE** (APB2PERIPH_BASE + 0x4400)
- #define **TIM11_BASE** (APB2PERIPH_BASE + 0x4800)
- #define **SPI5_BASE** (APB2PERIPH_BASE + 0x5000)
- #define **SPI6_BASE** (APB2PERIPH_BASE + 0x5400)
- #define **SAI1_BASE** (APB2PERIPH_BASE + 0x5800)
- #define **SAI1_Block_A_BASE** (SAI1_BASE + 0x004)
- #define **SAI1_Block_B_BASE** (SAI1_BASE + 0x024)
- #define **LTDC_BASE** (APB2PERIPH_BASE + 0x6800)
- #define **LTDC_Layer1_BASE** (LTDC_BASE + 0x84)
- #define **LTDC_Layer2_BASE** (LTDC_BASE + 0x104)
- #define **GPIOA_BASE** (AHB1PERIPH_BASE + 0x0000)
- #define **GPIOB_BASE** (AHB1PERIPH_BASE + 0x0400)
- #define **GPIOC_BASE** (AHB1PERIPH_BASE + 0x0800)
- #define **GPIOD_BASE** (AHB1PERIPH_BASE + 0x0C00)

- #define **GPIOE_BASE** (AHB1PERIPH_BASE + 0x1000)
- #define **GPIOF_BASE** (AHB1PERIPH_BASE + 0x1400)
- #define **GPIOG_BASE** (AHB1PERIPH_BASE + 0x1800)
- #define **GPIOH_BASE** (AHB1PERIPH_BASE + 0x1C00)
- #define **GPIOI_BASE** (AHB1PERIPH_BASE + 0x2000)
- #define **GPIOJ_BASE** (AHB1PERIPH_BASE + 0x2400)
- #define **GPIOK_BASE** (AHB1PERIPH_BASE + 0x2800)
- #define **CRC_BASE** (AHB1PERIPH_BASE + 0x3000)
- #define **RCC_BASE** (AHB1PERIPH_BASE + 0x3800)
- #define **FLASH_R_BASE** (AHB1PERIPH_BASE + 0x3C00)
- #define **DMA1_BASE** (AHB1PERIPH_BASE + 0x6000)
- #define **DMA1_Stream0_BASE** (DMA1_BASE + 0x010)
- #define **DMA1_Stream1_BASE** (DMA1_BASE + 0x028)
- #define **DMA1_Stream2_BASE** (DMA1_BASE + 0x040)
- #define **DMA1_Stream3_BASE** (DMA1_BASE + 0x058)
- #define **DMA1_Stream4_BASE** (DMA1_BASE + 0x070)
- #define **DMA1_Stream5_BASE** (DMA1_BASE + 0x088)
- #define **DMA1_Stream6_BASE** (DMA1_BASE + 0x0A0)
- #define **DMA1_Stream7_BASE** (DMA1_BASE + 0x0B8)
- #define **DMA2_BASE** (AHB1PERIPH_BASE + 0x6400)
- #define **DMA2_Stream0_BASE** (DMA2_BASE + 0x010)
- #define **DMA2_Stream1_BASE** (DMA2_BASE + 0x028)
- #define **DMA2_Stream2_BASE** (DMA2_BASE + 0x040)
- #define **DMA2_Stream3_BASE** (DMA2_BASE + 0x058)
- #define **DMA2_Stream4_BASE** (DMA2_BASE + 0x070)
- #define **DMA2_Stream5_BASE** (DMA2_BASE + 0x088)
- #define **DMA2_Stream6_BASE** (DMA2_BASE + 0x0A0)
- #define **DMA2_Stream7_BASE** (DMA2_BASE + 0x0B8)
- #define **ETH_BASE** (AHB1PERIPH_BASE + 0x8000)
- #define **ETH_MAC_BASE** (ETH_BASE)
- #define **ETH_MMC_BASE** (ETH_BASE + 0x0100)
- #define **ETH_PTP_BASE** (ETH_BASE + 0x0700)
- #define **ETH_DMA_BASE** (ETH_BASE + 0x1000)
- #define **DMA2D_BASE** (AHB1PERIPH_BASE + 0xB000)
- #define **DCMI_BASE** (AHB2PERIPH_BASE + 0x50000)
- #define **CRYP_BASE** (AHB2PERIPH_BASE + 0x60000)
- #define **HASH_BASE** (AHB2PERIPH_BASE + 0x60400)
- #define **HASH_DIGEST_BASE** (AHB2PERIPH_BASE + 0x60710)
- #define **RNG_BASE** (AHB2PERIPH_BASE + 0x60800)
- #define **DBGMCU_BASE** ((uint32_t)0xE0042000)
- #define **TIM2** ((TIM_TypeDef *) **TIM2_BASE**)
- #define **TIM3** ((TIM_TypeDef *) TIM3_BASE)
- #define **TIM4** ((TIM_TypeDef *) TIM4_BASE)
- #define **TIM5** ((TIM_TypeDef *) TIM5_BASE)
- #define **TIM6** ((TIM_TypeDef *) TIM6_BASE)
- #define **TIM7** ((TIM_TypeDef *) TIM7_BASE)
- #define **TIM12** ((TIM_TypeDef *) TIM12_BASE)
- #define **TIM13** ((TIM_TypeDef *) TIM13_BASE)
- #define **TIM14** ((TIM_TypeDef *) TIM14_BASE)
- #define **RTC** ((RTC_TypeDef *) RTC_BASE)
- #define **WWDG** ((WWDG_TypeDef *) WWDG_BASE)
- #define **IWDG** ((IWDG_TypeDef *) IWDG_BASE)
- #define **I2S2ext** ((SPI_TypeDef *) I2S2ext_BASE)
- #define **SPI2** ((SPI_TypeDef *) **SPI2_BASE**)
- #define **SPI3** ((SPI_TypeDef *) SPI3_BASE)

```
• #define I2S3ext ((SPI_TypeDef *) I2S3ext_BASE)
• #define USART2 ((USART_TypeDef *) USART2_BASE)
• #define USART3 ((USART_TypeDef *) USART3_BASE)
• #define UART4 ((USART_TypeDef *) UART4_BASE)
• #define UART5 ((USART_TypeDef *) UART5_BASE)
• #define I2C1 ((I2C_TypeDef *) I2C1_BASE)
• #define I2C2 ((I2C_TypeDef *) I2C2_BASE)
• #define I2C3 ((I2C_TypeDef *) I2C3_BASE)
• #define CAN1 ((CAN_TypeDef *) CAN1_BASE)
• #define CAN2 ((CAN_TypeDef *) CAN2_BASE)
• #define PWR ((PWR_TypeDef *) PWR_BASE)
• #define DAC ((DAC_TypeDef *) DAC_BASE)
• #define UART7 ((USART_TypeDef *) UART7_BASE)
• #define UART8 ((USART_TypeDef *) UART8_BASE)
• #define UART9 ((USART_TypeDef *) UART9_BASE)
• #define UART10 ((USART_TypeDef *) UART10_BASE)
• #define TIM1 ((TIM_TypeDef *) TIM1_BASE)
• #define TIM8 ((TIM_TypeDef *) TIM8_BASE)
• #define USART1 ((USART_TypeDef *) USART1_BASE)
• #define USART6 ((USART_TypeDef *) USART6_BASE)
• #define ADC ((ADC_Common_TypeDef *) ADC_BASE)
• #define ADC1 ((ADC_TypeDef *) ADC1_BASE)
• #define ADC2 ((ADC_TypeDef *) ADC2_BASE)
• #define ADC3 ((ADC_TypeDef *) ADC3_BASE)
• #define SDIO ((SDIO_TypeDef *) SDIO_BASE)
• #define SPI1 ((SPI_TypeDef *) SPI1_BASE)
• #define SPI4 ((SPI_TypeDef *) SPI4_BASE)
• #define SYSCFG ((SYSCFG_TypeDef *) SYSCFG_BASE)
• #define EXTI ((EXTI_TypeDef *) EXTI_BASE)
• #define TIM9 ((TIM_TypeDef *) TIM9_BASE)
• #define TIM10 ((TIM_TypeDef *) TIM10_BASE)
• #define TIM11 ((TIM_TypeDef *) TIM11_BASE)
• #define SPI5 ((SPI_TypeDef *) SPI5_BASE)
• #define SPI6 ((SPI_TypeDef *) SPI6_BASE)
• #define SAI1 ((SAI_TypeDef *) SAI1_BASE)
• #define SAI1_Block_A ((SAI_Block_TypeDef *) SAI1_Block_A_BASE)
• #define SAI1_Block_B ((SAI_Block_TypeDef *) SAI1_Block_B_BASE)
• #define LTDC ((LTDC_TypeDef *) LTDC_BASE)
• #define LTDC_Layer1 ((LTDC_Layer_TypeDef *) LTDC_Layer1_BASE)
• #define LTDC_Layer2 ((LTDC_Layer_TypeDef *) LTDC_Layer2_BASE)
• #define GPIOA ((GPIO_TypeDef *) GPIOA_BASE)
• #define GPIOB ((GPIO_TypeDef *) GPIOB_BASE)
• #define GPIOC ((GPIO_TypeDef *) GPIOC_BASE)
• #define GPIOD ((GPIO_TypeDef *) GPIOD_BASE)
• #define GPIOE ((GPIO_TypeDef *) GPIOE_BASE)
• #define GPIOF ((GPIO_TypeDef *) GPIOF_BASE)
• #define GPIOG ((GPIO_TypeDef *) GPIOG_BASE)
• #define GPIOH ((GPIO_TypeDef *) GPIOH_BASE)
• #define GPIOI ((GPIO_TypeDef *) GPIOI_BASE)
• #define GPIOJ ((GPIO_TypeDef *) GPIOJ_BASE)
• #define GPIOK ((GPIO_TypeDef *) GPIOK_BASE)
• #define CRC ((CRC_TypeDef *) CRC_BASE)
• #define RCC ((RCC_TypeDef *) RCC_BASE)
• #define FLASH ((FLASH_TypeDef *) FLASH_R_BASE)
• #define DMA1 ((DMA_TypeDef *) DMA1_BASE)
```

- #define **DMA1_Stream0** ((DMA_Stream_TypeDef *) DMA1_Stream0_BASE)
- #define **DMA1_Stream1** ((DMA_Stream_TypeDef *) DMA1_Stream1_BASE)
- #define **DMA1_Stream2** ((DMA_Stream_TypeDef *) DMA1_Stream2_BASE)
- #define **DMA1_Stream3** ((DMA_Stream_TypeDef *) DMA1_Stream3_BASE)
- #define **DMA1_Stream4** ((DMA_Stream_TypeDef *) DMA1_Stream4_BASE)
- #define **DMA1_Stream5** ((DMA_Stream_TypeDef *) DMA1_Stream5_BASE)
- #define **DMA1_Stream6** ((DMA_Stream_TypeDef *) DMA1_Stream6_BASE)
- #define **DMA1_Stream7** ((DMA_Stream_TypeDef *) DMA1_Stream7_BASE)
- #define **DMA2** ((DMA_TypeDef *) DMA2_BASE)
- #define **DMA2_Stream0** ((DMA_Stream_TypeDef *) DMA2_Stream0_BASE)
- #define **DMA2_Stream1** ((DMA_Stream_TypeDef *) DMA2_Stream1_BASE)
- #define **DMA2_Stream2** ((DMA_Stream_TypeDef *) DMA2_Stream2_BASE)
- #define **DMA2_Stream3** ((DMA_Stream_TypeDef *) DMA2_Stream3_BASE)
- #define **DMA2_Stream4** ((DMA_Stream_TypeDef *) DMA2_Stream4_BASE)
- #define **DMA2_Stream5** ((DMA_Stream_TypeDef *) DMA2_Stream5_BASE)
- #define **DMA2_Stream6** ((DMA_Stream_TypeDef *) DMA2_Stream6_BASE)
- #define **DMA2_Stream7** ((DMA_Stream_TypeDef *) DMA2_Stream7_BASE)
- #define **ETH** ((ETH_TypeDef *) ETH_BASE)
- #define **DMA2D** ((DMA2D_TypeDef *) DMA2D_BASE)
- #define **DCMI** ((DCMI_TypeDef *) DCMI_BASE)
- #define **CRYP** ((CRYP_TypeDef *) CRYP_BASE)
- #define **HASH** ((HASH_TypeDef *) HASH_BASE)
- #define **HASH_DIGEST** ((HASH_DIGEST_TypeDef *) HASH_DIGEST_BASE)
- #define **RNG** ((RNG_TypeDef *) RNG_BASE)
- #define **DBGMCU** ((DBGMCU_TypeDef *) DBGMCU_BASE)
- #define **ADC_SR_AWD** ((uint8_t)0x01)
- #define **ADC_SR_EOC** ((uint8_t)0x02)
- #define **ADC_SR_JEOC** ((uint8_t)0x04)
- #define **ADC_SR_JSTRT** ((uint8_t)0x08)
- #define **ADC_SR_STRT** ((uint8_t)0x10)
- #define **ADC_SR_OVR** ((uint8_t)0x20)
- #define **ADC_CR1_AWDCH** ((uint32_t)0x00000001F)
- #define **ADC_CR1_AWDCH_0** ((uint32_t)0x000000001)
- #define **ADC_CR1_AWDCH_1** ((uint32_t)0x000000002)
- #define **ADC_CR1_AWDCH_2** ((uint32_t)0x000000004)
- #define **ADC_CR1_AWDCH_3** ((uint32_t)0x000000008)
- #define **ADC_CR1_AWDCH_4** ((uint32_t)0x000000010)
- #define **ADC_CR1_EOCIE** ((uint32_t)0x000000020)
- #define **ADC_CR1_AWDIE** ((uint32_t)0x000000040)
- #define **ADC_CR1_JEOCIE** ((uint32_t)0x000000080)
- #define **ADC_CR1_SCAN** ((uint32_t)0x000000100)
- #define **ADC_CR1_AWDSGL** ((uint32_t)0x000000200)
- #define **ADC_CR1_JAUTO** ((uint32_t)0x000000400)
- #define **ADC_CR1_DISCEN** ((uint32_t)0x000000800)
- #define **ADC_CR1_JDISCEN** ((uint32_t)0x00001000)
- #define **ADC_CR1_DISCNUM** ((uint32_t)0x0000E000)
- #define **ADC_CR1_DISCNUM_0** ((uint32_t)0x00002000)
- #define **ADC_CR1_DISCNUM_1** ((uint32_t)0x00004000)
- #define **ADC_CR1_DISCNUM_2** ((uint32_t)0x00008000)
- #define **ADC_CR1_JAWDEN** ((uint32_t)0x00400000)
- #define **ADC_CR1_AWDEN** ((uint32_t)0x00800000)
- #define **ADC_CR1_RES** ((uint32_t)0x03000000)
- #define **ADC_CR1_RES_0** ((uint32_t)0x01000000)
- #define **ADC_CR1_RES_1** ((uint32_t)0x02000000)
- #define **ADC_CR1_OVRIE** ((uint32_t)0x04000000)

- #define ADC_CR2_ADON ((uint32_t)0x00000001)
- #define ADC_CR2_CONT ((uint32_t)0x00000002)
- #define ADC_CR2_DMA ((uint32_t)0x00000100)
- #define ADC_CR2_DDS ((uint32_t)0x00000200)
- #define ADC_CR2_EOCS ((uint32_t)0x00000400)
- #define ADC_CR2_ALIGN ((uint32_t)0x00000800)
- #define ADC_CR2_JEXTSEL ((uint32_t)0x000F0000)
- #define ADC_CR2_JEXTSEL_0 ((uint32_t)0x00010000)
- #define ADC_CR2_JEXTSEL_1 ((uint32_t)0x00020000)
- #define ADC_CR2_JEXTSEL_2 ((uint32_t)0x00040000)
- #define ADC_CR2_JEXTSEL_3 ((uint32_t)0x00080000)
- #define ADC_CR2_JEXTEN ((uint32_t)0x00300000)
- #define ADC_CR2_JEXTEN_0 ((uint32_t)0x00100000)
- #define ADC_CR2_JEXTEN_1 ((uint32_t)0x00200000)
- #define ADC_CR2_JSWSTART ((uint32_t)0x00400000)
- #define ADC_CR2_EXTSEL ((uint32_t)0x0F000000)
- #define ADC_CR2_EXTSEL_0 ((uint32_t)0x01000000)
- #define ADC_CR2_EXTSEL_1 ((uint32_t)0x02000000)
- #define ADC_CR2_EXTSEL_2 ((uint32_t)0x04000000)
- #define ADC_CR2_EXTSEL_3 ((uint32_t)0x08000000)
- #define ADC_CR2_EXTEN ((uint32_t)0x30000000)
- #define ADC_CR2_EXTEN_0 ((uint32_t)0x10000000)
- #define ADC_CR2_EXTEN_1 ((uint32_t)0x20000000)
- #define ADC_CR2_SWSTART ((uint32_t)0x40000000)
- #define ADC_SMPR1_SMP10 ((uint32_t)0x00000007)
- #define ADC_SMPR1_SMP10_0 ((uint32_t)0x00000001)
- #define ADC_SMPR1_SMP10_1 ((uint32_t)0x00000002)
- #define ADC_SMPR1_SMP10_2 ((uint32_t)0x00000004)
- #define ADC_SMPR1_SMP11 ((uint32_t)0x00000038)
- #define ADC_SMPR1_SMP11_0 ((uint32_t)0x00000008)
- #define ADC_SMPR1_SMP11_1 ((uint32_t)0x00000010)
- #define ADC_SMPR1_SMP11_2 ((uint32_t)0x00000020)
- #define ADC_SMPR1_SMP12 ((uint32_t)0x000001C0)
- #define ADC_SMPR1_SMP12_0 ((uint32_t)0x00000040)
- #define ADC_SMPR1_SMP12_1 ((uint32_t)0x00000080)
- #define ADC_SMPR1_SMP12_2 ((uint32_t)0x00000100)
- #define ADC_SMPR1_SMP13 ((uint32_t)0x00000E00)
- #define ADC_SMPR1_SMP13_0 ((uint32_t)0x00000200)
- #define ADC_SMPR1_SMP13_1 ((uint32_t)0x00000400)
- #define ADC_SMPR1_SMP13_2 ((uint32_t)0x00000800)
- #define ADC_SMPR1_SMP14 ((uint32_t)0x00007000)
- #define ADC_SMPR1_SMP14_0 ((uint32_t)0x00001000)
- #define ADC_SMPR1_SMP14_1 ((uint32_t)0x00002000)
- #define ADC_SMPR1_SMP14_2 ((uint32_t)0x00004000)
- #define ADC_SMPR1_SMP15 ((uint32_t)0x00038000)
- #define ADC_SMPR1_SMP15_0 ((uint32_t)0x00008000)
- #define ADC_SMPR1_SMP15_1 ((uint32_t)0x00010000)
- #define ADC_SMPR1_SMP15_2 ((uint32_t)0x00020000)
- #define ADC_SMPR1_SMP16 ((uint32_t)0x001C0000)
- #define ADC_SMPR1_SMP16_0 ((uint32_t)0x00040000)
- #define ADC_SMPR1_SMP16_1 ((uint32_t)0x00080000)
- #define ADC_SMPR1_SMP16_2 ((uint32_t)0x00100000)
- #define ADC_SMPR1_SMP17 ((uint32_t)0x00E00000)
- #define ADC_SMPR1_SMP17_0 ((uint32_t)0x00200000)
- #define ADC_SMPR1_SMP17_1 ((uint32_t)0x00400000)

- #define ADC_SMPR1_SMP17_2 ((uint32_t)0x00800000)
- #define ADC_SMPR1_SMP18 ((uint32_t)0x07000000)
- #define ADC_SMPR1_SMP18_0 ((uint32_t)0x01000000)
- #define ADC_SMPR1_SMP18_1 ((uint32_t)0x02000000)
- #define ADC_SMPR1_SMP18_2 ((uint32_t)0x04000000)
- #define ADC_SMPR2_SMP0 ((uint32_t)0x00000007)
- #define ADC_SMPR2_SMP0_0 ((uint32_t)0x00000001)
- #define ADC_SMPR2_SMP0_1 ((uint32_t)0x00000002)
- #define ADC_SMPR2_SMP0_2 ((uint32_t)0x00000004)
- #define ADC_SMPR2_SMP1 ((uint32_t)0x00000038)
- #define ADC_SMPR2_SMP1_0 ((uint32_t)0x00000008)
- #define ADC_SMPR2_SMP1_1 ((uint32_t)0x00000010)
- #define ADC_SMPR2_SMP1_2 ((uint32_t)0x00000020)
- #define ADC_SMPR2_SMP2 ((uint32_t)0x000001C0)
- #define ADC_SMPR2_SMP2_0 ((uint32_t)0x00000040)
- #define ADC_SMPR2_SMP2_1 ((uint32_t)0x00000080)
- #define ADC_SMPR2_SMP2_2 ((uint32_t)0x00000100)
- #define ADC_SMPR2_SMP3 ((uint32_t)0x00000E00)
- #define ADC_SMPR2_SMP3_0 ((uint32_t)0x00000200)
- #define ADC_SMPR2_SMP3_1 ((uint32_t)0x00000400)
- #define ADC_SMPR2_SMP3_2 ((uint32_t)0x00000800)
- #define ADC_SMPR2_SMP4 ((uint32_t)0x00007000)
- #define ADC_SMPR2_SMP4_0 ((uint32_t)0x00001000)
- #define ADC_SMPR2_SMP4_1 ((uint32_t)0x00002000)
- #define ADC_SMPR2_SMP4_2 ((uint32_t)0x00004000)
- #define ADC_SMPR2_SMP5 ((uint32_t)0x00038000)
- #define ADC_SMPR2_SMP5_0 ((uint32_t)0x00008000)
- #define ADC_SMPR2_SMP5_1 ((uint32_t)0x00010000)
- #define ADC_SMPR2_SMP5_2 ((uint32_t)0x00020000)
- #define ADC_SMPR2_SMP6 ((uint32_t)0x001C0000)
- #define ADC_SMPR2_SMP6_0 ((uint32_t)0x00040000)
- #define ADC_SMPR2_SMP6_1 ((uint32_t)0x00080000)
- #define ADC_SMPR2_SMP6_2 ((uint32_t)0x00100000)
- #define ADC_SMPR2_SMP7 ((uint32_t)0x00E00000)
- #define ADC_SMPR2_SMP7_0 ((uint32_t)0x00200000)
- #define ADC_SMPR2_SMP7_1 ((uint32_t)0x00400000)
- #define ADC_SMPR2_SMP7_2 ((uint32_t)0x00800000)
- #define ADC_SMPR2_SMP8 ((uint32_t)0x07000000)
- #define ADC_SMPR2_SMP8_0 ((uint32_t)0x01000000)
- #define ADC_SMPR2_SMP8_1 ((uint32_t)0x02000000)
- #define ADC_SMPR2_SMP8_2 ((uint32_t)0x04000000)
- #define ADC_SMPR2_SMP9 ((uint32_t)0x38000000)
- #define ADC_SMPR2_SMP9_0 ((uint32_t)0x08000000)
- #define ADC_SMPR2_SMP9_1 ((uint32_t)0x10000000)
- #define ADC_SMPR2_SMP9_2 ((uint32_t)0x20000000)
- #define ADC_JOFR1_JOFFSET1 ((uint16_t)0xFFFF)
- #define ADC_JOFR2_JOFFSET2 ((uint16_t)0xFFFF)
- #define ADC_JOFR3_JOFFSET3 ((uint16_t)0xFFFF)
- #define ADC_JOFR4_JOFFSET4 ((uint16_t)0xFFFF)
- #define ADC_HTR_LT ((uint16_t)0xFFFF)
- #define ADC_LTR_LT ((uint16_t)0xFFFF)
- #define ADC_SQR1_SQ13 ((uint32_t)0x00000001F)
- #define ADC_SQR1_SQ13_0 ((uint32_t)0x00000001)
- #define ADC_SQR1_SQ13_1 ((uint32_t)0x00000002)
- #define ADC_SQR1_SQ13_2 ((uint32_t)0x00000004)

```
• #define ADC_SQR1_SQ13_3 ((uint32_t)0x00000008)
• #define ADC_SQR1_SQ13_4 ((uint32_t)0x00000010)
• #define ADC_SQR1_SQ14 ((uint32_t)0x000003E0)
• #define ADC_SQR1_SQ14_0 ((uint32_t)0x00000020)
• #define ADC_SQR1_SQ14_1 ((uint32_t)0x00000040)
• #define ADC_SQR1_SQ14_2 ((uint32_t)0x00000080)
• #define ADC_SQR1_SQ14_3 ((uint32_t)0x00000100)
• #define ADC_SQR1_SQ14_4 ((uint32_t)0x00000200)
• #define ADC_SQR1_SQ15 ((uint32_t)0x00007C00)
• #define ADC_SQR1_SQ15_0 ((uint32_t)0x00000400)
• #define ADC_SQR1_SQ15_1 ((uint32_t)0x00000800)
• #define ADC_SQR1_SQ15_2 ((uint32_t)0x00001000)
• #define ADC_SQR1_SQ15_3 ((uint32_t)0x00002000)
• #define ADC_SQR1_SQ15_4 ((uint32_t)0x00004000)
• #define ADC_SQR1_SQ16 ((uint32_t)0x000F8000)
• #define ADC_SQR1_SQ16_0 ((uint32_t)0x00008000)
• #define ADC_SQR1_SQ16_1 ((uint32_t)0x00010000)
• #define ADC_SQR1_SQ16_2 ((uint32_t)0x00020000)
• #define ADC_SQR1_SQ16_3 ((uint32_t)0x00040000)
• #define ADC_SQR1_SQ16_4 ((uint32_t)0x00080000)
• #define ADC_SQR1_L ((uint32_t)0x00F00000)
• #define ADC_SQR1_L_0 ((uint32_t)0x00100000)
• #define ADC_SQR1_L_1 ((uint32_t)0x00200000)
• #define ADC_SQR1_L_2 ((uint32_t)0x00400000)
• #define ADC_SQR1_L_3 ((uint32_t)0x00800000)
• #define ADC_SQR2_SQ7 ((uint32_t)0x0000001F)
• #define ADC_SQR2_SQ7_0 ((uint32_t)0x00000001)
• #define ADC_SQR2_SQ7_1 ((uint32_t)0x00000002)
• #define ADC_SQR2_SQ7_2 ((uint32_t)0x00000004)
• #define ADC_SQR2_SQ7_3 ((uint32_t)0x00000008)
• #define ADC_SQR2_SQ7_4 ((uint32_t)0x00000010)
• #define ADC_SQR2_SQ8 ((uint32_t)0x000003E0)
• #define ADC_SQR2_SQ8_0 ((uint32_t)0x00000020)
• #define ADC_SQR2_SQ8_1 ((uint32_t)0x00000040)
• #define ADC_SQR2_SQ8_2 ((uint32_t)0x00000080)
• #define ADC_SQR2_SQ8_3 ((uint32_t)0x00000100)
• #define ADC_SQR2_SQ8_4 ((uint32_t)0x00000200)
• #define ADC_SQR2_SQ9 ((uint32_t)0x00007C00)
• #define ADC_SQR2_SQ9_0 ((uint32_t)0x00000400)
• #define ADC_SQR2_SQ9_1 ((uint32_t)0x00000800)
• #define ADC_SQR2_SQ9_2 ((uint32_t)0x00001000)
• #define ADC_SQR2_SQ9_3 ((uint32_t)0x00002000)
• #define ADC_SQR2_SQ9_4 ((uint32_t)0x00004000)
• #define ADC_SQR2_SQ10 ((uint32_t)0x000F8000)
• #define ADC_SQR2_SQ10_0 ((uint32_t)0x00008000)
• #define ADC_SQR2_SQ10_1 ((uint32_t)0x00010000)
• #define ADC_SQR2_SQ10_2 ((uint32_t)0x00020000)
• #define ADC_SQR2_SQ10_3 ((uint32_t)0x00040000)
• #define ADC_SQR2_SQ10_4 ((uint32_t)0x00080000)
• #define ADC_SQR2_SQ11 ((uint32_t)0x01F00000)
• #define ADC_SQR2_SQ11_0 ((uint32_t)0x00100000)
• #define ADC_SQR2_SQ11_1 ((uint32_t)0x00200000)
• #define ADC_SQR2_SQ11_2 ((uint32_t)0x00400000)
• #define ADC_SQR2_SQ11_3 ((uint32_t)0x00800000)
• #define ADC_SQR2_SQ11_4 ((uint32_t)0x01000000)
```

```
• #define ADC_SQR2_SQ12 ((uint32_t)0x3E000000)
• #define ADC_SQR2_SQ12_0 ((uint32_t)0x02000000)
• #define ADC_SQR2_SQ12_1 ((uint32_t)0x04000000)
• #define ADC_SQR2_SQ12_2 ((uint32_t)0x08000000)
• #define ADC_SQR2_SQ12_3 ((uint32_t)0x10000000)
• #define ADC_SQR2_SQ12_4 ((uint32_t)0x20000000)
• #define ADC_SQR3_SQ1 ((uint32_t)0x00000001F)
• #define ADC_SQR3_SQ1_0 ((uint32_t)0x000000001)
• #define ADC_SQR3_SQ1_1 ((uint32_t)0x000000002)
• #define ADC_SQR3_SQ1_2 ((uint32_t)0x000000004)
• #define ADC_SQR3_SQ1_3 ((uint32_t)0x000000008)
• #define ADC_SQR3_SQ1_4 ((uint32_t)0x000000010)
• #define ADC_SQR3_SQ2 ((uint32_t)0x0000003E0)
• #define ADC_SQR3_SQ2_0 ((uint32_t)0x000000020)
• #define ADC_SQR3_SQ2_1 ((uint32_t)0x000000040)
• #define ADC_SQR3_SQ2_2 ((uint32_t)0x000000080)
• #define ADC_SQR3_SQ2_3 ((uint32_t)0x000000100)
• #define ADC_SQR3_SQ2_4 ((uint32_t)0x000000200)
• #define ADC_SQR3_SQ3 ((uint32_t)0x00007C00)
• #define ADC_SQR3_SQ3_0 ((uint32_t)0x000000400)
• #define ADC_SQR3_SQ3_1 ((uint32_t)0x00000800)
• #define ADC_SQR3_SQ3_2 ((uint32_t)0x00001000)
• #define ADC_SQR3_SQ3_3 ((uint32_t)0x00002000)
• #define ADC_SQR3_SQ3_4 ((uint32_t)0x00004000)
• #define ADC_SQR3_SQ4 ((uint32_t)0x000F8000)
• #define ADC_SQR3_SQ4_0 ((uint32_t)0x00008000)
• #define ADC_SQR3_SQ4_1 ((uint32_t)0x00010000)
• #define ADC_SQR3_SQ4_2 ((uint32_t)0x00020000)
• #define ADC_SQR3_SQ4_3 ((uint32_t)0x00040000)
• #define ADC_SQR3_SQ4_4 ((uint32_t)0x00080000)
• #define ADC_SQR3_SQ5 ((uint32_t)0x01F00000)
• #define ADC_SQR3_SQ5_0 ((uint32_t)0x00100000)
• #define ADC_SQR3_SQ5_1 ((uint32_t)0x00200000)
• #define ADC_SQR3_SQ5_2 ((uint32_t)0x00400000)
• #define ADC_SQR3_SQ5_3 ((uint32_t)0x00800000)
• #define ADC_SQR3_SQ5_4 ((uint32_t)0x01000000)
• #define ADC_SQR3_SQ6 ((uint32_t)0x3E000000)
• #define ADC_SQR3_SQ6_0 ((uint32_t)0x02000000)
• #define ADC_SQR3_SQ6_1 ((uint32_t)0x04000000)
• #define ADC_SQR3_SQ6_2 ((uint32_t)0x08000000)
• #define ADC_SQR3_SQ6_3 ((uint32_t)0x10000000)
• #define ADC_SQR3_SQ6_4 ((uint32_t)0x20000000)
• #define ADC_JSQR_JSQ1 ((uint32_t)0x00000001F)
• #define ADC_JSQR_JSQ1_0 ((uint32_t)0x000000001)
• #define ADC_JSQR_JSQ1_1 ((uint32_t)0x000000002)
• #define ADC_JSQR_JSQ1_2 ((uint32_t)0x000000004)
• #define ADC_JSQR_JSQ1_3 ((uint32_t)0x000000008)
• #define ADC_JSQR_JSQ1_4 ((uint32_t)0x000000010)
• #define ADC_JSQR_JSQ2 ((uint32_t)0x0000003E0)
• #define ADC_JSQR_JSQ2_0 ((uint32_t)0x000000020)
• #define ADC_JSQR_JSQ2_1 ((uint32_t)0x000000040)
• #define ADC_JSQR_JSQ2_2 ((uint32_t)0x000000080)
• #define ADC_JSQR_JSQ2_3 ((uint32_t)0x000000100)
• #define ADC_JSQR_JSQ2_4 ((uint32_t)0x000000200)
• #define ADC_JSQR_JSQ3 ((uint32_t)0x00007C00)
```

```
• #define ADC_JSQR_JSQ3_0 ((uint32_t)0x00000400)
• #define ADC_JSQR_JSQ3_1 ((uint32_t)0x00000800)
• #define ADC_JSQR_JSQ3_2 ((uint32_t)0x00001000)
• #define ADC_JSQR_JSQ3_3 ((uint32_t)0x00002000)
• #define ADC_JSQR_JSQ3_4 ((uint32_t)0x00004000)
• #define ADC_JSQR_JSQ4 ((uint32_t)0x000F8000)
• #define ADC_JSQR_JSQ4_0 ((uint32_t)0x00008000)
• #define ADC_JSQR_JSQ4_1 ((uint32_t)0x00010000)
• #define ADC_JSQR_JSQ4_2 ((uint32_t)0x00020000)
• #define ADC_JSQR_JSQ4_3 ((uint32_t)0x00040000)
• #define ADC_JSQR_JSQ4_4 ((uint32_t)0x00080000)
• #define ADC_JSQR_JL ((uint32_t)0x00300000)
• #define ADC_JSQR_JL_0 ((uint32_t)0x00100000)
• #define ADC_JSQR_JL_1 ((uint32_t)0x00200000)
• #define ADC_JDR1_JDATA ((uint16_t)0xFFFF)
• #define ADC_JDR2_JDATA ((uint16_t)0xFFFF)
• #define ADC_JDR3_JDATA ((uint16_t)0xFFFF)
• #define ADC_JDR4_JDATA ((uint16_t)0xFFFF)
• #define ADC_DR_DATA ((uint32_t)0x0000FFFF)
• #define ADC_DR_ADC2DATA ((uint32_t)0xFFFF0000)
• #define ADC_CSR_AWD1 ((uint32_t)0x00000001)
• #define ADC_CSR_EOC1 ((uint32_t)0x00000002)
• #define ADC_CSR_JEOC1 ((uint32_t)0x00000004)
• #define ADC_CSR_JSTRT1 ((uint32_t)0x00000008)
• #define ADC_CSR_STRT1 ((uint32_t)0x00000010)
• #define ADC_CSR_OVR1 ((uint32_t)0x00000020)
• #define ADC_CSR_AWD2 ((uint32_t)0x00000100)
• #define ADC_CSR_EOC2 ((uint32_t)0x00000200)
• #define ADC_CSR_JEOC2 ((uint32_t)0x00000400)
• #define ADC_CSR_JSTRT2 ((uint32_t)0x00000800)
• #define ADC_CSR_STRT2 ((uint32_t)0x00001000)
• #define ADC_CSR_OVR2 ((uint32_t)0x00002000)
• #define ADC_CSR_AWD3 ((uint32_t)0x00010000)
• #define ADC_CSR_EOC3 ((uint32_t)0x00020000)
• #define ADC_CSR_JEOC3 ((uint32_t)0x00040000)
• #define ADC_CSR_JSTRT3 ((uint32_t)0x00080000)
• #define ADC_CSR_STRT3 ((uint32_t)0x00100000)
• #define ADC_CSR_OVR3 ((uint32_t)0x00200000)
• #define ADC_CSR_DOVRR1 ADC_CSR_OVR1
• #define ADC_CSR_DOVRR2 ADC_CSR_OVR2
• #define ADC_CSR_DOVRR3 ADC_CSR_OVR3
• #define ADC_CCR_MULTI ((uint32_t)0x0000001F)
• #define ADC_CCR_MULTI_0 ((uint32_t)0x00000001)
• #define ADC_CCR_MULTI_1 ((uint32_t)0x00000002)
• #define ADC_CCR_MULTI_2 ((uint32_t)0x00000004)
• #define ADC_CCR_MULTI_3 ((uint32_t)0x00000008)
• #define ADC_CCR_MULTI_4 ((uint32_t)0x00000010)
• #define ADC_CCR_DELAY ((uint32_t)0x00000F00)
• #define ADC_CCR_DELAY_0 ((uint32_t)0x00000100)
• #define ADC_CCR_DELAY_1 ((uint32_t)0x00000200)
• #define ADC_CCR_DELAY_2 ((uint32_t)0x00000400)
• #define ADC_CCR_DELAY_3 ((uint32_t)0x00000800)
• #define ADC_CCR_DDS ((uint32_t)0x00002000)
• #define ADC_CCR_DMA ((uint32_t)0x0000C000)
• #define ADC_CCR_DMA_0 ((uint32_t)0x00004000)
```

- #define ADC_CCR_DMA_1 ((uint32_t)0x00008000)
- #define ADC_CCR_ADCPRE ((uint32_t)0x00030000)
- #define ADC_CCR_ADCPRE_0 ((uint32_t)0x00010000)
- #define ADC_CCR_ADCPRE_1 ((uint32_t)0x00020000)
- #define ADC_CCR_VBATE ((uint32_t)0x00400000)
- #define ADC_CCR_TSVREFE ((uint32_t)0x00800000)
- #define ADC_CDR_DATA1 ((uint32_t)0x0000FFFF)
- #define ADC_CDR_DATA2 ((uint32_t)0xFFFFF000)
- #define CAN_MCR_INRQ ((uint16_t)0x0001)
- #define CAN_MCR_SLEEP ((uint16_t)0x0002)
- #define CAN_MCR_TXFP ((uint16_t)0x0004)
- #define CAN_MCR_RFLM ((uint16_t)0x0008)
- #define CAN_MCR_NART ((uint16_t)0x0010)
- #define CAN_MCR_AWUM ((uint16_t)0x0020)
- #define CAN_MCR_ABOM ((uint16_t)0x0040)
- #define CAN_MCR_TTCM ((uint16_t)0x0080)
- #define CAN_MCR_RESET ((uint16_t)0x8000)
- #define CAN_MSR_INAK ((uint16_t)0x0001)
- #define CAN_MSR_SLAK ((uint16_t)0x0002)
- #define CAN_MSR_ERRI ((uint16_t)0x0004)
- #define CAN_MSR_WKUI ((uint16_t)0x0008)
- #define CAN_MSR_SLAKI ((uint16_t)0x0010)
- #define CAN_MSR_TXM ((uint16_t)0x0100)
- #define CAN_MSR_RXM ((uint16_t)0x0200)
- #define CAN_MSR_SAMP ((uint16_t)0x0400)
- #define CAN_MSR_RX ((uint16_t)0x0800)
- #define CAN_TSR_RQCP0 ((uint32_t)0x00000001)
- #define CAN_TSR_TXOK0 ((uint32_t)0x00000002)
- #define CAN_TSR_ALST0 ((uint32_t)0x00000004)
- #define CAN_TSR_TERR0 ((uint32_t)0x00000008)
- #define CAN_TSR_ABRQ0 ((uint32_t)0x00000080)
- #define CAN_TSR_RQCP1 ((uint32_t)0x00000100)
- #define CAN_TSR_TXOK1 ((uint32_t)0x00000200)
- #define CAN_TSR_ALST1 ((uint32_t)0x00000400)
- #define CAN_TSR_TERR1 ((uint32_t)0x00000800)
- #define CAN_TSR_ABRQ1 ((uint32_t)0x00008000)
- #define CAN_TSR_RQCP2 ((uint32_t)0x00010000)
- #define CAN_TSR_TXOK2 ((uint32_t)0x00020000)
- #define CAN_TSR_ALST2 ((uint32_t)0x00040000)
- #define CAN_TSR_TERR2 ((uint32_t)0x00080000)
- #define CAN_TSR_ABRQ2 ((uint32_t)0x00800000)
- #define CAN_TSR_CODE ((uint32_t)0x03000000)
- #define CAN_TSR_TME ((uint32_t)0x1C000000)
- #define CAN_TSR_TME0 ((uint32_t)0x04000000)
- #define CAN_TSR_TME1 ((uint32_t)0x08000000)
- #define CAN_TSR_TME2 ((uint32_t)0x10000000)
- #define CAN_TSR_LOW ((uint32_t)0xE0000000)
- #define CAN_TSR_LOW0 ((uint32_t)0x20000000)
- #define CAN_TSR_LOW1 ((uint32_t)0x40000000)
- #define CAN_TSR_LOW2 ((uint32_t)0x80000000)
- #define CAN_RF0R_FMP0 ((uint8_t)0x03)
- #define CAN_RF0R_FULL0 ((uint8_t)0x08)
- #define CAN_RF0R_FOVR0 ((uint8_t)0x10)
- #define CAN_RF0R_RFOM0 ((uint8_t)0x20)
- #define CAN_RF1R_FMP1 ((uint8_t)0x03)

```
• #define CAN_RF1R_FULL1 ((uint8_t)0x08)
• #define CAN_RF1R_FOVR1 ((uint8_t)0x10)
• #define CAN_RF1R_RFOM1 ((uint8_t)0x20)
• #define CAN_IER_TMEIE ((uint32_t)0x00000001)
• #define CAN_IER_FMPIE0 ((uint32_t)0x00000002)
• #define CAN_IER_FFIE0 ((uint32_t)0x00000004)
• #define CAN_IER_FOVIE0 ((uint32_t)0x00000008)
• #define CAN_IER_FMPIE1 ((uint32_t)0x00000010)
• #define CAN_IER_FFIE1 ((uint32_t)0x00000020)
• #define CAN_IER_FOVIE1 ((uint32_t)0x00000040)
• #define CAN_IER_EWGIE ((uint32_t)0x00000100)
• #define CAN_IER_EPVIE ((uint32_t)0x00000200)
• #define CAN_IER_BOFIE ((uint32_t)0x00000400)
• #define CAN_IER_LCIE ((uint32_t)0x00000800)
• #define CAN_IER_ERRIE ((uint32_t)0x00008000)
• #define CAN_IER_WKUIE ((uint32_t)0x00010000)
• #define CAN_IER_SLKIE ((uint32_t)0x00020000)
• #define CAN_ESR_EWGF ((uint32_t)0x00000001)
• #define CAN_ESR_EPVF ((uint32_t)0x00000002)
• #define CAN_ESR_BOFF ((uint32_t)0x00000004)
• #define CAN_ESR_LEC ((uint32_t)0x00000070)
• #define CAN_ESR_LEC_0 ((uint32_t)0x00000010)
• #define CAN_ESR_LEC_1 ((uint32_t)0x00000020)
• #define CAN_ESR_LEC_2 ((uint32_t)0x00000040)
• #define CAN_ESR_TEC ((uint32_t)0x00FF0000)
• #define CAN_ESR_REC ((uint32_t)0xFF000000)
• #define CAN_BTR_BRP ((uint32_t)0x000003FF)
• #define CAN_BTR_TS1 ((uint32_t)0x000F0000)
• #define CAN_BTR_TS2 ((uint32_t)0x00700000)
• #define CAN_BTR_SJW ((uint32_t)0x03000000)
• #define CAN_BTR_LBKM ((uint32_t)0x40000000)
• #define CAN_BTR_SIIM ((uint32_t)0x80000000)
• #define CAN_TI0R_TXRQ ((uint32_t)0x00000001)
• #define CAN_TI0R_RTR ((uint32_t)0x00000002)
• #define CAN_TI0R_IDE ((uint32_t)0x00000004)
• #define CAN_TI0R_EXID ((uint32_t)0x001FFFF8)
• #define CAN_TI0R_STID ((uint32_t)0xFFE00000)
• #define CAN_TDT0R_DLc ((uint32_t)0x0000000F)
• #define CAN_TDT0R_TGT ((uint32_t)0x00000100)
• #define CAN_TDT0R_TIME ((uint32_t)0xFFFF0000)
• #define CAN_TDL0R_DATA0 ((uint32_t)0x000000FF)
• #define CAN_TDL0R_DATA1 ((uint32_t)0x0000FF00)
• #define CAN_TDL0R_DATA2 ((uint32_t)0x00FF0000)
• #define CAN_TDL0R_DATA3 ((uint32_t)0xFF000000)
• #define CAN_TDHO0R_DATA4 ((uint32_t)0x000000FF)
• #define CAN_TDHO0R_DATA5 ((uint32_t)0x0000FF00)
• #define CAN_TDHO0R_DATA6 ((uint32_t)0x00FF0000)
• #define CAN_TDHO0R_DATA7 ((uint32_t)0xFF000000)
• #define CAN_TI1R_TXRQ ((uint32_t)0x00000001)
• #define CAN_TI1R_RTR ((uint32_t)0x00000002)
• #define CAN_TI1R_IDE ((uint32_t)0x00000004)
• #define CAN_TI1R_EXID ((uint32_t)0x001FFFF8)
• #define CAN_TI1R_STID ((uint32_t)0xFFE00000)
• #define CAN_TDT1R_DLc ((uint32_t)0x0000000F)
• #define CAN_TDT1R_TGT ((uint32_t)0x00000100)
```

- #define CAN_TDT1R_TIME ((uint32_t)0xFFFF0000)
- #define CAN_TDL1R_DATA0 ((uint32_t)0x000000FF)
- #define CAN_TDL1R_DATA1 ((uint32_t)0x0000FF00)
- #define CAN_TDL1R_DATA2 ((uint32_t)0x00FF0000)
- #define CAN_TDL1R_DATA3 ((uint32_t)0xFF000000)
- #define CAN_TDH1R_DATA4 ((uint32_t)0x000000FF)
- #define CAN_TDH1R_DATA5 ((uint32_t)0x0000FF00)
- #define CAN_TDH1R_DATA6 ((uint32_t)0x00FF0000)
- #define CAN_TDH1R_DATA7 ((uint32_t)0xFF000000)
- #define CAN_TI2R_TXRQ ((uint32_t)0x00000001)
- #define CAN_TI2R_RTR ((uint32_t)0x00000002)
- #define CAN_TI2R_IDE ((uint32_t)0x00000004)
- #define CAN_TI2R_EXID ((uint32_t)0x001FFFFF8)
- #define CAN_TI2R_STID ((uint32_t)0xFFE00000)
- #define CAN_TDT2R_DLC ((uint32_t)0x0000000F)
- #define CAN_TDT2R_TGT ((uint32_t)0x00000100)
- #define CAN_TDT2R_TIME ((uint32_t)0xFFFF0000)
- #define CAN_TDL2R_DATA0 ((uint32_t)0x000000FF)
- #define CAN_TDL2R_DATA1 ((uint32_t)0x0000FF00)
- #define CAN_TDL2R_DATA2 ((uint32_t)0x00FF0000)
- #define CAN_TDL2R_DATA3 ((uint32_t)0xFF000000)
- #define CAN_TDH2R_DATA4 ((uint32_t)0x000000FF)
- #define CAN_TDH2R_DATA5 ((uint32_t)0x0000FF00)
- #define CAN_TDH2R_DATA6 ((uint32_t)0x00FF0000)
- #define CAN_TDH2R_DATA7 ((uint32_t)0xFF000000)
- #define CAN_RI0R_RTR ((uint32_t)0x00000002)
- #define CAN_RI0R_IDE ((uint32_t)0x00000004)
- #define CAN_RI0R_EXID ((uint32_t)0x001FFFFF8)
- #define CAN_RI0R_STID ((uint32_t)0xFFE00000)
- #define CAN_RDT0R_DLC ((uint32_t)0x0000000F)
- #define CAN_RDT0R_FMI ((uint32_t)0x0000FF00)
- #define CAN_RDT0R_TIME ((uint32_t)0xFFFF0000)
- #define CAN_RDL0R_DATA0 ((uint32_t)0x000000FF)
- #define CAN_RDL0R_DATA1 ((uint32_t)0x0000FF00)
- #define CAN_RDL0R_DATA2 ((uint32_t)0x00FF0000)
- #define CAN_RDL0R_DATA3 ((uint32_t)0xFF000000)
- #define CAN_RDH0R_DATA4 ((uint32_t)0x000000FF)
- #define CAN_RDH0R_DATA5 ((uint32_t)0x0000FF00)
- #define CAN_RDH0R_DATA6 ((uint32_t)0x00FF0000)
- #define CAN_RDH0R_DATA7 ((uint32_t)0xFF000000)
- #define CAN_RI1R_RTR ((uint32_t)0x00000002)
- #define CAN_RI1R_IDE ((uint32_t)0x00000004)
- #define CAN_RI1R_EXID ((uint32_t)0x001FFFFF8)
- #define CAN_RI1R_STID ((uint32_t)0xFFE00000)
- #define CAN_RDT1R_DLC ((uint32_t)0x0000000F)
- #define CAN_RDT1R_FMI ((uint32_t)0x0000FF00)
- #define CAN_RDT1R_TIME ((uint32_t)0xFFFF0000)
- #define CAN_RDL1R_DATA0 ((uint32_t)0x000000FF)
- #define CAN_RDL1R_DATA1 ((uint32_t)0x0000FF00)
- #define CAN_RDL1R_DATA2 ((uint32_t)0x00FF0000)
- #define CAN_RDL1R_DATA3 ((uint32_t)0xFF000000)
- #define CAN_RDH1R_DATA4 ((uint32_t)0x000000FF)
- #define CAN_RDH1R_DATA5 ((uint32_t)0x0000FF00)
- #define CAN_RDH1R_DATA6 ((uint32_t)0x00FF0000)
- #define CAN_RDH1R_DATA7 ((uint32_t)0xFF000000)

- #define CAN_FMR_INIT ((uint8_t)0x01)
- #define CAN_FM1R_FBM ((uint16_t)0x3FFF)
- #define CAN_FM1R_FBM0 ((uint16_t)0x0001)
- #define CAN_FM1R_FBM1 ((uint16_t)0x0002)
- #define CAN_FM1R_FBM2 ((uint16_t)0x0004)
- #define CAN_FM1R_FBM3 ((uint16_t)0x0008)
- #define CAN_FM1R_FBM4 ((uint16_t)0x0010)
- #define CAN_FM1R_FBM5 ((uint16_t)0x0020)
- #define CAN_FM1R_FBM6 ((uint16_t)0x0040)
- #define CAN_FM1R_FBM7 ((uint16_t)0x0080)
- #define CAN_FM1R_FBM8 ((uint16_t)0x0100)
- #define CAN_FM1R_FBM9 ((uint16_t)0x0200)
- #define CAN_FM1R_FBM10 ((uint16_t)0x0400)
- #define CAN_FM1R_FBM11 ((uint16_t)0x0800)
- #define CAN_FM1R_FBM12 ((uint16_t)0x1000)
- #define CAN_FM1R_FBM13 ((uint16_t)0x2000)
- #define CAN_FS1R_FSC ((uint16_t)0x3FFF)
- #define CAN_FS1R_FSC0 ((uint16_t)0x0001)
- #define CAN_FS1R_FSC1 ((uint16_t)0x0002)
- #define CAN_FS1R_FSC2 ((uint16_t)0x0004)
- #define CAN_FS1R_FSC3 ((uint16_t)0x0008)
- #define CAN_FS1R_FSC4 ((uint16_t)0x0010)
- #define CAN_FS1R_FSC5 ((uint16_t)0x0020)
- #define CAN_FS1R_FSC6 ((uint16_t)0x0040)
- #define CAN_FS1R_FSC7 ((uint16_t)0x0080)
- #define CAN_FS1R_FSC8 ((uint16_t)0x0100)
- #define CAN_FS1R_FSC9 ((uint16_t)0x0200)
- #define CAN_FS1R_FSC10 ((uint16_t)0x0400)
- #define CAN_FS1R_FSC11 ((uint16_t)0x0800)
- #define CAN_FS1R_FSC12 ((uint16_t)0x1000)
- #define CAN_FS1R_FSC13 ((uint16_t)0x2000)
- #define CAN_FFA1R_FFA ((uint16_t)0x3FFF)
- #define CAN_FFA1R_FFA0 ((uint16_t)0x0001)
- #define CAN_FFA1R_FFA1 ((uint16_t)0x0002)
- #define CAN_FFA1R_FFA2 ((uint16_t)0x0004)
- #define CAN_FFA1R_FFA3 ((uint16_t)0x0008)
- #define CAN_FFA1R_FFA4 ((uint16_t)0x0010)
- #define CAN_FFA1R_FFA5 ((uint16_t)0x0020)
- #define CAN_FFA1R_FFA6 ((uint16_t)0x0040)
- #define CAN_FFA1R_FFA7 ((uint16_t)0x0080)
- #define CAN_FFA1R_FFA8 ((uint16_t)0x0100)
- #define CAN_FFA1R_FFA9 ((uint16_t)0x0200)
- #define CAN_FFA1R_FFA10 ((uint16_t)0x0400)
- #define CAN_FFA1R_FFA11 ((uint16_t)0x0800)
- #define CAN_FFA1R_FFA12 ((uint16_t)0x1000)
- #define CAN_FFA1R_FFA13 ((uint16_t)0x2000)
- #define CAN_FA1R_FACT ((uint16_t)0x3FFF)
- #define CAN_FA1R_FACT0 ((uint16_t)0x0001)
- #define CAN_FA1R_FACT1 ((uint16_t)0x0002)
- #define CAN_FA1R_FACT2 ((uint16_t)0x0004)
- #define CAN_FA1R_FACT3 ((uint16_t)0x0008)
- #define CAN_FA1R_FACT4 ((uint16_t)0x0010)
- #define CAN_FA1R_FACT5 ((uint16_t)0x0020)
- #define CAN_FA1R_FACT6 ((uint16_t)0x0040)
- #define CAN_FA1R_FACT7 ((uint16_t)0x0080)

- #define CAN_FA1R_FACT8 ((uint16_t)0x0100)
- #define CAN_FA1R_FACT9 ((uint16_t)0x0200)
- #define CAN_FA1R_FACT10 ((uint16_t)0x0400)
- #define CAN_FA1R_FACT11 ((uint16_t)0x0800)
- #define CAN_FA1R_FACT12 ((uint16_t)0x1000)
- #define CAN_FA1R_FACT13 ((uint16_t)0x2000)
- #define CAN_F0R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F0R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F0R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F0R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F0R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F0R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F0R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F0R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F0R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F0R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F0R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F0R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F0R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F0R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F0R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F0R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F0R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F0R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F0R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F0R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F0R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F0R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F0R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F0R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F0R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F0R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F0R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F0R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F0R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F0R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F0R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F0R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F1R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F1R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F1R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F1R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F1R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F1R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F1R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F1R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F1R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F1R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F1R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F1R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F1R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F1R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F1R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F1R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F1R1_FB16 ((uint32_t)0x00010000)

```
• #define CAN_F1R1_FB17 ((uint32_t)0x00020000)
• #define CAN_F1R1_FB18 ((uint32_t)0x00040000)
• #define CAN_F1R1_FB19 ((uint32_t)0x00080000)
• #define CAN_F1R1_FB20 ((uint32_t)0x00100000)
• #define CAN_F1R1_FB21 ((uint32_t)0x00200000)
• #define CAN_F1R1_FB22 ((uint32_t)0x00400000)
• #define CAN_F1R1_FB23 ((uint32_t)0x00800000)
• #define CAN_F1R1_FB24 ((uint32_t)0x01000000)
• #define CAN_F1R1_FB25 ((uint32_t)0x02000000)
• #define CAN_F1R1_FB26 ((uint32_t)0x04000000)
• #define CAN_F1R1_FB27 ((uint32_t)0x08000000)
• #define CAN_F1R1_FB28 ((uint32_t)0x10000000)
• #define CAN_F1R1_FB29 ((uint32_t)0x20000000)
• #define CAN_F1R1_FB30 ((uint32_t)0x40000000)
• #define CAN_F1R1_FB31 ((uint32_t)0x80000000)
• #define CAN_F2R1_FB0 ((uint32_t)0x00000001)
• #define CAN_F2R1_FB1 ((uint32_t)0x00000002)
• #define CAN_F2R1_FB2 ((uint32_t)0x00000004)
• #define CAN_F2R1_FB3 ((uint32_t)0x00000008)
• #define CAN_F2R1_FB4 ((uint32_t)0x00000010)
• #define CAN_F2R1_FB5 ((uint32_t)0x00000020)
• #define CAN_F2R1_FB6 ((uint32_t)0x00000040)
• #define CAN_F2R1_FB7 ((uint32_t)0x00000080)
• #define CAN_F2R1_FB8 ((uint32_t)0x00000100)
• #define CAN_F2R1_FB9 ((uint32_t)0x00000200)
• #define CAN_F2R1_FB10 ((uint32_t)0x00000400)
• #define CAN_F2R1_FB11 ((uint32_t)0x00000800)
• #define CAN_F2R1_FB12 ((uint32_t)0x00001000)
• #define CAN_F2R1_FB13 ((uint32_t)0x00002000)
• #define CAN_F2R1_FB14 ((uint32_t)0x00004000)
• #define CAN_F2R1_FB15 ((uint32_t)0x00008000)
• #define CAN_F2R1_FB16 ((uint32_t)0x00010000)
• #define CAN_F2R1_FB17 ((uint32_t)0x00020000)
• #define CAN_F2R1_FB18 ((uint32_t)0x00040000)
• #define CAN_F2R1_FB19 ((uint32_t)0x00080000)
• #define CAN_F2R1_FB20 ((uint32_t)0x00100000)
• #define CAN_F2R1_FB21 ((uint32_t)0x00200000)
• #define CAN_F2R1_FB22 ((uint32_t)0x00400000)
• #define CAN_F2R1_FB23 ((uint32_t)0x00800000)
• #define CAN_F2R1_FB24 ((uint32_t)0x01000000)
• #define CAN_F2R1_FB25 ((uint32_t)0x02000000)
• #define CAN_F2R1_FB26 ((uint32_t)0x04000000)
• #define CAN_F2R1_FB27 ((uint32_t)0x08000000)
• #define CAN_F2R1_FB28 ((uint32_t)0x10000000)
• #define CAN_F2R1_FB29 ((uint32_t)0x20000000)
• #define CAN_F2R1_FB30 ((uint32_t)0x40000000)
• #define CAN_F2R1_FB31 ((uint32_t)0x80000000)
• #define CAN_F3R1_FB0 ((uint32_t)0x00000001)
• #define CAN_F3R1_FB1 ((uint32_t)0x00000002)
• #define CAN_F3R1_FB2 ((uint32_t)0x00000004)
• #define CAN_F3R1_FB3 ((uint32_t)0x00000008)
• #define CAN_F3R1_FB4 ((uint32_t)0x00000010)
• #define CAN_F3R1_FB5 ((uint32_t)0x00000020)
• #define CAN_F3R1_FB6 ((uint32_t)0x00000040)
• #define CAN_F3R1_FB7 ((uint32_t)0x00000080)
```

- #define CAN_F3R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F3R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F3R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F3R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F3R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F3R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F3R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F3R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F3R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F3R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F3R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F3R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F3R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F3R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F3R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F3R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F3R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F3R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F3R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F3R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F3R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F3R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F3R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F3R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F4R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F4R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F4R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F4R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F4R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F4R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F4R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F4R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F4R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F4R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F4R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F4R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F4R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F4R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F4R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F4R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F4R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F4R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F4R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F4R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F4R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F4R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F4R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F4R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F4R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F4R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F4R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F4R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F4R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F4R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F4R1_FB30 ((uint32_t)0x40000000)

```
• #define CAN_F4R1_FB31 ((uint32_t)0x80000000)
• #define CAN_F5R1_FB0 ((uint32_t)0x00000001)
• #define CAN_F5R1_FB1 ((uint32_t)0x00000002)
• #define CAN_F5R1_FB2 ((uint32_t)0x00000004)
• #define CAN_F5R1_FB3 ((uint32_t)0x00000008)
• #define CAN_F5R1_FB4 ((uint32_t)0x00000010)
• #define CAN_F5R1_FB5 ((uint32_t)0x00000020)
• #define CAN_F5R1_FB6 ((uint32_t)0x00000040)
• #define CAN_F5R1_FB7 ((uint32_t)0x00000080)
• #define CAN_F5R1_FB8 ((uint32_t)0x00000100)
• #define CAN_F5R1_FB9 ((uint32_t)0x00000200)
• #define CAN_F5R1_FB10 ((uint32_t)0x00000400)
• #define CAN_F5R1_FB11 ((uint32_t)0x00000800)
• #define CAN_F5R1_FB12 ((uint32_t)0x00001000)
• #define CAN_F5R1_FB13 ((uint32_t)0x00002000)
• #define CAN_F5R1_FB14 ((uint32_t)0x00004000)
• #define CAN_F5R1_FB15 ((uint32_t)0x00008000)
• #define CAN_F5R1_FB16 ((uint32_t)0x00010000)
• #define CAN_F5R1_FB17 ((uint32_t)0x00020000)
• #define CAN_F5R1_FB18 ((uint32_t)0x00040000)
• #define CAN_F5R1_FB19 ((uint32_t)0x00080000)
• #define CAN_F5R1_FB20 ((uint32_t)0x00100000)
• #define CAN_F5R1_FB21 ((uint32_t)0x00200000)
• #define CAN_F5R1_FB22 ((uint32_t)0x00400000)
• #define CAN_F5R1_FB23 ((uint32_t)0x00800000)
• #define CAN_F5R1_FB24 ((uint32_t)0x01000000)
• #define CAN_F5R1_FB25 ((uint32_t)0x02000000)
• #define CAN_F5R1_FB26 ((uint32_t)0x04000000)
• #define CAN_F5R1_FB27 ((uint32_t)0x08000000)
• #define CAN_F5R1_FB28 ((uint32_t)0x10000000)
• #define CAN_F5R1_FB29 ((uint32_t)0x20000000)
• #define CAN_F5R1_FB30 ((uint32_t)0x40000000)
• #define CAN_F5R1_FB31 ((uint32_t)0x80000000)
• #define CAN_F6R1_FB0 ((uint32_t)0x00000001)
• #define CAN_F6R1_FB1 ((uint32_t)0x00000002)
• #define CAN_F6R1_FB2 ((uint32_t)0x00000004)
• #define CAN_F6R1_FB3 ((uint32_t)0x00000008)
• #define CAN_F6R1_FB4 ((uint32_t)0x00000010)
• #define CAN_F6R1_FB5 ((uint32_t)0x00000020)
• #define CAN_F6R1_FB6 ((uint32_t)0x00000040)
• #define CAN_F6R1_FB7 ((uint32_t)0x00000080)
• #define CAN_F6R1_FB8 ((uint32_t)0x00000100)
• #define CAN_F6R1_FB9 ((uint32_t)0x00000200)
• #define CAN_F6R1_FB10 ((uint32_t)0x00000400)
• #define CAN_F6R1_FB11 ((uint32_t)0x00000800)
• #define CAN_F6R1_FB12 ((uint32_t)0x00001000)
• #define CAN_F6R1_FB13 ((uint32_t)0x00002000)
• #define CAN_F6R1_FB14 ((uint32_t)0x00004000)
• #define CAN_F6R1_FB15 ((uint32_t)0x00008000)
• #define CAN_F6R1_FB16 ((uint32_t)0x00010000)
• #define CAN_F6R1_FB17 ((uint32_t)0x00020000)
• #define CAN_F6R1_FB18 ((uint32_t)0x00040000)
• #define CAN_F6R1_FB19 ((uint32_t)0x00080000)
• #define CAN_F6R1_FB20 ((uint32_t)0x00100000)
• #define CAN_F6R1_FB21 ((uint32_t)0x00200000)
```

```
• #define CAN_F6R1_FB22 ((uint32_t)0x00400000)
• #define CAN_F6R1_FB23 ((uint32_t)0x00800000)
• #define CAN_F6R1_FB24 ((uint32_t)0x01000000)
• #define CAN_F6R1_FB25 ((uint32_t)0x02000000)
• #define CAN_F6R1_FB26 ((uint32_t)0x04000000)
• #define CAN_F6R1_FB27 ((uint32_t)0x08000000)
• #define CAN_F6R1_FB28 ((uint32_t)0x10000000)
• #define CAN_F6R1_FB29 ((uint32_t)0x20000000)
• #define CAN_F6R1_FB30 ((uint32_t)0x40000000)
• #define CAN_F6R1_FB31 ((uint32_t)0x80000000)
• #define CAN_F7R1_FB0 ((uint32_t)0x00000001)
• #define CAN_F7R1_FB1 ((uint32_t)0x00000002)
• #define CAN_F7R1_FB2 ((uint32_t)0x00000004)
• #define CAN_F7R1_FB3 ((uint32_t)0x00000008)
• #define CAN_F7R1_FB4 ((uint32_t)0x00000010)
• #define CAN_F7R1_FB5 ((uint32_t)0x00000020)
• #define CAN_F7R1_FB6 ((uint32_t)0x00000040)
• #define CAN_F7R1_FB7 ((uint32_t)0x00000080)
• #define CAN_F7R1_FB8 ((uint32_t)0x00000100)
• #define CAN_F7R1_FB9 ((uint32_t)0x00000200)
• #define CAN_F7R1_FB10 ((uint32_t)0x00000400)
• #define CAN_F7R1_FB11 ((uint32_t)0x00000800)
• #define CAN_F7R1_FB12 ((uint32_t)0x00001000)
• #define CAN_F7R1_FB13 ((uint32_t)0x00002000)
• #define CAN_F7R1_FB14 ((uint32_t)0x00004000)
• #define CAN_F7R1_FB15 ((uint32_t)0x00008000)
• #define CAN_F7R1_FB16 ((uint32_t)0x00010000)
• #define CAN_F7R1_FB17 ((uint32_t)0x00020000)
• #define CAN_F7R1_FB18 ((uint32_t)0x00040000)
• #define CAN_F7R1_FB19 ((uint32_t)0x00080000)
• #define CAN_F7R1_FB20 ((uint32_t)0x00100000)
• #define CAN_F7R1_FB21 ((uint32_t)0x00200000)
• #define CAN_F7R1_FB22 ((uint32_t)0x00400000)
• #define CAN_F7R1_FB23 ((uint32_t)0x00800000)
• #define CAN_F7R1_FB24 ((uint32_t)0x01000000)
• #define CAN_F7R1_FB25 ((uint32_t)0x02000000)
• #define CAN_F7R1_FB26 ((uint32_t)0x04000000)
• #define CAN_F7R1_FB27 ((uint32_t)0x08000000)
• #define CAN_F7R1_FB28 ((uint32_t)0x10000000)
• #define CAN_F7R1_FB29 ((uint32_t)0x20000000)
• #define CAN_F7R1_FB30 ((uint32_t)0x40000000)
• #define CAN_F7R1_FB31 ((uint32_t)0x80000000)
• #define CAN_F8R1_FB0 ((uint32_t)0x00000001)
• #define CAN_F8R1_FB1 ((uint32_t)0x00000002)
• #define CAN_F8R1_FB2 ((uint32_t)0x00000004)
• #define CAN_F8R1_FB3 ((uint32_t)0x00000008)
• #define CAN_F8R1_FB4 ((uint32_t)0x00000010)
• #define CAN_F8R1_FB5 ((uint32_t)0x00000020)
• #define CAN_F8R1_FB6 ((uint32_t)0x00000040)
• #define CAN_F8R1_FB7 ((uint32_t)0x00000080)
• #define CAN_F8R1_FB8 ((uint32_t)0x00000100)
• #define CAN_F8R1_FB9 ((uint32_t)0x00000200)
• #define CAN_F8R1_FB10 ((uint32_t)0x00000400)
• #define CAN_F8R1_FB11 ((uint32_t)0x00000800)
• #define CAN_F8R1_FB12 ((uint32_t)0x00001000)
```

- #define CAN_F8R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F8R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F8R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F8R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F8R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F8R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F8R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F8R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F8R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F8R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F8R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F8R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F8R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F8R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F8R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F8R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F8R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F8R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F8R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F9R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F9R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F9R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F9R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F9R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F9R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F9R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F9R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F9R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F9R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F9R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F9R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F9R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F9R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F9R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F9R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F9R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F9R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F9R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F9R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F9R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F9R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F9R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F9R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F9R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F9R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F9R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F9R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F9R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F9R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F9R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F9R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F10R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F10R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F10R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F10R1_FB3 ((uint32_t)0x00000008)

- #define CAN_F10R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F10R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F10R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F10R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F10R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F10R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F10R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F10R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F10R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F10R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F10R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F10R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F10R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F10R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F10R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F10R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F10R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F10R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F10R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F10R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F10R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F10R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F10R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F10R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F10R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F10R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F10R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F10R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F11R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F11R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F11R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F11R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F11R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F11R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F11R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F11R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F11R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F11R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F11R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F11R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F11R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F11R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F11R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F11R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F11R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F11R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F11R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F11R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F11R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F11R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F11R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F11R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F11R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F11R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F11R1_FB26 ((uint32_t)0x04000000)

- #define CAN_F11R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F11R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F11R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F11R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F11R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F12R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F12R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F12R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F12R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F12R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F12R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F12R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F12R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F12R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F12R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F12R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F12R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F12R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F12R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F12R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F12R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F12R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F12R1_FB17 ((uint32_t)0x00020000)
- #define CAN_F12R1_FB18 ((uint32_t)0x00040000)
- #define CAN_F12R1_FB19 ((uint32_t)0x00080000)
- #define CAN_F12R1_FB20 ((uint32_t)0x00100000)
- #define CAN_F12R1_FB21 ((uint32_t)0x00200000)
- #define CAN_F12R1_FB22 ((uint32_t)0x00400000)
- #define CAN_F12R1_FB23 ((uint32_t)0x00800000)
- #define CAN_F12R1_FB24 ((uint32_t)0x01000000)
- #define CAN_F12R1_FB25 ((uint32_t)0x02000000)
- #define CAN_F12R1_FB26 ((uint32_t)0x04000000)
- #define CAN_F12R1_FB27 ((uint32_t)0x08000000)
- #define CAN_F12R1_FB28 ((uint32_t)0x10000000)
- #define CAN_F12R1_FB29 ((uint32_t)0x20000000)
- #define CAN_F12R1_FB30 ((uint32_t)0x40000000)
- #define CAN_F12R1_FB31 ((uint32_t)0x80000000)
- #define CAN_F13R1_FB0 ((uint32_t)0x00000001)
- #define CAN_F13R1_FB1 ((uint32_t)0x00000002)
- #define CAN_F13R1_FB2 ((uint32_t)0x00000004)
- #define CAN_F13R1_FB3 ((uint32_t)0x00000008)
- #define CAN_F13R1_FB4 ((uint32_t)0x00000010)
- #define CAN_F13R1_FB5 ((uint32_t)0x00000020)
- #define CAN_F13R1_FB6 ((uint32_t)0x00000040)
- #define CAN_F13R1_FB7 ((uint32_t)0x00000080)
- #define CAN_F13R1_FB8 ((uint32_t)0x00000100)
- #define CAN_F13R1_FB9 ((uint32_t)0x00000200)
- #define CAN_F13R1_FB10 ((uint32_t)0x00000400)
- #define CAN_F13R1_FB11 ((uint32_t)0x00000800)
- #define CAN_F13R1_FB12 ((uint32_t)0x00001000)
- #define CAN_F13R1_FB13 ((uint32_t)0x00002000)
- #define CAN_F13R1_FB14 ((uint32_t)0x00004000)
- #define CAN_F13R1_FB15 ((uint32_t)0x00008000)
- #define CAN_F13R1_FB16 ((uint32_t)0x00010000)
- #define CAN_F13R1_FB17 ((uint32_t)0x00020000)

```
• #define CAN_F13R1_FB18 ((uint32_t)0x00040000)
• #define CAN_F13R1_FB19 ((uint32_t)0x00080000)
• #define CAN_F13R1_FB20 ((uint32_t)0x00100000)
• #define CAN_F13R1_FB21 ((uint32_t)0x00200000)
• #define CAN_F13R1_FB22 ((uint32_t)0x00400000)
• #define CAN_F13R1_FB23 ((uint32_t)0x00800000)
• #define CAN_F13R1_FB24 ((uint32_t)0x01000000)
• #define CAN_F13R1_FB25 ((uint32_t)0x02000000)
• #define CAN_F13R1_FB26 ((uint32_t)0x04000000)
• #define CAN_F13R1_FB27 ((uint32_t)0x08000000)
• #define CAN_F13R1_FB28 ((uint32_t)0x10000000)
• #define CAN_F13R1_FB29 ((uint32_t)0x20000000)
• #define CAN_F13R1_FB30 ((uint32_t)0x40000000)
• #define CAN_F13R1_FB31 ((uint32_t)0x80000000)
• #define CAN_F0R2_FB0 ((uint32_t)0x00000001)
• #define CAN_F0R2_FB1 ((uint32_t)0x00000002)
• #define CAN_F0R2_FB2 ((uint32_t)0x00000004)
• #define CAN_F0R2_FB3 ((uint32_t)0x00000008)
• #define CAN_F0R2_FB4 ((uint32_t)0x00000010)
• #define CAN_F0R2_FB5 ((uint32_t)0x00000020)
• #define CAN_F0R2_FB6 ((uint32_t)0x00000040)
• #define CAN_F0R2_FB7 ((uint32_t)0x00000080)
• #define CAN_F0R2_FB8 ((uint32_t)0x00000100)
• #define CAN_F0R2_FB9 ((uint32_t)0x00000200)
• #define CAN_F0R2_FB10 ((uint32_t)0x00000400)
• #define CAN_F0R2_FB11 ((uint32_t)0x00000800)
• #define CAN_F0R2_FB12 ((uint32_t)0x00001000)
• #define CAN_F0R2_FB13 ((uint32_t)0x00002000)
• #define CAN_F0R2_FB14 ((uint32_t)0x00004000)
• #define CAN_F0R2_FB15 ((uint32_t)0x00008000)
• #define CAN_F0R2_FB16 ((uint32_t)0x00010000)
• #define CAN_F0R2_FB17 ((uint32_t)0x00020000)
• #define CAN_F0R2_FB18 ((uint32_t)0x00040000)
• #define CAN_F0R2_FB19 ((uint32_t)0x00080000)
• #define CAN_F0R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F0R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F0R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F0R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F0R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F0R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F0R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F0R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F0R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F0R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F0R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F0R2_FB31 ((uint32_t)0x80000000)
• #define CAN_F1R2_FB0 ((uint32_t)0x00000001)
• #define CAN_F1R2_FB1 ((uint32_t)0x00000002)
• #define CAN_F1R2_FB2 ((uint32_t)0x00000004)
• #define CAN_F1R2_FB3 ((uint32_t)0x00000008)
• #define CAN_F1R2_FB4 ((uint32_t)0x00000010)
• #define CAN_F1R2_FB5 ((uint32_t)0x00000020)
• #define CAN_F1R2_FB6 ((uint32_t)0x00000040)
• #define CAN_F1R2_FB7 ((uint32_t)0x00000080)
• #define CAN_F1R2_FB8 ((uint32_t)0x00000100)
```

```
• #define CAN_F1R2_FB9 ((uint32_t)0x00000200)
• #define CAN_F1R2_FB10 ((uint32_t)0x00000400)
• #define CAN_F1R2_FB11 ((uint32_t)0x00000800)
• #define CAN_F1R2_FB12 ((uint32_t)0x00001000)
• #define CAN_F1R2_FB13 ((uint32_t)0x00002000)
• #define CAN_F1R2_FB14 ((uint32_t)0x00004000)
• #define CAN_F1R2_FB15 ((uint32_t)0x00008000)
• #define CAN_F1R2_FB16 ((uint32_t)0x00010000)
• #define CAN_F1R2_FB17 ((uint32_t)0x00020000)
• #define CAN_F1R2_FB18 ((uint32_t)0x00040000)
• #define CAN_F1R2_FB19 ((uint32_t)0x00080000)
• #define CAN_F1R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F1R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F1R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F1R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F1R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F1R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F1R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F1R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F1R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F1R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F1R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F1R2_FB31 ((uint32_t)0x80000000)
• #define CAN_F2R2_FB0 ((uint32_t)0x00000001)
• #define CAN_F2R2_FB1 ((uint32_t)0x00000002)
• #define CAN_F2R2_FB2 ((uint32_t)0x00000004)
• #define CAN_F2R2_FB3 ((uint32_t)0x00000008)
• #define CAN_F2R2_FB4 ((uint32_t)0x00000010)
• #define CAN_F2R2_FB5 ((uint32_t)0x00000020)
• #define CAN_F2R2_FB6 ((uint32_t)0x00000040)
• #define CAN_F2R2_FB7 ((uint32_t)0x00000080)
• #define CAN_F2R2_FB8 ((uint32_t)0x00000100)
• #define CAN_F2R2_FB9 ((uint32_t)0x00000200)
• #define CAN_F2R2_FB10 ((uint32_t)0x00000400)
• #define CAN_F2R2_FB11 ((uint32_t)0x00000800)
• #define CAN_F2R2_FB12 ((uint32_t)0x00001000)
• #define CAN_F2R2_FB13 ((uint32_t)0x00002000)
• #define CAN_F2R2_FB14 ((uint32_t)0x00004000)
• #define CAN_F2R2_FB15 ((uint32_t)0x00008000)
• #define CAN_F2R2_FB16 ((uint32_t)0x00010000)
• #define CAN_F2R2_FB17 ((uint32_t)0x00020000)
• #define CAN_F2R2_FB18 ((uint32_t)0x00040000)
• #define CAN_F2R2_FB19 ((uint32_t)0x00080000)
• #define CAN_F2R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F2R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F2R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F2R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F2R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F2R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F2R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F2R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F2R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F2R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F2R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F2R2_FB31 ((uint32_t)0x80000000)
```

- #define CAN_F3R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F3R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F3R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F3R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F3R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F3R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F3R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F3R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F3R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F3R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F3R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F3R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F3R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F3R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F3R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F3R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F3R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F3R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F3R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F3R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F3R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F3R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F3R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F3R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F3R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F3R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F3R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F3R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F3R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F3R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F3R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F3R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F4R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F4R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F4R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F4R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F4R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F4R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F4R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F4R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F4R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F4R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F4R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F4R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F4R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F4R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F4R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F4R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F4R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F4R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F4R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F4R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F4R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F4R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F4R2_FB22 ((uint32_t)0x00400000)

```
• #define CAN_F4R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F4R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F4R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F4R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F4R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F4R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F4R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F4R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F4R2_FB31 ((uint32_t)0x80000000)
• #define CAN_F5R2_FB0 ((uint32_t)0x00000001)
• #define CAN_F5R2_FB1 ((uint32_t)0x00000002)
• #define CAN_F5R2_FB2 ((uint32_t)0x00000004)
• #define CAN_F5R2_FB3 ((uint32_t)0x00000008)
• #define CAN_F5R2_FB4 ((uint32_t)0x00000010)
• #define CAN_F5R2_FB5 ((uint32_t)0x00000020)
• #define CAN_F5R2_FB6 ((uint32_t)0x00000040)
• #define CAN_F5R2_FB7 ((uint32_t)0x00000080)
• #define CAN_F5R2_FB8 ((uint32_t)0x00000100)
• #define CAN_F5R2_FB9 ((uint32_t)0x00000200)
• #define CAN_F5R2_FB10 ((uint32_t)0x00000400)
• #define CAN_F5R2_FB11 ((uint32_t)0x00000800)
• #define CAN_F5R2_FB12 ((uint32_t)0x00001000)
• #define CAN_F5R2_FB13 ((uint32_t)0x00002000)
• #define CAN_F5R2_FB14 ((uint32_t)0x00004000)
• #define CAN_F5R2_FB15 ((uint32_t)0x00008000)
• #define CAN_F5R2_FB16 ((uint32_t)0x00010000)
• #define CAN_F5R2_FB17 ((uint32_t)0x00020000)
• #define CAN_F5R2_FB18 ((uint32_t)0x00040000)
• #define CAN_F5R2_FB19 ((uint32_t)0x00080000)
• #define CAN_F5R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F5R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F5R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F5R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F5R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F5R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F5R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F5R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F5R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F5R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F5R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F5R2_FB31 ((uint32_t)0x80000000)
• #define CAN_F6R2_FB0 ((uint32_t)0x00000001)
• #define CAN_F6R2_FB1 ((uint32_t)0x00000002)
• #define CAN_F6R2_FB2 ((uint32_t)0x00000004)
• #define CAN_F6R2_FB3 ((uint32_t)0x00000008)
• #define CAN_F6R2_FB4 ((uint32_t)0x00000010)
• #define CAN_F6R2_FB5 ((uint32_t)0x00000020)
• #define CAN_F6R2_FB6 ((uint32_t)0x00000040)
• #define CAN_F6R2_FB7 ((uint32_t)0x00000080)
• #define CAN_F6R2_FB8 ((uint32_t)0x00000100)
• #define CAN_F6R2_FB9 ((uint32_t)0x00000200)
• #define CAN_F6R2_FB10 ((uint32_t)0x00000400)
• #define CAN_F6R2_FB11 ((uint32_t)0x00000800)
• #define CAN_F6R2_FB12 ((uint32_t)0x00001000)
• #define CAN_F6R2_FB13 ((uint32_t)0x00002000)
```

```
• #define CAN_F6R2_FB14 ((uint32_t)0x00004000)
• #define CAN_F6R2_FB15 ((uint32_t)0x00008000)
• #define CAN_F6R2_FB16 ((uint32_t)0x00010000)
• #define CAN_F6R2_FB17 ((uint32_t)0x00020000)
• #define CAN_F6R2_FB18 ((uint32_t)0x00040000)
• #define CAN_F6R2_FB19 ((uint32_t)0x00080000)
• #define CAN_F6R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F6R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F6R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F6R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F6R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F6R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F6R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F6R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F6R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F6R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F6R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F6R2_FB31 ((uint32_t)0x80000000)
• #define CAN_F7R2_FB0 ((uint32_t)0x00000001)
• #define CAN_F7R2_FB1 ((uint32_t)0x00000002)
• #define CAN_F7R2_FB2 ((uint32_t)0x00000004)
• #define CAN_F7R2_FB3 ((uint32_t)0x00000008)
• #define CAN_F7R2_FB4 ((uint32_t)0x00000010)
• #define CAN_F7R2_FB5 ((uint32_t)0x00000020)
• #define CAN_F7R2_FB6 ((uint32_t)0x00000040)
• #define CAN_F7R2_FB7 ((uint32_t)0x00000080)
• #define CAN_F7R2_FB8 ((uint32_t)0x00000100)
• #define CAN_F7R2_FB9 ((uint32_t)0x00000200)
• #define CAN_F7R2_FB10 ((uint32_t)0x00000400)
• #define CAN_F7R2_FB11 ((uint32_t)0x00000800)
• #define CAN_F7R2_FB12 ((uint32_t)0x00001000)
• #define CAN_F7R2_FB13 ((uint32_t)0x00002000)
• #define CAN_F7R2_FB14 ((uint32_t)0x00004000)
• #define CAN_F7R2_FB15 ((uint32_t)0x00008000)
• #define CAN_F7R2_FB16 ((uint32_t)0x00010000)
• #define CAN_F7R2_FB17 ((uint32_t)0x00020000)
• #define CAN_F7R2_FB18 ((uint32_t)0x00040000)
• #define CAN_F7R2_FB19 ((uint32_t)0x00080000)
• #define CAN_F7R2_FB20 ((uint32_t)0x00100000)
• #define CAN_F7R2_FB21 ((uint32_t)0x00200000)
• #define CAN_F7R2_FB22 ((uint32_t)0x00400000)
• #define CAN_F7R2_FB23 ((uint32_t)0x00800000)
• #define CAN_F7R2_FB24 ((uint32_t)0x01000000)
• #define CAN_F7R2_FB25 ((uint32_t)0x02000000)
• #define CAN_F7R2_FB26 ((uint32_t)0x04000000)
• #define CAN_F7R2_FB27 ((uint32_t)0x08000000)
• #define CAN_F7R2_FB28 ((uint32_t)0x10000000)
• #define CAN_F7R2_FB29 ((uint32_t)0x20000000)
• #define CAN_F7R2_FB30 ((uint32_t)0x40000000)
• #define CAN_F7R2_FB31 ((uint32_t)0x80000000)
• #define CAN_F8R2_FB0 ((uint32_t)0x00000001)
• #define CAN_F8R2_FB1 ((uint32_t)0x00000002)
• #define CAN_F8R2_FB2 ((uint32_t)0x00000004)
• #define CAN_F8R2_FB3 ((uint32_t)0x00000008)
• #define CAN_F8R2_FB4 ((uint32_t)0x00000010)
```

- #define CAN_F8R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F8R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F8R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F8R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F8R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F8R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F8R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F8R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F8R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F8R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F8R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F8R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F8R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F8R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F8R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F8R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F8R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F8R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F8R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F8R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F8R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F8R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F8R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F8R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F8R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F8R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F8R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F9R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F9R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F9R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F9R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F9R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F9R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F9R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F9R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F9R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F9R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F9R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F9R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F9R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F9R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F9R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F9R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F9R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F9R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F9R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F9R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F9R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F9R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F9R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F9R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F9R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F9R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F9R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F9R2_FB27 ((uint32_t)0x08000000)

- #define CAN_F9R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F9R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F9R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F9R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F10R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F10R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F10R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F10R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F10R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F10R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F10R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F10R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F10R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F10R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F10R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F10R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F10R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F10R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F10R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F10R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F10R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F10R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F10R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F10R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F10R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F10R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F10R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F10R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F10R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F10R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F10R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F10R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F10R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F10R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F10R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F10R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F11R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F11R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F11R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F11R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F11R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F11R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F11R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F11R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F11R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F11R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F11R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F11R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F11R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F11R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F11R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F11R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F11R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F11R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F11R2_FB18 ((uint32_t)0x00040000)

- #define CAN_F11R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F11R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F11R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F11R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F11R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F11R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F11R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F11R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F11R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F11R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F11R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F11R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F11R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F12R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F12R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F12R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F12R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F12R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F12R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F12R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F12R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F12R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F12R2_FB9 ((uint32_t)0x00000200)
- #define CAN_F12R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F12R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F12R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F12R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F12R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F12R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F12R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F12R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F12R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F12R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F12R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F12R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F12R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F12R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F12R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F12R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F12R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F12R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F12R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F12R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F12R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F12R2_FB31 ((uint32_t)0x80000000)
- #define CAN_F13R2_FB0 ((uint32_t)0x00000001)
- #define CAN_F13R2_FB1 ((uint32_t)0x00000002)
- #define CAN_F13R2_FB2 ((uint32_t)0x00000004)
- #define CAN_F13R2_FB3 ((uint32_t)0x00000008)
- #define CAN_F13R2_FB4 ((uint32_t)0x00000010)
- #define CAN_F13R2_FB5 ((uint32_t)0x00000020)
- #define CAN_F13R2_FB6 ((uint32_t)0x00000040)
- #define CAN_F13R2_FB7 ((uint32_t)0x00000080)
- #define CAN_F13R2_FB8 ((uint32_t)0x00000100)
- #define CAN_F13R2_FB9 ((uint32_t)0x00000200)

- #define CAN_F13R2_FB10 ((uint32_t)0x00000400)
- #define CAN_F13R2_FB11 ((uint32_t)0x00000800)
- #define CAN_F13R2_FB12 ((uint32_t)0x00001000)
- #define CAN_F13R2_FB13 ((uint32_t)0x00002000)
- #define CAN_F13R2_FB14 ((uint32_t)0x00004000)
- #define CAN_F13R2_FB15 ((uint32_t)0x00008000)
- #define CAN_F13R2_FB16 ((uint32_t)0x00010000)
- #define CAN_F13R2_FB17 ((uint32_t)0x00020000)
- #define CAN_F13R2_FB18 ((uint32_t)0x00040000)
- #define CAN_F13R2_FB19 ((uint32_t)0x00080000)
- #define CAN_F13R2_FB20 ((uint32_t)0x00100000)
- #define CAN_F13R2_FB21 ((uint32_t)0x00200000)
- #define CAN_F13R2_FB22 ((uint32_t)0x00400000)
- #define CAN_F13R2_FB23 ((uint32_t)0x00800000)
- #define CAN_F13R2_FB24 ((uint32_t)0x01000000)
- #define CAN_F13R2_FB25 ((uint32_t)0x02000000)
- #define CAN_F13R2_FB26 ((uint32_t)0x04000000)
- #define CAN_F13R2_FB27 ((uint32_t)0x08000000)
- #define CAN_F13R2_FB28 ((uint32_t)0x10000000)
- #define CAN_F13R2_FB29 ((uint32_t)0x20000000)
- #define CAN_F13R2_FB30 ((uint32_t)0x40000000)
- #define CAN_F13R2_FB31 ((uint32_t)0x80000000)
- #define CRC_DR_DR ((uint32_t)0xFFFFFFFF)
- #define CRC_IDR_IDR ((uint8_t)0xFF)
- #define CRYP_CR_RESET ((uint8_t)0x01)
- #define CRYP_CR_ALGODIR ((uint32_t)0x00000004)
- #define CRYP_CR_ALGOMODE ((uint32_t)0x00080038)
- #define CRYP_CR_ALGOMODE_0 ((uint32_t)0x00000008)
- #define CRYP_CR_ALGOMODE_1 ((uint32_t)0x00000010)
- #define CRYP_CR_ALGOMODE_2 ((uint32_t)0x00000020)
- #define CRYP_CR_ALGOMODE_TDES_ECB ((uint32_t)0x00000000)
- #define CRYP_CR_ALGOMODE_TDES_CBC ((uint32_t)0x00000008)
- #define CRYP_CR_ALGOMODE_DES_ECB ((uint32_t)0x00000010)
- #define CRYP_CR_ALGOMODE_DES_CBC ((uint32_t)0x00000018)
- #define CRYP_CR_ALGOMODE_AES_ECB ((uint32_t)0x00000020)
- #define CRYP_CR_ALGOMODE_AES_CBC ((uint32_t)0x00000028)
- #define CRYP_CR_ALGOMODE_AES_CTR ((uint32_t)0x00000030)
- #define CRYP_CR_ALGOMODE_AES_KEY ((uint32_t)0x00000038)
- #define CRYP_CR_DATATYPE ((uint32_t)0x000000C0)
- #define CRYP_CR_DATATYPE_0 ((uint32_t)0x00000040)
- #define CRYP_CR_DATATYPE_1 ((uint32_t)0x00000080)
- #define CRYP_CR_KEYSIZE ((uint32_t)0x00000300)
- #define CRYP_CR_KEYSIZE_0 ((uint32_t)0x00000100)
- #define CRYP_CR_KEYSIZE_1 ((uint32_t)0x00000200)
- #define CRYP_CR_FFLUSH ((uint32_t)0x00004000)
- #define CRYP_CR_CRYPTEN ((uint32_t)0x00008000)
- #define CRYP_CR_GCM_CCMPH ((uint32_t)0x00030000)
- #define CRYP_CR_GCM_CCMPH_0 ((uint32_t)0x00010000)
- #define CRYP_CR_GCM_CCMPH_1 ((uint32_t)0x00020000)
- #define CRYP_CR_ALGOMODE_3 ((uint32_t)0x00080000)
- #define CRYP_SR_IFEM ((uint32_t)0x00000001)
- #define CRYP_SR_IFNF ((uint32_t)0x00000002)
- #define CRYP_SR_OFNE ((uint32_t)0x00000004)
- #define CRYP_SR_OFFU ((uint32_t)0x00000008)
- #define CRYP_SR_BUSY ((uint32_t)0x00000010)

- #define **CRYP_DMACR_DIEN** ((uint32_t)0x00000001)
- #define **CRYP_DMACR_DOEN** ((uint32_t)0x00000002)
- #define **CRYP_IMSCR_INIM** ((uint32_t)0x00000001)
- #define **CRYP_IMSCR_OUTIM** ((uint32_t)0x00000002)
- #define **CRYP_RISR_OUTRIS** ((uint32_t)0x00000001)
- #define **CRYP_RISR_INRIS** ((uint32_t)0x00000002)
- #define **CRYP_MISR_INMIS** ((uint32_t)0x00000001)
- #define **CRYP_MISR_OUTMIS** ((uint32_t)0x00000002)
- #define **DAC_CR_EN1** ((uint32_t)0x00000001)
- #define **DAC_CR_BOFF1** ((uint32_t)0x00000002)
- #define **DAC_CR_TEN1** ((uint32_t)0x00000004)
- #define **DAC_CR_TSEL1** ((uint32_t)0x00000038)
- #define **DAC_CR_TSEL1_0** ((uint32_t)0x00000008)
- #define **DAC_CR_TSEL1_1** ((uint32_t)0x00000010)
- #define **DAC_CR_TSEL1_2** ((uint32_t)0x00000020)
- #define **DAC_CR_WAVE1** ((uint32_t)0x000000C0)
- #define **DAC_CR_WAVE1_0** ((uint32_t)0x00000040)
- #define **DAC_CR_WAVE1_1** ((uint32_t)0x00000080)
- #define **DAC_CR_MAMP1** ((uint32_t)0x00000F00)
- #define **DAC_CR_MAMP1_0** ((uint32_t)0x00000100)
- #define **DAC_CR_MAMP1_1** ((uint32_t)0x00000200)
- #define **DAC_CR_MAMP1_2** ((uint32_t)0x00000400)
- #define **DAC_CR_MAMP1_3** ((uint32_t)0x00000800)
- #define **DAC_CR_DMAEN1** ((uint32_t)0x00001000)
- #define **DAC_CR_DMAUDRIE1** ((uint32_t)0x00002000)
- #define **DAC_CR_EN2** ((uint32_t)0x00010000)
- #define **DAC_CR_BOFF2** ((uint32_t)0x00020000)
- #define **DAC_CR_TEN2** ((uint32_t)0x00040000)
- #define **DAC_CR_TSEL2** ((uint32_t)0x00380000)
- #define **DAC_CR_TSEL2_0** ((uint32_t)0x00080000)
- #define **DAC_CR_TSEL2_1** ((uint32_t)0x00100000)
- #define **DAC_CR_TSEL2_2** ((uint32_t)0x00200000)
- #define **DAC_CR_WAVE2** ((uint32_t)0x00C00000)
- #define **DAC_CR_WAVE2_0** ((uint32_t)0x00400000)
- #define **DAC_CR_WAVE2_1** ((uint32_t)0x00800000)
- #define **DAC_CR_MAMP2** ((uint32_t)0x0F000000)
- #define **DAC_CR_MAMP2_0** ((uint32_t)0x01000000)
- #define **DAC_CR_MAMP2_1** ((uint32_t)0x02000000)
- #define **DAC_CR_MAMP2_2** ((uint32_t)0x04000000)
- #define **DAC_CR_MAMP2_3** ((uint32_t)0x08000000)
- #define **DAC_CR_DMAEN2** ((uint32_t)0x10000000)
- #define **DAC_CR_DMAUDRIE2** ((uint32_t)0x20000000U)
- #define **DAC_SWTRIGR_SWTRIG1** ((uint8_t)0x01)
- #define **DAC_SWTRIGR_SWTRIG2** ((uint8_t)0x02)
- #define **DAC_DHR12R1_DACC1DHR** ((uint16_t)0xFFFF)
- #define **DAC_DHR12L1_DACC1DHR** ((uint16_t)0xFFFF0)
- #define **DAC_DHR8R1_DACC1DHR** ((uint8_t)0xFF)
- #define **DAC_DHR12R2_DACC2DHR** ((uint16_t)0xFFFF)
- #define **DAC_DHR12L2_DACC2DHR** ((uint16_t)0xFFFF0)
- #define **DAC_DHR8R2_DACC2DHR** ((uint8_t)0xFF)
- #define **DAC_DHR12RD_DACC1DHR** ((uint32_t)0x00000FFF)
- #define **DAC_DHR12RD_DACC2DHR** ((uint32_t)0x0FFF0000)
- #define **DAC_DHR12LD_DACC1DHR** ((uint32_t)0x0000FFF0)
- #define **DAC_DHR12LD_DACC2DHR** ((uint32_t)0xFFFF0000)
- #define **DAC_DHR8RD_DACC1DHR** ((uint16_t)0x00FF)

- #define **DAC_DHR8RD_DACC2DHR** ((uint16_t)0xFF00)
- #define **DAC_DOR1_DACC1DOR** ((uint16_t)0xFFFF)
- #define **DAC_DOR2_DACC2DOR** ((uint16_t)0xFFFF)
- #define **DAC_SR_DMAUDR1** ((uint32_t)0x00002000)
- #define **DAC_SR_DMAUDR2** ((uint32_t)0x20000000)
- #define **DCMI_CR_CAPTURE** ((uint32_t)0x00000001)
- #define **DCMI_CR_CM** ((uint32_t)0x00000002)
- #define **DCMI_CR_CROP** ((uint32_t)0x00000004)
- #define **DCMI_CR_JPEG** ((uint32_t)0x00000008)
- #define **DCMI_CR_ESS** ((uint32_t)0x00000010)
- #define **DCMI_CR_PCKPOL** ((uint32_t)0x00000020)
- #define **DCMI_CR_HSPOL** ((uint32_t)0x00000040)
- #define **DCMI_CR_VSPOL** ((uint32_t)0x00000080)
- #define **DCMI_CR_FCRC_0** ((uint32_t)0x00000100)
- #define **DCMI_CR_FCRC_1** ((uint32_t)0x00000200)
- #define **DCMI_CR_EDM_0** ((uint32_t)0x00000400)
- #define **DCMI_CR_EDM_1** ((uint32_t)0x00000800)
- #define **DCMI_CR_CRE** ((uint32_t)0x00001000)
- #define **DCMI_CR_ENABLE** ((uint32_t)0x00004000)
- #define **DCMI_SR_HSYNC** ((uint32_t)0x00000001)
- #define **DCMI_SR_VSYNC** ((uint32_t)0x00000002)
- #define **DCMI_SR_FNE** ((uint32_t)0x00000004)
- #define **DCMI_RIS_FRAME_RIS** ((uint32_t)0x00000001)
- #define **DCMI_RIS_OVR_RIS** ((uint32_t)0x00000002)
- #define **DCMI_RIS_ERR_RIS** ((uint32_t)0x00000004)
- #define **DCMI_RIS_VSYNC_RIS** ((uint32_t)0x00000008)
- #define **DCMI_RIS_LINE_RIS** ((uint32_t)0x00000010)
- #define **DCMI_RISR_FRAME_RIS** DCMI_RIS_FRAME_RIS
- #define **DCMI_RISR_OVR_RIS** DCMI_RIS_OVR_RIS
- #define **DCMI_RISR_ERR_RIS** DCMI_RIS_ERR_RIS
- #define **DCMI_RISR_VSYNC_RIS** DCMI_RIS_VSYNC_RIS
- #define **DCMI_RISR_LINE_RIS** DCMI_RIS_LINE_RIS
- #define **DCMI_RISR_OVF_RIS** DCMI_RIS_OVR_RIS
- #define **DCMI_IER_FRAME_IE** ((uint32_t)0x00000001)
- #define **DCMI_IER_OVR_IE** ((uint32_t)0x00000002)
- #define **DCMI_IER_ERR_IE** ((uint32_t)0x00000004)
- #define **DCMI_IER_VSYNC_IE** ((uint32_t)0x00000008)
- #define **DCMI_IER_LINE_IE** ((uint32_t)0x00000010)
- #define **DCMI_IER_OVF_IE** DCMI_IER_OVR_IE
- #define **DCMI_MIS_FRAME_MIS** ((uint32_t)0x00000001)
- #define **DCMI_MIS_OVR_MIS** ((uint32_t)0x00000002)
- #define **DCMI_MIS_ERR_MIS** ((uint32_t)0x00000004)
- #define **DCMI_MIS_VSYNC_MIS** ((uint32_t)0x00000008)
- #define **DCMI_MIS_LINE_MIS** ((uint32_t)0x00000010)
- #define **DCMI_MISR_FRAME_MIS** DCMI_MIS_FRAME_MIS
- #define **DCMI_MISR_OVF_MIS** DCMI_MIS_OVR_MIS
- #define **DCMI_MISR_ERR_MIS** DCMI_MIS_ERR_MIS
- #define **DCMI_MISR_VSYNC_MIS** DCMI_MIS_VSYNC_MIS
- #define **DCMI_MISR_LINE_MIS** DCMI_MIS_LINE_MIS
- #define **DCMI_ICR_FRAME_ISC** ((uint32_t)0x00000001)
- #define **DCMI_ICR_OVR_ISC** ((uint32_t)0x00000002)
- #define **DCMI_ICR_ERR_ISC** ((uint32_t)0x00000004)
- #define **DCMI_ICR_VSYNC_ISC** ((uint32_t)0x00000008)
- #define **DCMI_ICR_LINE_ISC** ((uint32_t)0x00000010)
- #define **DCMI_ICR_OVF_ISC** DCMI_ICR_OVR_ISC

```
• #define DCMI_ESCR_FSC ((uint32_t)0x000000FF)
• #define DCMI_ESCR_LSC ((uint32_t)0x0000FF00)
• #define DCMI_ESCR_LEC ((uint32_t)0x00FF0000)
• #define DCMI_ESCR_FEC ((uint32_t)0xFF000000)
• #define DCMI_ESUR_FSU ((uint32_t)0x000000FF)
• #define DCMI_ESUR_LSU ((uint32_t)0x0000FF00)
• #define DCMI_ESUR_LEU ((uint32_t)0x00FF0000)
• #define DCMI_ESUR_FEU ((uint32_t)0xFF000000)
• #define DCMI_CWSTRT_HOFFCNT ((uint32_t)0x00003FFF)
• #define DCMI_CWSTRT_VST ((uint32_t)0x1FFF0000)
• #define DCMI_CWSIZE_CAPCNT ((uint32_t)0x00003FFF)
• #define DCMI_CWSIZE_VLINE ((uint32_t)0x3FFF0000)
• #define DCMI_DR_BYTE0 ((uint32_t)0x000000FF)
• #define DCMI_DR_BYTE1 ((uint32_t)0x0000FF00)
• #define DCMI_DR_BYTE2 ((uint32_t)0x00FF0000)
• #define DCMI_DR_BYTE3 ((uint32_t)0xFF000000)
• #define DFSDM_CHCFG1_DFSDMEN ((uint32_t)0x80000000)
• #define DFSDM_CHCFG1_CKOUTSRC ((uint32_t)0x40000000)
• #define DFSDM_CHCFG1_CKOUTDIV ((uint32_t)0x00FF0000)
• #define DFSDM_CHCFG1_DATPACK ((uint32_t)0x0000C000)
• #define DFSDM_CHCFG1_DATPACK_1 ((uint32_t)0x00008000)
• #define DFSDM_CHCFG1_DATPACK_0 ((uint32_t)0x00004000)
• #define DFSDM_CHCFG1_DATMPX ((uint32_t)0x00003000)
• #define DFSDM_CHCFG1_DATMPX_1 ((uint32_t)0x00002000)
• #define DFSDM_CHCFG1_DATMPX_0 ((uint32_t)0x00001000)
• #define DFSDM_CHCFG1_CHINSEL ((uint32_t)0x00000100)
• #define DFSDM_CHCFG1_CHEN ((uint32_t)0x00000080)
• #define DFSDM_CHCFG1_CKABEN ((uint32_t)0x00000040)
• #define DFSDM_CHCFG1_SCDEN ((uint32_t)0x00000020)
• #define DFSDM_CHCFG1_SPICKSEL ((uint32_t)0x0000000C)
• #define DFSDM_CHCFG1_SPICKSEL_1 ((uint32_t)0x00000008)
• #define DFSDM_CHCFG1_SPICKSEL_0 ((uint32_t)0x00000004)
• #define DFSDM_CHCFG1_SITP ((uint32_t)0x00000003)
• #define DFSDM_CHCFG1_SITP_1 ((uint32_t)0x00000002)
• #define DFSDM_CHCFG1_SITP_0 ((uint32_t)0x00000001)
• #define DFSDM_CHCFG2_OFFSET ((uint32_t)0xFFFFFFF0)
• #define DFSDM_CHCFG2_DTRBS ((uint32_t)0x000000F8)
• #define DFSDM_CHAWSCDR_AWFORD ((uint32_t)0x00C00000)
• #define DFSDM_CHAWSCDR_AWFORD_1 ((uint32_t)0x00800000)
• #define DFSDM_CHAWSCDR_AWFORD_0 ((uint32_t)0x00400000)
• #define DFSDM_CHAWSCDR_AWFOSR ((uint32_t)0x001F0000)
• #define DFSDM_CHAWSCDR_BKSCD ((uint32_t)0x0000F000)
• #define DFSDM_CHAWSCDR_SCDT ((uint32_t)0x000000FF)
• #define DFSDM_CHWDATR_WDATA ((uint32_t)0x0000FFFF)
• #define DFSDM_CHDATINR_INDATA0 ((uint32_t)0x0000FFFF)
• #define DFSDM_CHDATINR_INDATA1 ((uint32_t)0xFFFF0000)
• #define DFSDM_FLTCR1_AWFSEL ((uint32_t)0x40000000)
• #define DFSDM_FLTCR1_FAST ((uint32_t)0x20000000)
• #define DFSDM_FLTCR1_RCH ((uint32_t)0x07000000)
• #define DFSDM_FLTCR1_RDMAEN ((uint32_t)0x00200000)
• #define DFSDM_FLTCR1_RSYNC ((uint32_t)0x00080000)
• #define DFSDM_FLTCR1_RCONT ((uint32_t)0x00040000)
• #define DFSDM_FLTCR1_RSWSTART ((uint32_t)0x00020000)
• #define DFSDM_FLTCR1_JEXTEN ((uint32_t)0x00006000)
• #define DFSDM_FLTCR1_JEXTEN_1 ((uint32_t)0x00004000)
```

```
• #define DFSDM_FLTCR1_JEXTEN_0 ((uint32_t)0x00002000)
• #define DFSDM_FLTCR1_JEXTSEL ((uint32_t)0x00000700)
• #define DFSDM_FLTCR1_JEXTSEL_2 ((uint32_t)0x00000400)
• #define DFSDM_FLTCR1_JEXTSEL_1 ((uint32_t)0x00000200)
• #define DFSDM_FLTCR1_JEXTSEL_0 ((uint32_t)0x00000100)
• #define DFSDM_FLTCR1_JDMAEN ((uint32_t)0x00000020)
• #define DFSDM_FLTCR1_JSCAN ((uint32_t)0x00000010)
• #define DFSDM_FLTCR1_JSYNC ((uint32_t)0x00000008)
• #define DFSDM_FLTCR1_JSWSTART ((uint32_t)0x00000002)
• #define DFSDM_FLTCR1_DFEN ((uint32_t)0x00000001)
• #define DFSDM_FLTCR2_AWDCH ((uint32_t)0x000F0000)
• #define DFSDM_FLTCR2_EXCH ((uint32_t)0x00000F00)
• #define DFSDM_FLTCR2_CKABIE ((uint32_t)0x00000040)
• #define DFSDM_FLTCR2_SCDIE ((uint32_t)0x00000020)
• #define DFSDM_FLTCR2_AWDIE ((uint32_t)0x00000010)
• #define DFSDM_FLTCR2_ROVRIE ((uint32_t)0x00000008)
• #define DFSDM_FLTCR2_JOVRIE ((uint32_t)0x00000004)
• #define DFSDM_FLTCR2_REOCIE ((uint32_t)0x00000002)
• #define DFSDM_FLTCR2_JEOCIE ((uint32_t)0x00000001)
• #define DFSDM_FLTISR_SCDF ((uint32_t)0x0F000000)
• #define DFSDM_FLTISR_CKABF ((uint32_t)0x000F0000)
• #define DFSDM_FLTISR_RCIP ((uint32_t)0x00004000)
• #define DFSDM_FLTISR_JCIP ((uint32_t)0x00002000)
• #define DFSDM_FLTISR_AWDF ((uint32_t)0x00000010)
• #define DFSDM_FLTISR_ROVRF ((uint32_t)0x00000008)
• #define DFSDM_FLTISR_JOVRF ((uint32_t)0x00000004)
• #define DFSDM_FLTISR_REOCF ((uint32_t)0x00000002)
• #define DFSDM_FLTISR_JEOCF ((uint32_t)0x00000001)
• #define DFSDM_FLTICR_CLRSCSDF ((uint32_t)0x0F000000)
• #define DFSDM_FLTICR_CLRCKABF ((uint32_t)0x000F0000)
• #define DFSDM_FLTICR_CLRROVRF ((uint32_t)0x00000008)
• #define DFSDM_FLTICR_CLRJOVRF ((uint32_t)0x00000004)
• #define DFSDM_FLTJCHGR_JCHG ((uint32_t)0x000000FF)
• #define DFSDM_FLTFCR_FORD ((uint32_t)0xE0000000)
• #define DFSDM_FLTFCR_FORD_2 ((uint32_t)0x80000000)
• #define DFSDM_FLTFCR_FORD_1 ((uint32_t)0x40000000)
• #define DFSDM_FLTFCR_FORD_0 ((uint32_t)0x20000000)
• #define DFSDM_FLTFCR_FOSR ((uint32_t)0x03FF0000)
• #define DFSDM_FLTFCR_IOSR ((uint32_t)0x0000000FF)
• #define DFSDM_FLTJDATAR_JDATA ((uint32_t)0xFFFFFFF00)
• #define DFSDM_FLTJDATAR_JDATACH ((uint32_t)0x000000007)
• #define DFSDM_FLTRDATAR_RDATA ((uint32_t)0xFFFFFFF00)
• #define DFSDM_FLTRDATAR_RPEND ((uint32_t)0x000000010)
• #define DFSDM_FLTRDATAR_RDATAANCH ((uint32_t)0x000000007)
• #define DFSDM_FLTAWHTR_AWHT ((uint32_t)0xFFFFFFF00)
• #define DFSDM_FLTAWHTR_BKAWH ((uint32_t)0x0000000F)
• #define DFSDM_FLTAWLTR_AWLTL ((uint32_t)0xFFFFFFF00)
• #define DFSDM_FLTAWLTR_BKAWL ((uint32_t)0x0000000F)
• #define DFSDM_FLTAWSR_AWHTF ((uint32_t)0x00000F00)
• #define DFSDM_FLTAWSR_AWLTF ((uint32_t)0x0000000F)
• #define DFSDM_FLTAWCFR_CLRAWHTF ((uint32_t)0x000000F00)
• #define DFSDM_FLTAWCFR_CLRAWLTF ((uint32_t)0x0000000F)
• #define DFSDM_FLTEXMAX_EXMAX ((uint32_t)0xFFFFFFF00)
• #define DFSDM_FLTEXMAX_EXMAXCH ((uint32_t)0x00000007)
• #define DFSDM_FLTEXMIN_EXMIN ((uint32_t)0xFFFFFFF00)
```

- #define **DFSDM_FLTEXMIN_EXMINCH** ((uint32_t)0x00000007)
- #define **DFSDM_FLTCNVTIMR_CNVCNT** ((uint32_t)0xFFFFFFF0)
- #define **DMA_SxCR_CHSEL** ((uint32_t)0xE0000000)
- #define **DMA_SxCR_CHSEL_0** ((uint32_t)0x02000000)
- #define **DMA_SxCR_CHSEL_1** ((uint32_t)0x04000000)
- #define **DMA_SxCR_CHSEL_2** ((uint32_t)0x08000000)
- #define **DMA_SxCR_MBURST** ((uint32_t)0x01800000)
- #define **DMA_SxCR_MBURST_0** ((uint32_t)0x00800000)
- #define **DMA_SxCR_MBURST_1** ((uint32_t)0x01000000)
- #define **DMA_SxCR_PBURST** ((uint32_t)0x00600000)
- #define **DMA_SxCR_PBURST_0** ((uint32_t)0x00200000)
- #define **DMA_SxCR_PBURST_1** ((uint32_t)0x00400000)
- #define **DMA_SxCR_ACK** ((uint32_t)0x00100000)
- #define **DMA_SxCR_CT** ((uint32_t)0x00080000)
- #define **DMA_SxCR_DBM** ((uint32_t)0x00040000)
- #define **DMA_SxCR_PL** ((uint32_t)0x00030000)
- #define **DMA_SxCR_PL_0** ((uint32_t)0x00010000)
- #define **DMA_SxCR_PL_1** ((uint32_t)0x00020000)
- #define **DMA_SxCR_PINCOS** ((uint32_t)0x00008000)
- #define **DMA_SxCR_MSIZE** ((uint32_t)0x00006000)
- #define **DMA_SxCR_MSIZE_0** ((uint32_t)0x00002000)
- #define **DMA_SxCR_MSIZE_1** ((uint32_t)0x00004000)
- #define **DMA_SxCR_PSIZE** ((uint32_t)0x00001800)
- #define **DMA_SxCR_PSIZE_0** ((uint32_t)0x00000800)
- #define **DMA_SxCR_PSIZE_1** ((uint32_t)0x00001000)
- #define **DMA_SxCR_MINC** ((uint32_t)0x0000400)
- #define **DMA_SxCR_PINC** ((uint32_t)0x00000200)
- #define **DMA_SxCR_CIRC** ((uint32_t)0x00000100)
- #define **DMA_SxCR_DIR** ((uint32_t)0x000000C0)
- #define **DMA_SxCR_DIR_0** ((uint32_t)0x00000040)
- #define **DMA_SxCR_DIR_1** ((uint32_t)0x00000080)
- #define **DMA_SxCR_PFCTRL** ((uint32_t)0x00000020)
- #define **DMA_SxCR_TCIE** ((uint32_t)0x00000010)
- #define **DMA_SxCR_HTIE** ((uint32_t)0x00000008)
- #define **DMA_SxCR_TEIE** ((uint32_t)0x00000004)
- #define **DMA_SxCR_DMEIE** ((uint32_t)0x00000002)
- #define **DMA_SxCR_EN** ((uint32_t)0x00000001)
- #define **DMA_SxNDT** ((uint32_t)0x0000FFFF)
- #define **DMA_SxNDT_0** ((uint32_t)0x00000001)
- #define **DMA_SxNDT_1** ((uint32_t)0x00000002)
- #define **DMA_SxNDT_2** ((uint32_t)0x00000004)
- #define **DMA_SxNDT_3** ((uint32_t)0x00000008)
- #define **DMA_SxNDT_4** ((uint32_t)0x00000010)
- #define **DMA_SxNDT_5** ((uint32_t)0x00000020)
- #define **DMA_SxNDT_6** ((uint32_t)0x00000040)
- #define **DMA_SxNDT_7** ((uint32_t)0x00000080)
- #define **DMA_SxNDT_8** ((uint32_t)0x00000100)
- #define **DMA_SxNDT_9** ((uint32_t)0x00000200)
- #define **DMA_SxNDT_10** ((uint32_t)0x00000400)
- #define **DMA_SxNDT_11** ((uint32_t)0x00000800)
- #define **DMA_SxNDT_12** ((uint32_t)0x00001000)
- #define **DMA_SxNDT_13** ((uint32_t)0x00002000)
- #define **DMA_SxNDT_14** ((uint32_t)0x00004000)
- #define **DMA_SxNDT_15** ((uint32_t)0x00008000)
- #define **DMA_SxFCR_FEIE** ((uint32_t)0x00000080)

- #define **DMA_SxFCR_FS** ((uint32_t)0x00000038)
- #define **DMA_SxFCR_FS_0** ((uint32_t)0x00000008)
- #define **DMA_SxFCR_FS_1** ((uint32_t)0x00000010)
- #define **DMA_SxFCR_FS_2** ((uint32_t)0x00000020)
- #define **DMA_SxFCR_DMDIS** ((uint32_t)0x00000004)
- #define **DMA_SxFCR_FTH** ((uint32_t)0x00000003)
- #define **DMA_SxFCR_FTH_0** ((uint32_t)0x00000001)
- #define **DMA_SxFCR_FTH_1** ((uint32_t)0x00000002)
- #define **DMA_LISR_TCIF3** ((uint32_t)0x08000000)
- #define **DMA_LISR_HTIF3** ((uint32_t)0x04000000)
- #define **DMA_LISR_TEIF3** ((uint32_t)0x02000000)
- #define **DMA_LISR_DMEIF3** ((uint32_t)0x01000000)
- #define **DMA_LISR_FEIF3** ((uint32_t)0x00400000)
- #define **DMA_LISR_TCIF2** ((uint32_t)0x00200000)
- #define **DMA_LISR_HTIF2** ((uint32_t)0x00100000)
- #define **DMA_LISR_TEIF2** ((uint32_t)0x00080000)
- #define **DMA_LISR_DMEIF2** ((uint32_t)0x00040000)
- #define **DMA_LISR_FEIF2** ((uint32_t)0x00010000)
- #define **DMA_LISR_TCIF1** ((uint32_t)0x00000800)
- #define **DMA_LISR_HTIF1** ((uint32_t)0x00000400)
- #define **DMA_LISR_TEIF1** ((uint32_t)0x00000200)
- #define **DMA_LISR_DMEIF1** ((uint32_t)0x00000100)
- #define **DMA_LISR_FEIF1** ((uint32_t)0x00000040)
- #define **DMA_LISR_TCIF0** ((uint32_t)0x00000020)
- #define **DMA_LISR_Htif0** ((uint32_t)0x00000010)
- #define **DMA_LISR_TEIF0** ((uint32_t)0x00000008)
- #define **DMA_LISR_DMEIF0** ((uint32_t)0x00000004)
- #define **DMA_LISR_FEIF0** ((uint32_t)0x00000001)
- #define **DMA_HISR_TCIF7** ((uint32_t)0x08000000)
- #define **DMA_HISR_HTIF7** ((uint32_t)0x04000000)
- #define **DMA_HISR_TEIF7** ((uint32_t)0x02000000)
- #define **DMA_HISR_DMEIF7** ((uint32_t)0x01000000)
- #define **DMA_HISR_FEIF7** ((uint32_t)0x00400000)
- #define **DMA_HISR_TCIF6** ((uint32_t)0x00200000)
- #define **DMA_HISR_Htif6** ((uint32_t)0x00100000)
- #define **DMA_HISR_TEIF6** ((uint32_t)0x00080000)
- #define **DMA_HISR_DMEIF6** ((uint32_t)0x00040000)
- #define **DMA_HISR_FEIF6** ((uint32_t)0x00010000)
- #define **DMA_HISR_TCIF5** ((uint32_t)0x00000800)
- #define **DMA_HISR_Htif5** ((uint32_t)0x00000400)
- #define **DMA_HISR_TEIF5** ((uint32_t)0x00000200)
- #define **DMA_HISR_DMEIF5** ((uint32_t)0x00000100)
- #define **DMA_HISR_FEIF5** ((uint32_t)0x00000040)
- #define **DMA_HISR_TCIF4** ((uint32_t)0x00000020)
- #define **DMA_HISR_Htif4** ((uint32_t)0x00000010)
- #define **DMA_HISR_TEIF4** ((uint32_t)0x00000008)
- #define **DMA_HISR_DMEIF4** ((uint32_t)0x00000004)
- #define **DMA_HISR_FEIF4** ((uint32_t)0x00000001)
- #define **DMA_LIFCR_CTCIF3** ((uint32_t)0x08000000)
- #define **DMA_LIFCR_CHTIF3** ((uint32_t)0x04000000)
- #define **DMA_LIFCR_CTEIF3** ((uint32_t)0x02000000)
- #define **DMA_LIFCR_CDMEIF3** ((uint32_t)0x01000000)
- #define **DMA_LIFCR_CFEIF3** ((uint32_t)0x00400000)
- #define **DMA_LIFCR_CTCIF2** ((uint32_t)0x00200000)
- #define **DMA_LIFCR_CHTIF2** ((uint32_t)0x00100000)

```
• #define DMA_LIFCR_CTEIF2 ((uint32_t)0x00080000)
• #define DMA_LIFCR_CDMEIF2 ((uint32_t)0x00040000)
• #define DMA_LIFCR_CFEIF2 ((uint32_t)0x00010000)
• #define DMA_LIFCR_CTCIF1 ((uint32_t)0x00000800)
• #define DMA_LIFCR_CHTIF1 ((uint32_t)0x00000400)
• #define DMA_LIFCR_CTEIF1 ((uint32_t)0x00000200)
• #define DMA_LIFCR_CDMEIF1 ((uint32_t)0x000000100)
• #define DMA_LIFCR_CFEIF1 ((uint32_t)0x00000040)
• #define DMA_LIFCR_CTCIFO ((uint32_t)0x00000020)
• #define DMA_LIFCR_CHTIFO ((uint32_t)0x00000010)
• #define DMA_LIFCR_CTEIFO ((uint32_t)0x00000008)
• #define DMA_LIFCR_CDMEIFO ((uint32_t)0x00000004)
• #define DMA_LIFCR_CFEIFO ((uint32_t)0x00000001)
• #define DMA_HIFCR_CTCIF7 ((uint32_t)0x08000000)
• #define DMA_HIFCR_CHTIF7 ((uint32_t)0x04000000)
• #define DMA_HIFCR_CTEIF7 ((uint32_t)0x02000000)
• #define DMA_HIFCR_CDMEIF7 ((uint32_t)0x01000000)
• #define DMA_HIFCR_CFEIF7 ((uint32_t)0x00400000)
• #define DMA_HIFCR_CTCIF6 ((uint32_t)0x00200000)
• #define DMA_HIFCR_CHTIF6 ((uint32_t)0x00100000)
• #define DMA_HIFCR_CTEIF6 ((uint32_t)0x00080000)
• #define DMA_HIFCR_CDMEIF6 ((uint32_t)0x00040000)
• #define DMA_HIFCR_CFEIF6 ((uint32_t)0x00010000)
• #define DMA_HIFCR_CTCIF5 ((uint32_t)0x00000800)
• #define DMA_HIFCR_CHTIF5 ((uint32_t)0x00000400)
• #define DMA_HIFCR_CTEIF5 ((uint32_t)0x00000200)
• #define DMA_HIFCR_CDMEIF5 ((uint32_t)0x00000100)
• #define DMA_HIFCR_CFEIF5 ((uint32_t)0x00000040)
• #define DMA_HIFCR_CTCIF4 ((uint32_t)0x00000020)
• #define DMA_HIFCR_CHTIF4 ((uint32_t)0x00000010)
• #define DMA_HIFCR_CTEIF4 ((uint32_t)0x00000008)
• #define DMA_HIFCR_CDMEIF4 ((uint32_t)0x00000004)
• #define DMA_HIFCR_CFEIF4 ((uint32_t)0x00000001)
• #define DMA2D_CR_START ((uint32_t)0x00000001)
• #define DMA2D_CR_SUSP ((uint32_t)0x00000002)
• #define DMA2D_CR_ABORT ((uint32_t)0x00000004)
• #define DMA2D_CR_TEIE ((uint32_t)0x00000100)
• #define DMA2D_CR_TCIE ((uint32_t)0x00000200)
• #define DMA2D_CR_TWIE ((uint32_t)0x00000400)
• #define DMA2D_CR_CAEIE ((uint32_t)0x00000800)
• #define DMA2D_CR_CTCIE ((uint32_t)0x00001000)
• #define DMA2D_CR_CEIE ((uint32_t)0x00002000)
• #define DMA2D_CR_MODE ((uint32_t)0x00030000)
• #define DMA2D_ISR_TEIF ((uint32_t)0x00000001)
• #define DMA2D_ISR_TCIF ((uint32_t)0x00000002)
• #define DMA2D_ISR_TWIF ((uint32_t)0x00000004)
• #define DMA2D_ISR_CAEIF ((uint32_t)0x00000008)
• #define DMA2D_ISR_CTCIF ((uint32_t)0x00000010)
• #define DMA2D_ISR_CEIF ((uint32_t)0x00000020)
• #define DMA2D_IFCR_CTEIF ((uint32_t)0x00000001)
• #define DMA2D_IFCR_CTCIF ((uint32_t)0x00000002)
• #define DMA2D_IFCR_CTWIF ((uint32_t)0x00000004)
• #define DMA2D_IFCR_CAE CIF ((uint32_t)0x00000008)
• #define DMA2D_IFCR_CCTCIF ((uint32_t)0x00000010)
• #define DMA2D_IFCR_CCEIF ((uint32_t)0x00000020)
```

```
• #define DMA2D_IFSR_CTEIF DMA2D_IFCR_CTEIF
• #define DMA2D_IFSR_CTCIF DMA2D_IFCR_CTCIF
• #define DMA2D_IFSR_CTWIF DMA2D_IFCR_CTWIF
• #define DMA2D_IFSR_CCAEIF DMA2D_IFCR_CAECIF
• #define DMA2D_IFSR_CCTCIF DMA2D_IFCR_CCTCIF
• #define DMA2D_IFSR_CCEIF DMA2D_IFCR_CCEIF
• #define DMA2D_FGMAR_MA ((uint32_t)0xFFFFFFFF)
• #define DMA2D_FGOR_LO ((uint32_t)0x00003FFF)
• #define DMA2D_BGMAR_MA ((uint32_t)0xFFFFFFFF)
• #define DMA2D_BGOR_LO ((uint32_t)0x00003FFF)
• #define DMA2D_FGPFCCR_CM ((uint32_t)0x0000000F)
• #define DMA2D_FGPFCCR_CM_0 ((uint32_t)0x00000001)
• #define DMA2D_FGPFCCR_CM_1 ((uint32_t)0x00000002)
• #define DMA2D_FGPFCCR_CM_2 ((uint32_t)0x00000004)
• #define DMA2D_FGPFCCR_CM_3 ((uint32_t)0x00000008)
• #define DMA2D_FGPFCCR_CCM ((uint32_t)0x00000010)
• #define DMA2D_FGPFCCR_START ((uint32_t)0x00000020)
• #define DMA2D_FGPFCCR_CS ((uint32_t)0x0000FF00)
• #define DMA2D_FGPFCCR_AM ((uint32_t)0x00030000)
• #define DMA2D_FGPFCCR_AM_0 ((uint32_t)0x00010000)
• #define DMA2D_FGPFCCR_AM_1 ((uint32_t)0x00020000)
• #define DMA2D_FGPFCCR_ALPHA ((uint32_t)0xFF000000)
• #define DMA2D_FGCOLR_BLUE ((uint32_t)0x000000FF)
• #define DMA2D_FGCOLR_GREEN ((uint32_t)0x0000FF00)
• #define DMA2D_FGCOLR_RED ((uint32_t)0x00FF0000)
• #define DMA2D_BGPFCCR_CM ((uint32_t)0x0000000F)
• #define DMA2D_BGPFCCR_CM_0 ((uint32_t)0x00000001)
• #define DMA2D_BGPFCCR_CM_1 ((uint32_t)0x00000002)
• #define DMA2D_BGPFCCR_CM_2 ((uint32_t)0x00000004)
• #define DMA2D_BGPFCCR_CM_3 ((uint32_t)0x00000008)
• #define DMA2D_BGPFCCR_CCM ((uint32_t)0x00000010)
• #define DMA2D_BGPFCCR_START ((uint32_t)0x00000020)
• #define DMA2D_BGPFCCR_CS ((uint32_t)0x0000FF00)
• #define DMA2D_BGPFCCR_AM ((uint32_t)0x00030000)
• #define DMA2D_BGPFCCR_AM_0 ((uint32_t)0x00010000)
• #define DMA2D_BGPFCCR_AM_1 ((uint32_t)0x00020000)
• #define DMA2D_BGPFCCR_ALPHA ((uint32_t)0xFF000000)
• #define DMA2D_BGCOLR_BLUE ((uint32_t)0x000000FF)
• #define DMA2D_BGCOLR_GREEN ((uint32_t)0x0000FF00)
• #define DMA2D_BGCOLR_RED ((uint32_t)0x00FF0000)
• #define DMA2D_FGCMAR_MA ((uint32_t)0xFFFFFFFF)
• #define DMA2D_BGCMAR_MA ((uint32_t)0xFFFFFFFF)
• #define DMA2D_OPFCCR_CM ((uint32_t)0x00000007)
• #define DMA2D_OPFCCR_CM_0 ((uint32_t)0x00000001)
• #define DMA2D_OPFCCR_CM_1 ((uint32_t)0x00000002)
• #define DMA2D_OPFCCR_CM_2 ((uint32_t)0x00000004)
• #define DMA2D_OCOLR_BLUE_1 ((uint32_t)0x000000FF)
• #define DMA2D_OCOLR_GREEN_1 ((uint32_t)0x0000FF00)
• #define DMA2D_OCOLR_RED_1 ((uint32_t)0x00FF0000)
• #define DMA2D_OCOLR_ALPHA_1 ((uint32_t)0xFF000000)
• #define DMA2D_OCOLR_BLUE_2 ((uint32_t)0x0000001F)
• #define DMA2D_OCOLR_GREEN_2 ((uint32_t)0x000007E0)
• #define DMA2D_OCOLR_RED_2 ((uint32_t)0x0000F800)
• #define DMA2D_OCOLR_BLUE_3 ((uint32_t)0x0000001F)
• #define DMA2D_OCOLR_GREEN_3 ((uint32_t)0x000003E0)
```

- #define DMA2D_OCOLR_RED_3 ((uint32_t)0x000007C00)
- #define DMA2D_OCOLR_ALPHA_3 ((uint32_t)0x00008000)
- #define DMA2D_OCOLR_BLUE_4 ((uint32_t)0x0000000F)
- #define DMA2D_OCOLR_GREEN_4 ((uint32_t)0x000000F0)
- #define DMA2D_OCOLR_RED_4 ((uint32_t)0x00000F00)
- #define DMA2D_OCOLR_ALPHA_4 ((uint32_t)0x0000F000)
- #define DMA2D_OMAR_MA ((uint32_t)0xFFFFFFFF)
- #define DMA2D_OOR_LO ((uint32_t)0x00003FFF)
- #define DMA2D_NLR_NL ((uint32_t)0x0000FFFF)
- #define DMA2D_NLR_PL ((uint32_t)0x3FFF0000)
- #define DMA2D_LWR_LW ((uint32_t)0x0000FFFF)
- #define DMA2D_AMTCR_EN ((uint32_t)0x00000001)
- #define DMA2D_AMTCR_DT ((uint32_t)0x0000FF00)
- #define EXTI_IMR_MR0 ((uint32_t)0x00000001)
- #define EXTI_IMR_MR1 ((uint32_t)0x00000002)
- #define EXTI_IMR_MR2 ((uint32_t)0x00000004)
- #define EXTI_IMR_MR3 ((uint32_t)0x00000008)
- #define EXTI_IMR_MR4 ((uint32_t)0x00000010)
- #define EXTI_IMR_MR5 ((uint32_t)0x00000020)
- #define EXTI_IMR_MR6 ((uint32_t)0x00000040)
- #define EXTI_IMR_MR7 ((uint32_t)0x00000080)
- #define EXTI_IMR_MR8 ((uint32_t)0x00000100)
- #define EXTI_IMR_MR9 ((uint32_t)0x00000200)
- #define EXTI_IMR_MR10 ((uint32_t)0x00000400)
- #define EXTI_IMR_MR11 ((uint32_t)0x00000800)
- #define EXTI_IMR_MR12 ((uint32_t)0x00001000)
- #define EXTI_IMR_MR13 ((uint32_t)0x00002000)
- #define EXTI_IMR_MR14 ((uint32_t)0x00004000)
- #define EXTI_IMR_MR15 ((uint32_t)0x00008000)
- #define EXTI_IMR_MR16 ((uint32_t)0x00010000)
- #define EXTI_IMR_MR17 ((uint32_t)0x00020000)
- #define EXTI_IMR_MR18 ((uint32_t)0x00040000)
- #define EXTI_IMR_MR19 ((uint32_t)0x00080000)
- #define EXTI_IMR_MR23 ((uint32_t)0x00800000)
- #define EXTI_EMR_MR0 ((uint32_t)0x00000001)
- #define EXTI_EMR_MR1 ((uint32_t)0x00000002)
- #define EXTI_EMR_MR2 ((uint32_t)0x00000004)
- #define EXTI_EMR_MR3 ((uint32_t)0x00000008)
- #define EXTI_EMR_MR4 ((uint32_t)0x00000010)
- #define EXTI_EMR_MR5 ((uint32_t)0x00000020)
- #define EXTI_EMR_MR6 ((uint32_t)0x00000040)
- #define EXTI_EMR_MR7 ((uint32_t)0x00000080)
- #define EXTI_EMR_MR8 ((uint32_t)0x00000100)
- #define EXTI_EMR_MR9 ((uint32_t)0x00000200)
- #define EXTI_EMR_MR10 ((uint32_t)0x00000400)
- #define EXTI_EMR_MR11 ((uint32_t)0x00000800)
- #define EXTI_EMR_MR12 ((uint32_t)0x00001000)
- #define EXTI_EMR_MR13 ((uint32_t)0x00002000)
- #define EXTI_EMR_MR14 ((uint32_t)0x00004000)
- #define EXTI_EMR_MR15 ((uint32_t)0x00008000)
- #define EXTI_EMR_MR16 ((uint32_t)0x00010000)
- #define EXTI_EMR_MR17 ((uint32_t)0x00020000)
- #define EXTI_EMR_MR18 ((uint32_t)0x00040000)
- #define EXTI_EMR_MR19 ((uint32_t)0x00080000)
- #define EXTI_EMR_MR23 ((uint32_t)0x00800000)

```
• #define EXTI_RTSR_TR0 ((uint32_t)0x00000001)
• #define EXTI_RTSR_TR1 ((uint32_t)0x00000002)
• #define EXTI_RTSR_TR2 ((uint32_t)0x00000004)
• #define EXTI_RTSR_TR3 ((uint32_t)0x00000008)
• #define EXTI_RTSR_TR4 ((uint32_t)0x00000010)
• #define EXTI_RTSR_TR5 ((uint32_t)0x00000020)
• #define EXTI_RTSR_TR6 ((uint32_t)0x00000040)
• #define EXTI_RTSR_TR7 ((uint32_t)0x00000080)
• #define EXTI_RTSR_TR8 ((uint32_t)0x00000100)
• #define EXTI_RTSR_TR9 ((uint32_t)0x00000200)
• #define EXTI_RTSR_TR10 ((uint32_t)0x00000400)
• #define EXTI_RTSR_TR11 ((uint32_t)0x00000800)
• #define EXTI_RTSR_TR12 ((uint32_t)0x00001000)
• #define EXTI_RTSR_TR13 ((uint32_t)0x00002000)
• #define EXTI_RTSR_TR14 ((uint32_t)0x00004000)
• #define EXTI_RTSR_TR15 ((uint32_t)0x00008000)
• #define EXTI_RTSR_TR16 ((uint32_t)0x00010000)
• #define EXTI_RTSR_TR17 ((uint32_t)0x00020000)
• #define EXTI_RTSR_TR18 ((uint32_t)0x00040000)
• #define EXTI_RTSR_TR19 ((uint32_t)0x00080000)
• #define EXTI_RTSR_TR23 ((uint32_t)0x00800000)
• #define EXTI_FTSR_TR0 ((uint32_t)0x00000001)
• #define EXTI_FTSR_TR1 ((uint32_t)0x00000002)
• #define EXTI_FTSR_TR2 ((uint32_t)0x00000004)
• #define EXTI_FTSR_TR3 ((uint32_t)0x00000008)
• #define EXTI_FTSR_TR4 ((uint32_t)0x00000010)
• #define EXTI_FTSR_TR5 ((uint32_t)0x00000020)
• #define EXTI_FTSR_TR6 ((uint32_t)0x00000040)
• #define EXTI_FTSR_TR7 ((uint32_t)0x00000080)
• #define EXTI_FTSR_TR8 ((uint32_t)0x00000100)
• #define EXTI_FTSR_TR9 ((uint32_t)0x00000200)
• #define EXTI_FTSR_TR10 ((uint32_t)0x00000400)
• #define EXTI_FTSR_TR11 ((uint32_t)0x00000800)
• #define EXTI_FTSR_TR12 ((uint32_t)0x00001000)
• #define EXTI_FTSR_TR13 ((uint32_t)0x00002000)
• #define EXTI_FTSR_TR14 ((uint32_t)0x00004000)
• #define EXTI_FTSR_TR15 ((uint32_t)0x00008000)
• #define EXTI_FTSR_TR16 ((uint32_t)0x00010000)
• #define EXTI_FTSR_TR17 ((uint32_t)0x00020000)
• #define EXTI_FTSR_TR18 ((uint32_t)0x00040000)
• #define EXTI_FTSR_TR19 ((uint32_t)0x00080000)
• #define EXTI_FTSR_TR23 ((uint32_t)0x00800000)
• #define EXTI_SWIER_SWIER0 ((uint32_t)0x00000001)
• #define EXTI_SWIER_SWIER1 ((uint32_t)0x00000002)
• #define EXTI_SWIER_SWIER2 ((uint32_t)0x00000004)
• #define EXTI_SWIER_SWIER3 ((uint32_t)0x00000008)
• #define EXTI_SWIER_SWIER4 ((uint32_t)0x00000010)
• #define EXTI_SWIER_SWIER5 ((uint32_t)0x00000020)
• #define EXTI_SWIER_SWIER6 ((uint32_t)0x00000040)
• #define EXTI_SWIER_SWIER7 ((uint32_t)0x00000080)
• #define EXTI_SWIER_SWIER8 ((uint32_t)0x00000100)
• #define EXTI_SWIER_SWIER9 ((uint32_t)0x00000200)
• #define EXTI_SWIER_SWIER10 ((uint32_t)0x00000400)
• #define EXTI_SWIER_SWIER11 ((uint32_t)0x00000800)
• #define EXTI_SWIER_SWIER12 ((uint32_t)0x00001000)
```

- #define **EXTI_SWIER_SWIER13** ((uint32_t)0x00002000)
- #define **EXTI_SWIER_SWIER14** ((uint32_t)0x00004000)
- #define **EXTI_SWIER_SWIER15** ((uint32_t)0x00008000)
- #define **EXTI_SWIER_SWIER16** ((uint32_t)0x00010000)
- #define **EXTI_SWIER_SWIER17** ((uint32_t)0x00020000)
- #define **EXTI_SWIER_SWIER18** ((uint32_t)0x00040000)
- #define **EXTI_SWIER_SWIER19** ((uint32_t)0x00080000)
- #define **EXTI_SWIER_SWIER23** ((uint32_t)0x00800000)
- #define **EXTI_PR_PR0** ((uint32_t)0x00000001)
- #define **EXTI_PR_PR1** ((uint32_t)0x00000002)
- #define **EXTI_PR_PR2** ((uint32_t)0x00000004)
- #define **EXTI_PR_PR3** ((uint32_t)0x00000008)
- #define **EXTI_PR_PR4** ((uint32_t)0x00000010)
- #define **EXTI_PR_PR5** ((uint32_t)0x00000020)
- #define **EXTI_PR_PR6** ((uint32_t)0x00000040)
- #define **EXTI_PR_PR7** ((uint32_t)0x00000080)
- #define **EXTI_PR_PR8** ((uint32_t)0x00000100)
- #define **EXTI_PR_PR9** ((uint32_t)0x00000200)
- #define **EXTI_PR_PR10** ((uint32_t)0x00000400)
- #define **EXTI_PR_PR11** ((uint32_t)0x00000800)
- #define **EXTI_PR_PR12** ((uint32_t)0x00001000)
- #define **EXTI_PR_PR13** ((uint32_t)0x00002000)
- #define **EXTI_PR_PR14** ((uint32_t)0x00004000)
- #define **EXTI_PR_PR15** ((uint32_t)0x00008000)
- #define **EXTI_PR_PR16** ((uint32_t)0x00010000)
- #define **EXTI_PR_PR17** ((uint32_t)0x00020000)
- #define **EXTI_PR_PR18** ((uint32_t)0x00040000)
- #define **EXTI_PR_PR19** ((uint32_t)0x00080000)
- #define **EXTI_PR_PR23** ((uint32_t)0x00800000)
- #define **FLASH_ACR_LATENCY** ((uint32_t)0x0000000F)
- #define **FLASH_ACR_LATENCY_0WS** ((uint32_t)0x00000000)
- #define **FLASH_ACR_LATENCY_1WS** ((uint32_t)0x00000001)
- #define **FLASH_ACR_LATENCY_2WS** ((uint32_t)0x00000002)
- #define **FLASH_ACR_LATENCY_3WS** ((uint32_t)0x00000003)
- #define **FLASH_ACR_LATENCY_4WS** ((uint32_t)0x00000004)
- #define **FLASH_ACR_LATENCY_5WS** ((uint32_t)0x00000005)
- #define **FLASH_ACR_LATENCY_6WS** ((uint32_t)0x00000006)
- #define **FLASH_ACR_LATENCY_7WS** ((uint32_t)0x00000007)
- #define **FLASH_ACR_LATENCY_8WS** ((uint32_t)0x00000008)
- #define **FLASH_ACR_LATENCY_9WS** ((uint32_t)0x00000009)
- #define **FLASH_ACR_LATENCY_10WS** ((uint32_t)0x0000000A)
- #define **FLASH_ACR_LATENCY_11WS** ((uint32_t)0x0000000B)
- #define **FLASH_ACR_LATENCY_12WS** ((uint32_t)0x0000000C)
- #define **FLASH_ACR_LATENCY_13WS** ((uint32_t)0x0000000D)
- #define **FLASH_ACR_LATENCY_14WS** ((uint32_t)0x0000000E)
- #define **FLASH_ACR_LATENCY_15WS** ((uint32_t)0x0000000F)
- #define **FLASH_ACR_PRFTEN** ((uint32_t)0x00000100)
- #define **FLASH_ACR_ICEN** ((uint32_t)0x00000200)
- #define **FLASH_ACR_DCEN** ((uint32_t)0x00000400)
- #define **FLASH_ACR_ICRST** ((uint32_t)0x00000800)
- #define **FLASH_ACR_DCRST** ((uint32_t)0x00001000)
- #define **FLASH_ACR_BYTE0_ADDRESS** ((uint32_t)0x40023C00)
- #define **FLASH_ACR_BYTE2_ADDRESS** ((uint32_t)0x40023C03)
- #define **FLASH_SR_EOP** ((uint32_t)0x00000001)
- #define **FLASH_SR_SOP** ((uint32_t)0x00000002)

- #define **FLASH_SR_WRPERR** ((uint32_t)0x00000010)
- #define **FLASH_SR_PGAERR** ((uint32_t)0x00000020)
- #define **FLASH_SR_PGPERR** ((uint32_t)0x00000040)
- #define **FLASH_SR_PGSERR** ((uint32_t)0x00000080)
- #define **FLASH_SR_BSY** ((uint32_t)0x00010000)
- #define **FLASH_CR_PG** ((uint32_t)0x00000001)
- #define **FLASH_CR_SER** ((uint32_t)0x00000002)
- #define **FLASH_CR_MER** ((uint32_t)0x00000004)
- #define **FLASH_CR_MER1** FLASH_CR_MER
- #define **FLASH_CR_SNBB** ((uint32_t)0x000000F8)
- #define **FLASH_CR_SNBB_0** ((uint32_t)0x00000008)
- #define **FLASH_CR_SNBB_1** ((uint32_t)0x00000010)
- #define **FLASH_CR_SNBB_2** ((uint32_t)0x00000020)
- #define **FLASH_CR_SNBB_3** ((uint32_t)0x00000040)
- #define **FLASH_CR_SNBB_4** ((uint32_t)0x00000040)
- #define **FLASH_CR_PSIZE** ((uint32_t)0x00000300)
- #define **FLASH_CR_PSIZE_0** ((uint32_t)0x00000100)
- #define **FLASH_CR_PSIZE_1** ((uint32_t)0x00000200)
- #define **FLASH_CR_MER2** ((uint32_t)0x00008000)
- #define **FLASH_CR_STRT** ((uint32_t)0x00010000)
- #define **FLASH_CR_EOPIE** ((uint32_t)0x01000000)
- #define **FLASH_CR_LOCK** ((uint32_t)0x80000000)
- #define **FLASH_OPTCR_OPTLOCK** ((uint32_t)0x00000001)
- #define **FLASH_OPTCR_OPTSTRT** ((uint32_t)0x00000002)
- #define **FLASH_OPTCR_BORLEV_0** ((uint32_t)0x00000004)
- #define **FLASH_OPTCR_BORLEV_1** ((uint32_t)0x00000008)
- #define **FLASH_OPTCR_BORLEV** ((uint32_t)0x0000000C)
- #define **FLASH_OPTCR_BFB2** ((uint32_t)0x00000010)
- #define **FLASH_OPTCR_WDG_SW** ((uint32_t)0x00000020)
- #define **FLASH_OPTCR_nRST_STOP** ((uint32_t)0x00000040)
- #define **FLASH_OPTCR_nRST_STDBY** ((uint32_t)0x00000080)
- #define **FLASH_OPTCR_RDP** ((uint32_t)0x0000FF00)
- #define **FLASH_OPTCR_RDP_0** ((uint32_t)0x00000100)
- #define **FLASH_OPTCR_RDP_1** ((uint32_t)0x00000200)
- #define **FLASH_OPTCR_RDP_2** ((uint32_t)0x00000400)
- #define **FLASH_OPTCR_RDP_3** ((uint32_t)0x00000800)
- #define **FLASH_OPTCR_RDP_4** ((uint32_t)0x00001000)
- #define **FLASH_OPTCR_RDP_5** ((uint32_t)0x00002000)
- #define **FLASH_OPTCR_RDP_6** ((uint32_t)0x00004000)
- #define **FLASH_OPTCR_RDP_7** ((uint32_t)0x00008000)
- #define **FLASH_OPTCR_nWRP** ((uint32_t)0xFFFF0000)
- #define **FLASH_OPTCR_nWRP_0** ((uint32_t)0x00010000)
- #define **FLASH_OPTCR_nWRP_1** ((uint32_t)0x00020000)
- #define **FLASH_OPTCR_nWRP_2** ((uint32_t)0x00040000)
- #define **FLASH_OPTCR_nWRP_3** ((uint32_t)0x00080000)
- #define **FLASH_OPTCR_nWRP_4** ((uint32_t)0x00100000)
- #define **FLASH_OPTCR_nWRP_5** ((uint32_t)0x00200000)
- #define **FLASH_OPTCR_nWRP_6** ((uint32_t)0x00400000)
- #define **FLASH_OPTCR_nWRP_7** ((uint32_t)0x00800000)
- #define **FLASH_OPTCR_nWRP_8** ((uint32_t)0x01000000)
- #define **FLASH_OPTCR_nWRP_9** ((uint32_t)0x02000000)
- #define **FLASH_OPTCR_nWRP_10** ((uint32_t)0x04000000)
- #define **FLASH_OPTCR_nWRP_11** ((uint32_t)0x08000000)
- #define **FLASH_OPTCR_DB1M** ((uint32_t)0x40000000)
- #define **FLASH_OPTCR_SPRMOD** ((uint32_t)0x80000000)

```
• #define FLASH_OPTCR1_nWRP ((uint32_t)0xFFFF0000)
• #define FLASH_OPTCR1_nWRP_0 ((uint32_t)0x00010000)
• #define FLASH_OPTCR1_nWRP_1 ((uint32_t)0x00020000)
• #define FLASH_OPTCR1_nWRP_2 ((uint32_t)0x00040000)
• #define FLASH_OPTCR1_nWRP_3 ((uint32_t)0x00080000)
• #define FLASH_OPTCR1_nWRP_4 ((uint32_t)0x00100000)
• #define FLASH_OPTCR1_nWRP_5 ((uint32_t)0x00200000)
• #define FLASH_OPTCR1_nWRP_6 ((uint32_t)0x00400000)
• #define FLASH_OPTCR1_nWRP_7 ((uint32_t)0x00800000)
• #define FLASH_OPTCR1_nWRP_8 ((uint32_t)0x01000000)
• #define FLASH_OPTCR1_nWRP_9 ((uint32_t)0x02000000)
• #define FLASH_OPTCR1_nWRP_10 ((uint32_t)0x04000000)
• #define FLASH_OPTCR1_nWRP_11 ((uint32_t)0x08000000)
• #define GPIO_MODER_MODER0 ((uint32_t)0x00000003)
• #define GPIO_MODER_MODER0_0 ((uint32_t)0x00000001)
• #define GPIO_MODER_MODER0_1 ((uint32_t)0x00000002)
• #define GPIO_MODER_MODER1 ((uint32_t)0x0000000C)
• #define GPIO_MODER_MODER1_0 ((uint32_t)0x00000004)
• #define GPIO_MODER_MODER1_1 ((uint32_t)0x00000008)
• #define GPIO_MODER_MODER2 ((uint32_t)0x00000030)
• #define GPIO_MODER_MODER2_0 ((uint32_t)0x00000010)
• #define GPIO_MODER_MODER2_1 ((uint32_t)0x00000020)
• #define GPIO_MODER_MODER3 ((uint32_t)0x000000C0)
• #define GPIO_MODER_MODER3_0 ((uint32_t)0x00000040)
• #define GPIO_MODER_MODER3_1 ((uint32_t)0x00000080)
• #define GPIO_MODER_MODER4 ((uint32_t)0x00000300)
• #define GPIO_MODER_MODER4_0 ((uint32_t)0x00000100)
• #define GPIO_MODER_MODER4_1 ((uint32_t)0x00000200)
• #define GPIO_MODER_MODER5 ((uint32_t)0x00000C00)
• #define GPIO_MODER_MODER5_0 ((uint32_t)0x00000400)
• #define GPIO_MODER_MODER5_1 ((uint32_t)0x00000800)
• #define GPIO_MODER_MODER6 ((uint32_t)0x00003000)
• #define GPIO_MODER_MODER6_0 ((uint32_t)0x00001000)
• #define GPIO_MODER_MODER6_1 ((uint32_t)0x00002000)
• #define GPIO_MODER_MODER7 ((uint32_t)0x0000C000)
• #define GPIO_MODER_MODER7_0 ((uint32_t)0x00004000)
• #define GPIO_MODER_MODER7_1 ((uint32_t)0x00008000)
• #define GPIO_MODER_MODER8 ((uint32_t)0x00030000)
• #define GPIO_MODER_MODER8_0 ((uint32_t)0x00010000)
• #define GPIO_MODER_MODER8_1 ((uint32_t)0x00020000)
• #define GPIO_MODER_MODER9 ((uint32_t)0x000C0000)
• #define GPIO_MODER_MODER9_0 ((uint32_t)0x00040000)
• #define GPIO_MODER_MODER9_1 ((uint32_t)0x00080000)
• #define GPIO_MODER_MODER10 ((uint32_t)0x00300000)
• #define GPIO_MODER_MODER10_0 ((uint32_t)0x00100000)
• #define GPIO_MODER_MODER10_1 ((uint32_t)0x00200000)
• #define GPIO_MODER_MODER11 ((uint32_t)0x00C00000)
• #define GPIO_MODER_MODER11_0 ((uint32_t)0x00400000)
• #define GPIO_MODER_MODER11_1 ((uint32_t)0x00800000)
• #define GPIO_MODER_MODER12 ((uint32_t)0x03000000)
• #define GPIO_MODER_MODER12_0 ((uint32_t)0x01000000)
• #define GPIO_MODER_MODER12_1 ((uint32_t)0x02000000)
• #define GPIO_MODER_MODER13 ((uint32_t)0x0C000000)
• #define GPIO_MODER_MODER13_0 ((uint32_t)0x04000000)
• #define GPIO_MODER_MODER13_1 ((uint32_t)0x08000000)
```

- #define **GPIO_MODER_MODER14** ((uint32_t)0x30000000)
- #define **GPIO_MODER_MODER14_0** ((uint32_t)0x10000000)
- #define **GPIO_MODER_MODER14_1** ((uint32_t)0x20000000)
- #define **GPIO_MODER_MODER15** ((uint32_t)0xC0000000)
- #define **GPIO_MODER_MODER15_0** ((uint32_t)0x40000000)
- #define **GPIO_MODER_MODER15_1** ((uint32_t)0x80000000)
- #define **GPIO_OTYPER_OT_0** ((uint32_t)0x00000001)
- #define **GPIO_OTYPER_OT_1** ((uint32_t)0x00000002)
- #define **GPIO_OTYPER_OT_2** ((uint32_t)0x00000004)
- #define **GPIO_OTYPER_OT_3** ((uint32_t)0x00000008)
- #define **GPIO_OTYPER_OT_4** ((uint32_t)0x00000010)
- #define **GPIO_OTYPER_OT_5** ((uint32_t)0x00000020)
- #define **GPIO_OTYPER_OT_6** ((uint32_t)0x00000040)
- #define **GPIO_OTYPER_OT_7** ((uint32_t)0x00000080)
- #define **GPIO_OTYPER_OT_8** ((uint32_t)0x00000100)
- #define **GPIO_OTYPER_OT_9** ((uint32_t)0x00000200)
- #define **GPIO_OTYPER_OT_10** ((uint32_t)0x00000400)
- #define **GPIO_OTYPER_OT_11** ((uint32_t)0x00000800)
- #define **GPIO_OTYPER_OT_12** ((uint32_t)0x00001000)
- #define **GPIO_OTYPER_OT_13** ((uint32_t)0x00002000)
- #define **GPIO_OTYPER_OT_14** ((uint32_t)0x00004000)
- #define **GPIO_OTYPER_OT_15** ((uint32_t)0x00008000)
- #define **GPIO_OSPEEDER_OSPEEDR0** ((uint32_t)0x00000003)
- #define **GPIO_OSPEEDER_OSPEEDR0_0** ((uint32_t)0x00000001)
- #define **GPIO_OSPEEDER_OSPEEDR0_1** ((uint32_t)0x00000002)
- #define **GPIO_OSPEEDER_OSPEEDR1** ((uint32_t)0x0000000C)
- #define **GPIO_OSPEEDER_OSPEEDR1_0** ((uint32_t)0x00000004)
- #define **GPIO_OSPEEDER_OSPEEDR1_1** ((uint32_t)0x00000008)
- #define **GPIO_OSPEEDER_OSPEEDR2** ((uint32_t)0x00000030)
- #define **GPIO_OSPEEDER_OSPEEDR2_0** ((uint32_t)0x00000010)
- #define **GPIO_OSPEEDER_OSPEEDR2_1** ((uint32_t)0x00000020)
- #define **GPIO_OSPEEDER_OSPEEDR3** ((uint32_t)0x000000C0)
- #define **GPIO_OSPEEDER_OSPEEDR3_0** ((uint32_t)0x00000040)
- #define **GPIO_OSPEEDER_OSPEEDR3_1** ((uint32_t)0x00000080)
- #define **GPIO_OSPEEDER_OSPEEDR4** ((uint32_t)0x00000300)
- #define **GPIO_OSPEEDER_OSPEEDR4_0** ((uint32_t)0x00000100)
- #define **GPIO_OSPEEDER_OSPEEDR4_1** ((uint32_t)0x00000200)
- #define **GPIO_OSPEEDER_OSPEEDR5** ((uint32_t)0x00000C00)
- #define **GPIO_OSPEEDER_OSPEEDR5_0** ((uint32_t)0x00000400)
- #define **GPIO_OSPEEDER_OSPEEDR5_1** ((uint32_t)0x00000800)
- #define **GPIO_OSPEEDER_OSPEEDR6** ((uint32_t)0x00003000)
- #define **GPIO_OSPEEDER_OSPEEDR6_0** ((uint32_t)0x00001000)
- #define **GPIO_OSPEEDER_OSPEEDR6_1** ((uint32_t)0x00002000)
- #define **GPIO_OSPEEDER_OSPEEDR7** ((uint32_t)0x0000C000)
- #define **GPIO_OSPEEDER_OSPEEDR7_0** ((uint32_t)0x00004000)
- #define **GPIO_OSPEEDER_OSPEEDR7_1** ((uint32_t)0x00008000)
- #define **GPIO_OSPEEDER_OSPEEDR8** ((uint32_t)0x00030000)
- #define **GPIO_OSPEEDER_OSPEEDR8_0** ((uint32_t)0x00010000)
- #define **GPIO_OSPEEDER_OSPEEDR8_1** ((uint32_t)0x00020000)
- #define **GPIO_OSPEEDER_OSPEEDR9** ((uint32_t)0x000C0000)
- #define **GPIO_OSPEEDER_OSPEEDR9_0** ((uint32_t)0x00040000)
- #define **GPIO_OSPEEDER_OSPEEDR9_1** ((uint32_t)0x00080000)
- #define **GPIO_OSPEEDER_OSPEEDR10** ((uint32_t)0x00300000)
- #define **GPIO_OSPEEDER_OSPEEDR10_0** ((uint32_t)0x00100000)
- #define **GPIO_OSPEEDER_OSPEEDR10_1** ((uint32_t)0x00200000)

- #define **GPIO_OSPEEDER_OSPEEDR11** ((uint32_t)0x00C00000)
- #define **GPIO_OSPEEDER_OSPEEDR11_0** ((uint32_t)0x00400000)
- #define **GPIO_OSPEEDER_OSPEEDR11_1** ((uint32_t)0x00800000)
- #define **GPIO_OSPEEDER_OSPEEDR12** ((uint32_t)0x03000000)
- #define **GPIO_OSPEEDER_OSPEEDR12_0** ((uint32_t)0x01000000)
- #define **GPIO_OSPEEDER_OSPEEDR12_1** ((uint32_t)0x02000000)
- #define **GPIO_OSPEEDER_OSPEEDR13** ((uint32_t)0x0C000000)
- #define **GPIO_OSPEEDER_OSPEEDR13_0** ((uint32_t)0x04000000)
- #define **GPIO_OSPEEDER_OSPEEDR13_1** ((uint32_t)0x08000000)
- #define **GPIO_OSPEEDER_OSPEEDR14** ((uint32_t)0x30000000)
- #define **GPIO_OSPEEDER_OSPEEDR14_0** ((uint32_t)0x10000000)
- #define **GPIO_OSPEEDER_OSPEEDR14_1** ((uint32_t)0x20000000)
- #define **GPIO_OSPEEDER_OSPEEDR15** ((uint32_t)0xC0000000)
- #define **GPIO_OSPEEDER_OSPEEDR15_0** ((uint32_t)0x40000000)
- #define **GPIO_OSPEEDER_OSPEEDR15_1** ((uint32_t)0x80000000)
- #define **GPIO_PUPDR_PUPDR0** ((uint32_t)0x00000003)
- #define **GPIO_PUPDR_PUPDR0_0** ((uint32_t)0x00000001)
- #define **GPIO_PUPDR_PUPDR0_1** ((uint32_t)0x00000002)
- #define **GPIO_PUPDR_PUPDR1** ((uint32_t)0x0000000C)
- #define **GPIO_PUPDR_PUPDR1_0** ((uint32_t)0x00000004)
- #define **GPIO_PUPDR_PUPDR1_1** ((uint32_t)0x00000008)
- #define **GPIO_PUPDR_PUPDR2** ((uint32_t)0x00000030)
- #define **GPIO_PUPDR_PUPDR2_0** ((uint32_t)0x00000010)
- #define **GPIO_PUPDR_PUPDR2_1** ((uint32_t)0x00000020)
- #define **GPIO_PUPDR_PUPDR3** ((uint32_t)0x000000C0)
- #define **GPIO_PUPDR_PUPDR3_0** ((uint32_t)0x00000040)
- #define **GPIO_PUPDR_PUPDR3_1** ((uint32_t)0x00000080)
- #define **GPIO_PUPDR_PUPDR4** ((uint32_t)0x00000300)
- #define **GPIO_PUPDR_PUPDR4_0** ((uint32_t)0x00000100)
- #define **GPIO_PUPDR_PUPDR4_1** ((uint32_t)0x00000200)
- #define **GPIO_PUPDR_PUPDR5** ((uint32_t)0x00000C00)
- #define **GPIO_PUPDR_PUPDR5_0** ((uint32_t)0x00000400)
- #define **GPIO_PUPDR_PUPDR5_1** ((uint32_t)0x00000800)
- #define **GPIO_PUPDR_PUPDR6** ((uint32_t)0x00003000)
- #define **GPIO_PUPDR_PUPDR6_0** ((uint32_t)0x00001000)
- #define **GPIO_PUPDR_PUPDR6_1** ((uint32_t)0x00002000)
- #define **GPIO_PUPDR_PUPDR7** ((uint32_t)0x0000C000)
- #define **GPIO_PUPDR_PUPDR7_0** ((uint32_t)0x00004000)
- #define **GPIO_PUPDR_PUPDR7_1** ((uint32_t)0x00008000)
- #define **GPIO_PUPDR_PUPDR8** ((uint32_t)0x00030000)
- #define **GPIO_PUPDR_PUPDR8_0** ((uint32_t)0x00010000)
- #define **GPIO_PUPDR_PUPDR8_1** ((uint32_t)0x00020000)
- #define **GPIO_PUPDR_PUPDR9** ((uint32_t)0x000C0000)
- #define **GPIO_PUPDR_PUPDR9_0** ((uint32_t)0x00040000)
- #define **GPIO_PUPDR_PUPDR9_1** ((uint32_t)0x00080000)
- #define **GPIO_PUPDR_PUPDR10** ((uint32_t)0x00300000)
- #define **GPIO_PUPDR_PUPDR10_0** ((uint32_t)0x00100000)
- #define **GPIO_PUPDR_PUPDR10_1** ((uint32_t)0x00200000)
- #define **GPIO_PUPDR_PUPDR11** ((uint32_t)0x00C00000)
- #define **GPIO_PUPDR_PUPDR11_0** ((uint32_t)0x00400000)
- #define **GPIO_PUPDR_PUPDR11_1** ((uint32_t)0x00800000)
- #define **GPIO_PUPDR_PUPDR12** ((uint32_t)0x03000000)
- #define **GPIO_PUPDR_PUPDR12_0** ((uint32_t)0x01000000)
- #define **GPIO_PUPDR_PUPDR12_1** ((uint32_t)0x02000000)
- #define **GPIO_PUPDR_PUPDR13** ((uint32_t)0x0C000000)

```
• #define GPIO_PUPDR_PUPDR13_0 ((uint32_t)0x04000000)
• #define GPIO_PUPDR_PUPDR13_1 ((uint32_t)0x08000000)
• #define GPIO_PUPDR_PUPDR14 ((uint32_t)0x30000000)
• #define GPIO_PUPDR_PUPDR14_0 ((uint32_t)0x10000000)
• #define GPIO_PUPDR_PUPDR14_1 ((uint32_t)0x20000000)
• #define GPIO_PUPDR_PUPDR15 ((uint32_t)0xC0000000)
• #define GPIO_PUPDR_PUPDR15_0 ((uint32_t)0x40000000)
• #define GPIO_PUPDR_PUPDR15_1 ((uint32_t)0x80000000)
• #define GPIO_IDR_IDR_0 ((uint32_t)0x00000001)
• #define GPIO_IDR_IDR_1 ((uint32_t)0x00000002)
• #define GPIO_IDR_IDR_2 ((uint32_t)0x00000004)
• #define GPIO_IDR_IDR_3 ((uint32_t)0x00000008)
• #define GPIO_IDR_IDR_4 ((uint32_t)0x00000010)
• #define GPIO_IDR_IDR_5 ((uint32_t)0x00000020)
• #define GPIO_IDR_IDR_6 ((uint32_t)0x00000040)
• #define GPIO_IDR_IDR_7 ((uint32_t)0x00000080)
• #define GPIO_IDR_IDR_8 ((uint32_t)0x00000100)
• #define GPIO_IDR_IDR_9 ((uint32_t)0x00000200)
• #define GPIO_IDR_IDR_10 ((uint32_t)0x00000400)
• #define GPIO_IDR_IDR_11 ((uint32_t)0x00000800)
• #define GPIO_IDR_IDR_12 ((uint32_t)0x00001000)
• #define GPIO_IDR_IDR_13 ((uint32_t)0x00002000)
• #define GPIO_IDR_IDR_14 ((uint32_t)0x00004000)
• #define GPIO_IDR_IDR_15 ((uint32_t)0x00008000)
• #define GPIO_OTYPER_IDR_0 GPIO_IDR_IDR_0
• #define GPIO_OTYPER_IDR_1 GPIO_IDR_IDR_1
• #define GPIO_OTYPER_IDR_2 GPIO_IDR_IDR_2
• #define GPIO_OTYPER_IDR_3 GPIO_IDR_IDR_3
• #define GPIO_OTYPER_IDR_4 GPIO_IDR_IDR_4
• #define GPIO_OTYPER_IDR_5 GPIO_IDR_IDR_5
• #define GPIO_OTYPER_IDR_6 GPIO_IDR_IDR_6
• #define GPIO_OTYPER_IDR_7 GPIO_IDR_IDR_7
• #define GPIO_OTYPER_IDR_8 GPIO_IDR_IDR_8
• #define GPIO_OTYPER_IDR_9 GPIO_IDR_IDR_9
• #define GPIO_OTYPER_IDR_10 GPIO_IDR_IDR_10
• #define GPIO_OTYPER_IDR_11 GPIO_IDR_IDR_11
• #define GPIO_OTYPER_IDR_12 GPIO_IDR_IDR_12
• #define GPIO_OTYPER_IDR_13 GPIO_IDR_IDR_13
• #define GPIO_OTYPER_IDR_14 GPIO_IDR_IDR_14
• #define GPIO_OTYPER_IDR_15 GPIO_IDR_IDR_15
• #define GPIO_ODR_ODR_0 ((uint32_t)0x00000001)
• #define GPIO_ODR_ODR_1 ((uint32_t)0x00000002)
• #define GPIO_ODR_ODR_2 ((uint32_t)0x00000004)
• #define GPIO_ODR_ODR_3 ((uint32_t)0x00000008)
• #define GPIO_ODR_ODR_4 ((uint32_t)0x00000010)
• #define GPIO_ODR_ODR_5 ((uint32_t)0x00000020)
• #define GPIO_ODR_ODR_6 ((uint32_t)0x00000040)
• #define GPIO_ODR_ODR_7 ((uint32_t)0x00000080)
• #define GPIO_ODR_ODR_8 ((uint32_t)0x00000100)
• #define GPIO_ODR_ODR_9 ((uint32_t)0x00000200)
• #define GPIO_ODR_ODR_10 ((uint32_t)0x00000400)
• #define GPIO_ODR_ODR_11 ((uint32_t)0x00000800)
• #define GPIO_ODR_ODR_12 ((uint32_t)0x00001000)
• #define GPIO_ODR_ODR_13 ((uint32_t)0x00002000)
• #define GPIO_ODR_ODR_14 ((uint32_t)0x00004000)
```

```
• #define GPIO_ODR_ODR_15 ((uint32_t)0x00008000)
• #define GPIO_OTYPER_ODR_0 GPIO_ODR_ODR_0
• #define GPIO_OTYPER_ODR_1 GPIO_ODR_ODR_1
• #define GPIO_OTYPER_ODR_2 GPIO_ODR_ODR_2
• #define GPIO_OTYPER_ODR_3 GPIO_ODR_ODR_3
• #define GPIO_OTYPER_ODR_4 GPIO_ODR_ODR_4
• #define GPIO_OTYPER_ODR_5 GPIO_ODR_ODR_5
• #define GPIO_OTYPER_ODR_6 GPIO_ODR_ODR_6
• #define GPIO_OTYPER_ODR_7 GPIO_ODR_ODR_7
• #define GPIO_OTYPER_ODR_8 GPIO_ODR_ODR_8
• #define GPIO_OTYPER_ODR_9 GPIO_ODR_ODR_9
• #define GPIO_OTYPER_ODR_10 GPIO_ODR_ODR_10
• #define GPIO_OTYPER_ODR_11 GPIO_ODR_ODR_11
• #define GPIO_OTYPER_ODR_12 GPIO_ODR_ODR_12
• #define GPIO_OTYPER_ODR_13 GPIO_ODR_ODR_13
• #define GPIO_OTYPER_ODR_14 GPIO_ODR_ODR_14
• #define GPIO_OTYPER_ODR_15 GPIO_ODR_ODR_15
• #define GPIO_BSRR_BS_0 ((uint32_t)0x00000001)
• #define GPIO_BSRR_BS_1 ((uint32_t)0x00000002)
• #define GPIO_BSRR_BS_2 ((uint32_t)0x00000004)
• #define GPIO_BSRR_BS_3 ((uint32_t)0x00000008)
• #define GPIO_BSRR_BS_4 ((uint32_t)0x00000010)
• #define GPIO_BSRR_BS_5 ((uint32_t)0x00000020)
• #define GPIO_BSRR_BS_6 ((uint32_t)0x00000040)
• #define GPIO_BSRR_BS_7 ((uint32_t)0x00000080)
• #define GPIO_BSRR_BS_8 ((uint32_t)0x00000100)
• #define GPIO_BSRR_BS_9 ((uint32_t)0x00000200)
• #define GPIO_BSRR_BS_10 ((uint32_t)0x00000400)
• #define GPIO_BSRR_BS_11 ((uint32_t)0x00000800)
• #define GPIO_BSRR_BS_12 ((uint32_t)0x00001000)
• #define GPIO_BSRR_BS_13 ((uint32_t)0x00002000)
• #define GPIO_BSRR_BS_14 ((uint32_t)0x00004000)
• #define GPIO_BSRR_BS_15 ((uint32_t)0x00008000)
• #define GPIO_BSRR_BR_0 ((uint32_t)0x00010000)
• #define GPIO_BSRR_BR_1 ((uint32_t)0x00020000)
• #define GPIO_BSRR_BR_2 ((uint32_t)0x00040000)
• #define GPIO_BSRR_BR_3 ((uint32_t)0x00080000)
• #define GPIO_BSRR_BR_4 ((uint32_t)0x00100000)
• #define GPIO_BSRR_BR_5 ((uint32_t)0x00200000)
• #define GPIO_BSRR_BR_6 ((uint32_t)0x00400000)
• #define GPIO_BSRR_BR_7 ((uint32_t)0x00800000)
• #define GPIO_BSRR_BR_8 ((uint32_t)0x01000000)
• #define GPIO_BSRR_BR_9 ((uint32_t)0x02000000)
• #define GPIO_BSRR_BR_10 ((uint32_t)0x04000000)
• #define GPIO_BSRR_BR_11 ((uint32_t)0x08000000)
• #define GPIO_BSRR_BR_12 ((uint32_t)0x10000000)
• #define GPIO_BSRR_BR_13 ((uint32_t)0x20000000)
• #define GPIO_BSRR_BR_14 ((uint32_t)0x40000000)
• #define GPIO_BSRR_BR_15 ((uint32_t)0x80000000)
• #define HASH_CR_INIT ((uint32_t)0x00000004)
• #define HASH_CR_DMAE ((uint32_t)0x00000008)
• #define HASH_CR_DATATYPE ((uint32_t)0x00000030)
• #define HASH_CR_DATATYPE_0 ((uint32_t)0x00000010)
• #define HASH_CR_DATATYPE_1 ((uint32_t)0x00000020)
• #define HASH_CR_MODE ((uint32_t)0x00000040)
```

```
• #define HASH_CR_ALGO ((uint32_t)0x00040080)
• #define HASH_CR_ALGO_0 ((uint32_t)0x00000080)
• #define HASH_CR_ALGO_1 ((uint32_t)0x00040000)
• #define HASH_CR_NBW ((uint32_t)0x00000F00)
• #define HASH_CR_NBW_0 ((uint32_t)0x00000100)
• #define HASH_CR_NBW_1 ((uint32_t)0x00000200)
• #define HASH_CR_NBW_2 ((uint32_t)0x00000400)
• #define HASH_CR_NBW_3 ((uint32_t)0x00000800)
• #define HASH_CR_DINNE ((uint32_t)0x00001000)
• #define HASH_CR_MDMAT ((uint32_t)0x00002000)
• #define HASH_CR_LKEY ((uint32_t)0x00010000)
• #define HASH_STR_NBW ((uint32_t)0x0000001F)
• #define HASH_STR_NBW_0 ((uint32_t)0x00000001)
• #define HASH_STR_NBW_1 ((uint32_t)0x00000002)
• #define HASH_STR_NBW_2 ((uint32_t)0x00000004)
• #define HASH_STR_NBW_3 ((uint32_t)0x00000008)
• #define HASH_STR_NBW_4 ((uint32_t)0x00000010)
• #define HASH_STR_DCAL ((uint32_t)0x00000100)
• #define HASH_IMR_DINIM ((uint32_t)0x00000001)
• #define HASH_IMR_DCIM ((uint32_t)0x00000002)
• #define HASH_SR_DINIS ((uint32_t)0x00000001)
• #define HASH_SR_DCIS ((uint32_t)0x00000002)
• #define HASH_SR_DMAS ((uint32_t)0x00000004)
• #define HASH_SR_BUSY ((uint32_t)0x00000008)
• #define I2C_CR1_PE ((uint16_t)0x0001)
• #define I2C_CR1_SMBUS ((uint16_t)0x0002)
• #define I2C_CR1_SMBTYPE ((uint16_t)0x0008)
• #define I2C_CR1_ENARP ((uint16_t)0x0010)
• #define I2C_CR1_ENPEC ((uint16_t)0x0020)
• #define I2C_CR1_ENGC ((uint16_t)0x0040)
• #define I2C_CR1_NOSTRETCH ((uint16_t)0x0080)
• #define I2C_CR1_START ((uint16_t)0x0100)
• #define I2C_CR1_STOP ((uint16_t)0x0200)
• #define I2C_CR1_ACK ((uint16_t)0x0400)
• #define I2C_CR1_POS ((uint16_t)0x0800)
• #define I2C_CR1_PEC ((uint16_t)0x1000)
• #define I2C_CR1_ALERT ((uint16_t)0x2000)
• #define I2C_CR1_SWRST ((uint16_t)0x8000)
• #define I2C_CR2_FREQ ((uint16_t)0x003F)
• #define I2C_CR2_FREQ_0 ((uint16_t)0x0001)
• #define I2C_CR2_FREQ_1 ((uint16_t)0x0002)
• #define I2C_CR2_FREQ_2 ((uint16_t)0x0004)
• #define I2C_CR2_FREQ_3 ((uint16_t)0x0008)
• #define I2C_CR2_FREQ_4 ((uint16_t)0x0010)
• #define I2C_CR2_FREQ_5 ((uint16_t)0x0020)
• #define I2C_CR2_IERREN ((uint16_t)0x0100)
• #define I2C_CR2_IETVTEN ((uint16_t)0x0200)
• #define I2C_CR2_ITBUFEN ((uint16_t)0x0400)
• #define I2C_CR2_DMAEN ((uint16_t)0x0800)
• #define I2C_CR2_LAST ((uint16_t)0x1000)
• #define I2C_OAR1_ADD1_7 ((uint16_t)0x00FE)
• #define I2C_OAR1_ADD8_9 ((uint16_t)0x0300)
• #define I2C_OAR1_ADD0 ((uint16_t)0x0001)
• #define I2C_OAR1_ADD1 ((uint16_t)0x0002)
• #define I2C_OAR1_ADD2 ((uint16_t)0x0004)
```

- #define I2C_OAR1_ADD3 ((uint16_t)0x0008)
- #define I2C_OAR1_ADD4 ((uint16_t)0x0010)
- #define I2C_OAR1_ADD5 ((uint16_t)0x0020)
- #define I2C_OAR1_ADD6 ((uint16_t)0x0040)
- #define I2C_OAR1_ADD7 ((uint16_t)0x0080)
- #define I2C_OAR1_ADD8 ((uint16_t)0x0100)
- #define I2C_OAR1_ADD9 ((uint16_t)0x0200)
- #define I2C_OAR1_ADDMODE ((uint16_t)0x8000)
- #define I2C_OAR2_ENDUAL ((uint8_t)0x01)
- #define I2C_OAR2_ADD2 ((uint8_t)0xFE)
- #define I2C_DR_DR ((uint8_t)0xFF)
- #define I2C_SR1_SB ((uint16_t)0x0001)
- #define I2C_SR1_ADDR ((uint16_t)0x0002)
- #define I2C_SR1_BTF ((uint16_t)0x0004)
- #define I2C_SR1_ADD10 ((uint16_t)0x0008)
- #define I2C_SR1_STOPF ((uint16_t)0x0010)
- #define I2C_SR1_RXNE ((uint16_t)0x0040)
- #define I2C_SR1_TXE ((uint16_t)0x0080)
- #define I2C_SR1_BERR ((uint16_t)0x0100)
- #define I2C_SR1_ARLO ((uint16_t)0x0200)
- #define I2C_SR1_AF ((uint16_t)0x0400)
- #define I2C_SR1_OVR ((uint16_t)0x0800)
- #define I2C_SR1_PECERR ((uint16_t)0x1000)
- #define I2C_SR1_TIMEOUT ((uint16_t)0x4000)
- #define I2C_SR1_SMBALERT ((uint16_t)0x8000)
- #define I2C_SR2_MSL ((uint16_t)0x0001)
- #define I2C_SR2_BUSY ((uint16_t)0x0002)
- #define I2C_SR2_TRA ((uint16_t)0x0004)
- #define I2C_SR2_GENCALL ((uint16_t)0x0010)
- #define I2C_SR2_SMBDEFAULT ((uint16_t)0x0020)
- #define I2C_SR2_SMBHOST ((uint16_t)0x0040)
- #define I2C_SR2_DUALF ((uint16_t)0x0080)
- #define I2C_SR2_PEC ((uint16_t)0xFF00)
- #define I2C_CCR_CCR ((uint16_t)0xFFFF)
- #define I2C_CCR_DUTY ((uint16_t)0x4000)
- #define I2C_CCR_FS ((uint16_t)0x8000)
- #define I2C_TRISE_TRISE ((uint8_t)0x3F)
- #define I2C_FLTR_DNF ((uint8_t)0x0F)
- #define I2C_FLTR_ANOFF ((uint8_t)0x10)
- #define IWDG_KR_KEY ((uint16_t)0xFFFF)
- #define IWDG_PR_PR ((uint8_t)0x07)
- #define IWDG_PR_PR_0 ((uint8_t)0x01)
- #define IWDG_PR_PR_1 ((uint8_t)0x02)
- #define IWDG_PR_PR_2 ((uint8_t)0x04)
- #define IWDG_RLR_RL ((uint16_t)0xFFFF)
- #define IWDG_SR_PVU ((uint8_t)0x01)
- #define IWDG_SR_RVU ((uint8_t)0x02)
- #define LTDC_SSCR_VSH ((uint32_t)0x0000007FF)
- #define LTDC_SSCR_HSW ((uint32_t)0x0FFF0000)
- #define LTDC_BPCR_AVBP ((uint32_t)0x0000007FF)
- #define LTDC_BPCR_AHBP ((uint32_t)0x0FFF0000)
- #define LTDC_AWCR_AAH ((uint32_t)0x0000007FF)
- #define LTDC_AWCR_AAW ((uint32_t)0x0FFF0000)
- #define LTDC_TWCR_TOTALH ((uint32_t)0x0000007FF)
- #define LTDC_TWCR_TOTALW ((uint32_t)0x0FFF0000)

```
• #define LTDC_GCR_LTDCEN ((uint32_t)0x00000001)
• #define LTDC_GCR_DBW ((uint32_t)0x00000070)
• #define LTDC_GCR_DGW ((uint32_t)0x00000700)
• #define LTDC_GCR_DRW ((uint32_t)0x00007000)
• #define LTDC_GCR_DEN ((uint32_t)0x00010000)
• #define LTDC_GCR_PCPOL ((uint32_t)0x10000000)
• #define LTDC_GCR_DEPOL ((uint32_t)0x20000000)
• #define LTDC_GCR_VSPOL ((uint32_t)0x40000000)
• #define LTDC_GCR_HSPOL ((uint32_t)0x80000000)
• #define LTDC_GCR_DTEN LTDC_GCR_DEN
• #define LTDC_SRCR_IMR ((uint32_t)0x00000001)
• #define LTDC_SRCR_VBR ((uint32_t)0x00000002)
• #define LTDC_BCCR_BCBLUE ((uint32_t)0x000000FF)
• #define LTDC_BCCR_BCGREEN ((uint32_t)0x0000FF00)
• #define LTDC_BCCR_BCRED ((uint32_t)0x00FF0000)
• #define LTDC_IER_LIE ((uint32_t)0x00000001)
• #define LTDC_IER_FUIE ((uint32_t)0x00000002)
• #define LTDC_IER_TERRIE ((uint32_t)0x00000004)
• #define LTDC_IER_RRIE ((uint32_t)0x00000008)
• #define LTDC_ISR_LIF ((uint32_t)0x00000001)
• #define LTDC_ISR_FUIF ((uint32_t)0x00000002)
• #define LTDC_ISR_TERRIF ((uint32_t)0x00000004)
• #define LTDC_ISR_RRIIF ((uint32_t)0x00000008)
• #define LTDC_ICR_CLIF ((uint32_t)0x00000001)
• #define LTDC_ICR_CFUIF ((uint32_t)0x00000002)
• #define LTDC_ICR_CTERRIF ((uint32_t)0x00000004)
• #define LTDC_ICR_CRRIF ((uint32_t)0x00000008)
• #define LTDC_LPCR_LPOS ((uint32_t)0x000007FF)
• #define LTDC_CPSR_CYPOS ((uint32_t)0x0000FFFF)
• #define LTDC_CPSR_CXPOS ((uint32_t)0xFFFF0000)
• #define LTDC_CDSR_VDES ((uint32_t)0x00000001)
• #define LTDC_CDSR_HDES ((uint32_t)0x00000002)
• #define LTDC_CDSR_VSYNCs ((uint32_t)0x00000004)
• #define LTDC_CDSR_HSYNCs ((uint32_t)0x00000008)
• #define LTDC_LxCR_LEN ((uint32_t)0x00000001)
• #define LTDC_LxCR_COLKEN ((uint32_t)0x00000002)
• #define LTDC_LxCR_CLUTEN ((uint32_t)0x00000010)
• #define LTDC_LxWHPCR_WHSTPOS ((uint32_t)0x00000FFF)
• #define LTDC_LxWHPCR_WHSPPOS ((uint32_t)0xFFFF0000)
• #define LTDC_LxWVPCR_WVSTPOS ((uint32_t)0x00000FFF)
• #define LTDC_LxWVPCR_WVSPPOS ((uint32_t)0xFFFF0000)
• #define LTDC_LxCKCR_CKBLUE ((uint32_t)0x000000FF)
• #define LTDC_LxCKCR_CKGREEN ((uint32_t)0x0000FF00)
• #define LTDC_LxCKCR_CKRED ((uint32_t)0x00FF0000)
• #define LTDC_LxPFCR_PF ((uint32_t)0x00000007)
• #define LTDC_LxCACR_CONSTA ((uint32_t)0x000000FF)
• #define LTDC_LxDCCR_DCBLUE ((uint32_t)0x000000FF)
• #define LTDC_LxDCCR_DCGREEN ((uint32_t)0x0000FF00)
• #define LTDC_LxDCCR_DCRED ((uint32_t)0x00FF0000)
• #define LTDC_LxDCCR_DCALPHA ((uint32_t)0xFF000000)
• #define LTDC_LxBFCR_BF2 ((uint32_t)0x00000007)
• #define LTDC_LxBFCR_BF1 ((uint32_t)0x00000700)
• #define LTDC_LxCFBAR_CFBADD ((uint32_t)0xFFFFFFFF)
• #define LTDC_LxCFBLR_CFBLL ((uint32_t)0x00001FFF)
• #define LTDC_LxCFBLR_CFBP ((uint32_t)0x1FFF0000)
```

- #define LTDC_LxCFBLNR_CFBLNBR ((uint32_t)0x000007FF)
- #define LTDC_LxCLUTWR_BLUE ((uint32_t)0x000000FF)
- #define LTDC_LxCLUTWR_GREEN ((uint32_t)0x0000FF00)
- #define LTDC_LxCLUTWR_RED ((uint32_t)0x00FF0000)
- #define LTDC_LxCLUTWR_CLUTADD ((uint32_t)0xFF000000)
- #define PWR_CR_LPDS ((uint32_t)0x00000001)
- #define PWR_CR_PDDS ((uint32_t)0x00000002)
- #define PWR_CR_CWUF ((uint32_t)0x00000004)
- #define PWR_CR_CSBF ((uint32_t)0x00000008)
- #define PWR_CR_PVDE ((uint32_t)0x00000010)
- #define PWR_CR_PLS ((uint32_t)0x000000E0)
- #define PWR_CR_PLS_0 ((uint32_t)0x00000020)
- #define PWR_CR_PLS_1 ((uint32_t)0x00000040)
- #define PWR_CR_PLS_2 ((uint32_t)0x00000080)
- #define PWR_CR_PLS_LEV0 ((uint32_t)0x00000000)
- #define PWR_CR_PLS_LEV1 ((uint32_t)0x00000020)
- #define PWR_CR_PLS_LEV2 ((uint32_t)0x00000040)
- #define PWR_CR_PLS_LEV3 ((uint32_t)0x00000060)
- #define PWR_CR_PLS_LEV4 ((uint32_t)0x00000080)
- #define PWR_CR_PLS_LEV5 ((uint32_t)0x000000A0)
- #define PWR_CR_PLS_LEV6 ((uint32_t)0x000000C0)
- #define PWR_CR_PLS_LEV7 ((uint32_t)0x000000E0)
- #define PWR_CR_DBP ((uint32_t)0x00000100)
- #define PWR_CR_FPDS ((uint32_t)0x00000200)
- #define PWR_CR_LPUDS ((uint32_t)0x00000400)
- #define PWR_CR_MRUDS ((uint32_t)0x00000800)
- #define PWR_CR_LPLVDS ((uint32_t)0x00000400)
- #define PWR_CR_MRLVDS ((uint32_t)0x00000800)
- #define PWR_CR_ADCDC1 ((uint32_t)0x00002000)
- #define PWR_CR_VOS ((uint32_t)0x0000C000)
- #define PWR_CR_VOS_0 ((uint32_t)0x00004000)
- #define PWR_CR_VOS_1 ((uint32_t)0x00008000)
- #define PWR_CR_ODEN ((uint32_t)0x00010000)
- #define PWR_CR_ODSWEN ((uint32_t)0x00020000)
- #define PWR_CR_UDEN ((uint32_t)0x000C0000)
- #define PWR_CR_UDEN_0 ((uint32_t)0x00040000)
- #define PWR_CR_UDEN_1 ((uint32_t)0x00080000)
- #define PWR_CR_FMSSR ((uint32_t)0x00100000)
- #define PWR_CR_FISSR ((uint32_t)0x00200000)
- #define PWR_CR_PMODE PWR_CR_VOS
- #define PWR_CSR_WUF ((uint32_t)0x00000001)
- #define PWR_CSR_SBF ((uint32_t)0x00000002)
- #define PWR_CSR_PVDO ((uint32_t)0x00000004)
- #define PWR_CSR_BRR ((uint32_t)0x00000008)
- #define PWR_CSR_WUPP ((uint32_t)0x00000080)
- #define PWR_CSR_EWUP ((uint32_t)0x00000100)
- #define PWR_CSR_BRE ((uint32_t)0x00000200)
- #define PWR_CSR_VOSRDY ((uint32_t)0x00004000)
- #define PWR_CSR_ODRDY ((uint32_t)0x00010000)
- #define PWR_CSR_ODSWRDY ((uint32_t)0x00020000)
- #define PWR_CSR_UDSWRDY ((uint32_t)0x000C0000)
- #define PWR_CSR_REGRDY PWR_CSR_VOSRDY
- #define RCC_CR_HSION ((uint32_t)0x00000001)
- #define RCC_CR_HSIRDY ((uint32_t)0x00000002)
- #define RCC_CR_HSITRIM ((uint32_t)0x000000F8)

- #define **RCC_CR_HSITRIM_0** ((uint32_t)0x00000008)
- #define **RCC_CR_HSITRIM_1** ((uint32_t)0x00000010)
- #define **RCC_CR_HSITRIM_2** ((uint32_t)0x00000020)
- #define **RCC_CR_HSITRIM_3** ((uint32_t)0x00000040)
- #define **RCC_CR_HSITRIM_4** ((uint32_t)0x00000080)
- #define **RCC_CR_HSICAL** ((uint32_t)0x0000FF00)
- #define **RCC_CR_HSICAL_0** ((uint32_t)0x00000100)
- #define **RCC_CR_HSICAL_1** ((uint32_t)0x00000200)
- #define **RCC_CR_HSICAL_2** ((uint32_t)0x00000400)
- #define **RCC_CR_HSICAL_3** ((uint32_t)0x00000800)
- #define **RCC_CR_HSICAL_4** ((uint32_t)0x00001000)
- #define **RCC_CR_HSICAL_5** ((uint32_t)0x00002000)
- #define **RCC_CR_HSICAL_6** ((uint32_t)0x00004000)
- #define **RCC_CR_HSICAL_7** ((uint32_t)0x00008000)
- #define **RCC_CR_HSEON** ((uint32_t)0x00010000)
- #define **RCC_CR_HSERDY** ((uint32_t)0x00020000)
- #define **RCC_CR_HSEBYP** ((uint32_t)0x00040000)
- #define **RCC_CR_CSSON** ((uint32_t)0x00080000)
- #define **RCC_CR_PLLON** ((uint32_t)0x01000000)
- #define **RCC_CR_PLLRDY** ((uint32_t)0x02000000)
- #define **RCC_CR_PLLI2SON** ((uint32_t)0x04000000)
- #define **RCC_CR_PLLI2SRDY** ((uint32_t)0x08000000)
- #define **RCC_CR_PLLSAION** ((uint32_t)0x10000000)
- #define **RCC_CR_PLLSAIRDY** ((uint32_t)0x20000000)
- #define **RCC_PLLCFG_R_PLLM** ((uint32_t)0x0000003F)
- #define **RCC_PLLCFG_R_PLLM_0** ((uint32_t)0x00000001)
- #define **RCC_PLLCFG_R_PLLM_1** ((uint32_t)0x00000002)
- #define **RCC_PLLCFG_R_PLLM_2** ((uint32_t)0x00000004)
- #define **RCC_PLLCFG_R_PLLM_3** ((uint32_t)0x00000008)
- #define **RCC_PLLCFG_R_PLLM_4** ((uint32_t)0x00000010)
- #define **RCC_PLLCFG_R_PLLM_5** ((uint32_t)0x00000020)
- #define **RCC_PLLCFG_R_PLLN** ((uint32_t)0x00007FC0)
- #define **RCC_PLLCFG_R_PLLN_0** ((uint32_t)0x00000040)
- #define **RCC_PLLCFG_R_PLLN_1** ((uint32_t)0x00000080)
- #define **RCC_PLLCFG_R_PLLN_2** ((uint32_t)0x00000100)
- #define **RCC_PLLCFG_R_PLLN_3** ((uint32_t)0x00000200)
- #define **RCC_PLLCFG_R_PLLN_4** ((uint32_t)0x00000400)
- #define **RCC_PLLCFG_R_PLLN_5** ((uint32_t)0x00000800)
- #define **RCC_PLLCFG_R_PLLN_6** ((uint32_t)0x00001000)
- #define **RCC_PLLCFG_R_PLLN_7** ((uint32_t)0x00002000)
- #define **RCC_PLLCFG_R_PLLN_8** ((uint32_t)0x00004000)
- #define **RCC_PLLCFG_R_PLLP** ((uint32_t)0x00030000)
- #define **RCC_PLLCFG_R_PLLP_0** ((uint32_t)0x00010000)
- #define **RCC_PLLCFG_R_PLLP_1** ((uint32_t)0x00020000)
- #define **RCC_PLLCFG_R_PLLSRC** ((uint32_t)0x00400000)
- #define **RCC_PLLCFG_R_PLLSRC_HSE** ((uint32_t)0x00400000)
- #define **RCC_PLLCFG_R_PLLSRC_HSI** ((uint32_t)0x00000000)
- #define **RCC_PLLCFG_R_PLLQ** ((uint32_t)0x0F000000)
- #define **RCC_PLLCFG_R_PLLQ_0** ((uint32_t)0x01000000)
- #define **RCC_PLLCFG_R_PLLQ_1** ((uint32_t)0x02000000)
- #define **RCC_PLLCFG_R_PLLQ_2** ((uint32_t)0x04000000)
- #define **RCC_PLLCFG_R_PLLQ_3** ((uint32_t)0x08000000)
- #define **RCC_CFGR_SW** ((uint32_t)0x00000003)
- #define **RCC_CFGR_SW_0** ((uint32_t)0x00000001)
- #define **RCC_CFGR_SW_1** ((uint32_t)0x00000002)

- #define **RCC_CFGR_SW_HSI** ((uint32_t)0x00000000)
- #define **RCC_CFGR_SW_HSE** ((uint32_t)0x00000001)
- #define **RCC_CFGR_SW_PLL** ((uint32_t)0x00000002)
- #define **RCC_CFGR_SWS** ((uint32_t)0x0000000C)
- #define **RCC_CFGR_SWS_0** ((uint32_t)0x00000004)
- #define **RCC_CFGR_SWS_1** ((uint32_t)0x00000008)
- #define **RCC_CFGR_SWS_HSI** ((uint32_t)0x00000000)
- #define **RCC_CFGR_SWS_HSE** ((uint32_t)0x00000004)
- #define **RCC_CFGR_SWS_PLL** ((uint32_t)0x00000008)
- #define **RCC_CFGR_HPRE** ((uint32_t)0x000000F0)
- #define **RCC_CFGR_HPRE_0** ((uint32_t)0x00000010)
- #define **RCC_CFGR_HPRE_1** ((uint32_t)0x00000020)
- #define **RCC_CFGR_HPRE_2** ((uint32_t)0x00000040)
- #define **RCC_CFGR_HPRE_3** ((uint32_t)0x00000080)
- #define **RCC_CFGR_HPRE_DIV1** ((uint32_t)0x00000000)
- #define **RCC_CFGR_HPRE_DIV2** ((uint32_t)0x00000080)
- #define **RCC_CFGR_HPRE_DIV4** ((uint32_t)0x00000090)
- #define **RCC_CFGR_HPRE_DIV8** ((uint32_t)0x000000A0)
- #define **RCC_CFGR_HPRE_DIV16** ((uint32_t)0x000000B0)
- #define **RCC_CFGR_HPRE_DIV64** ((uint32_t)0x000000C0)
- #define **RCC_CFGR_HPRE_DIV128** ((uint32_t)0x000000D0)
- #define **RCC_CFGR_HPRE_DIV256** ((uint32_t)0x000000E0)
- #define **RCC_CFGR_HPRE_DIV512** ((uint32_t)0x000000F0)
- #define **RCC_CFGR_PPREG1** ((uint32_t)0x00001C00)
- #define **RCC_CFGR_PPREG1_0** ((uint32_t)0x00000400)
- #define **RCC_CFGR_PPREG1_1** ((uint32_t)0x00000800)
- #define **RCC_CFGR_PPREG1_2** ((uint32_t)0x00001000)
- #define **RCC_CFGR_PPREG1_DIV1** ((uint32_t)0x00000000)
- #define **RCC_CFGR_PPREG1_DIV2** ((uint32_t)0x00001000)
- #define **RCC_CFGR_PPREG1_DIV4** ((uint32_t)0x00001400)
- #define **RCC_CFGR_PPREG1_DIV8** ((uint32_t)0x00001800)
- #define **RCC_CFGR_PPREG1_DIV16** ((uint32_t)0x00001C00)
- #define **RCC_CFGR_PPREG2** ((uint32_t)0x0000E000)
- #define **RCC_CFGR_PPREG2_0** ((uint32_t)0x00002000)
- #define **RCC_CFGR_PPREG2_1** ((uint32_t)0x00004000)
- #define **RCC_CFGR_PPREG2_2** ((uint32_t)0x00008000)
- #define **RCC_CFGR_PPREG2_DIV1** ((uint32_t)0x00000000)
- #define **RCC_CFGR_PPREG2_DIV2** ((uint32_t)0x00008000)
- #define **RCC_CFGR_PPREG2_DIV4** ((uint32_t)0x0000A000)
- #define **RCC_CFGR_PPREG2_DIV8** ((uint32_t)0x0000C000)
- #define **RCC_CFGR_PPREG2_DIV16** ((uint32_t)0x0000E000)
- #define **RCC_CFGR_RTCPPRE** ((uint32_t)0x001F0000)
- #define **RCC_CFGR_RTCPPRE_0** ((uint32_t)0x00010000)
- #define **RCC_CFGR_RTCPPRE_1** ((uint32_t)0x00020000)
- #define **RCC_CFGR_RTCPPRE_2** ((uint32_t)0x00040000)
- #define **RCC_CFGR_RTCPPRE_3** ((uint32_t)0x00080000)
- #define **RCC_CFGR_RTCPPRE_4** ((uint32_t)0x00100000)
- #define **RCC_CFGR_MCO1** ((uint32_t)0x00600000)
- #define **RCC_CFGR_MCO1_0** ((uint32_t)0x00200000)
- #define **RCC_CFGR_MCO1_1** ((uint32_t)0x00400000)
- #define **RCC_CFGR_I2SSRC** ((uint32_t)0x00800000)
- #define **RCC_CFGR_MCO1PRE** ((uint32_t)0x07000000)
- #define **RCC_CFGR_MCO1PRE_0** ((uint32_t)0x01000000)
- #define **RCC_CFGR_MCO1PRE_1** ((uint32_t)0x02000000)
- #define **RCC_CFGR_MCO1PRE_2** ((uint32_t)0x04000000)

```
• #define RCC_CFGR_MCO2PRE ((uint32_t)0x38000000)
• #define RCC_CFGR_MCO2PRE_0 ((uint32_t)0x08000000)
• #define RCC_CFGR_MCO2PRE_1 ((uint32_t)0x10000000)
• #define RCC_CFGR_MCO2PRE_2 ((uint32_t)0x20000000)
• #define RCC_CFGR_MCO2 ((uint32_t)0xC0000000)
• #define RCC_CFGR_MCO2_0 ((uint32_t)0x40000000)
• #define RCC_CFGR_MCO2_1 ((uint32_t)0x80000000)
• #define RCC_CIR_LSIRDYF ((uint32_t)0x00000001)
• #define RCC_CIR_LSERDYF ((uint32_t)0x00000002)
• #define RCC_CIR_HSIRDYF ((uint32_t)0x00000004)
• #define RCC_CIR_HSERDYF ((uint32_t)0x00000008)
• #define RCC_CIR_PLLRDYF ((uint32_t)0x00000010)
• #define RCC_CIR_PLLI2SRDYF ((uint32_t)0x00000020)
• #define RCC_CIR_PLLSAIRDYF ((uint32_t)0x00000040)
• #define RCC_CIR_CSSF ((uint32_t)0x00000080)
• #define RCC_CIR_LSIRDYIE ((uint32_t)0x00000100)
• #define RCC_CIR_LSERDYIE ((uint32_t)0x00000200)
• #define RCC_CIR_HSIRDYIE ((uint32_t)0x00000400)
• #define RCC_CIR_HSERDYIE ((uint32_t)0x00000800)
• #define RCC_CIR_PLLRDYIE ((uint32_t)0x00001000)
• #define RCC_CIR_PLLI2SRDYIE ((uint32_t)0x00002000)
• #define RCC_CIR_PLLSAIRDYIE ((uint32_t)0x00004000)
• #define RCC_CIR_LSIRDYC ((uint32_t)0x00010000)
• #define RCC_CIR_LSERDYC ((uint32_t)0x00020000)
• #define RCC_CIR_HSIRDYC ((uint32_t)0x00040000)
• #define RCC_CIR_HSERDYC ((uint32_t)0x00080000)
• #define RCC_CIR_PLLRDYC ((uint32_t)0x00100000)
• #define RCC_CIR_PLLI2SRDYC ((uint32_t)0x00200000)
• #define RCC_CIR_PLLSAIRDYC ((uint32_t)0x00400000)
• #define RCC_CIR_CSSC ((uint32_t)0x00800000)
• #define RCC_AHB1RSTR_GPIOARST ((uint32_t)0x00000001)
• #define RCC_AHB1RSTR_GPIOBRST ((uint32_t)0x00000002)
• #define RCC_AHB1RSTR_GPIOCRST ((uint32_t)0x00000004)
• #define RCC_AHB1RSTR_GPIODRST ((uint32_t)0x00000008)
• #define RCC_AHB1RSTR_GPIOERST ((uint32_t)0x00000010)
• #define RCC_AHB1RSTR_GPIOFRST ((uint32_t)0x00000020)
• #define RCC_AHB1RSTR_GPIORST ((uint32_t)0x00000040)
• #define RCC_AHB1RSTR_GPIOHRST ((uint32_t)0x00000080)
• #define RCC_AHB1RSTR_GPIOIRST ((uint32_t)0x00000100)
• #define RCC_AHB1RSTR_GPIOJRST ((uint32_t)0x00000200)
• #define RCC_AHB1RSTR_GPIOKRST ((uint32_t)0x00000400)
• #define RCC_AHB1RSTR_CRCRST ((uint32_t)0x00001000)
• #define RCC_AHB1RSTR_DMA1RST ((uint32_t)0x00200000)
• #define RCC_AHB1RSTR_DMA2RST ((uint32_t)0x00400000)
• #define RCC_AHB1RSTR_DMA2DRST ((uint32_t)0x00800000)
• #define RCC_AHB1RSTR_ETHMACRST ((uint32_t)0x02000000)
• #define RCC_AHB1RSTR_OTGHRST ((uint32_t)0x10000000)
• #define RCC_AHB2RSTR_DCMIRST ((uint32_t)0x00000001)
• #define RCC_AHB2RSTR_CRYPRST ((uint32_t)0x00000010)
• #define RCC_AHB2RSTR_HASHRST ((uint32_t)0x00000020)
• #define RCC_AHB2RSTR_HSAHRST RCC_AHB2RSTR_HASHRST
• #define RCC_AHB2RSTR_RNGRST ((uint32_t)0x00000040)
• #define RCC_AHB2RSTR_OTGFSRST ((uint32_t)0x00000080)
• #define RCC_APB1RSTR_TIM2RST ((uint32_t)0x00000001)
• #define RCC_APB1RSTR_TIM3RST ((uint32_t)0x00000002)
```

- #define **RCC_APB1RSTR_TIM4RST** ((uint32_t)0x00000004)
- #define **RCC_APB1RSTR_TIM5RST** ((uint32_t)0x00000008)
- #define **RCC_APB1RSTR_TIM6RST** ((uint32_t)0x00000010)
- #define **RCC_APB1RSTR_TIM7RST** ((uint32_t)0x00000020)
- #define **RCC_APB1RSTR_TIM12RST** ((uint32_t)0x00000040)
- #define **RCC_APB1RSTR_TIM13RST** ((uint32_t)0x00000080)
- #define **RCC_APB1RSTR_TIM14RST** ((uint32_t)0x00000100)
- #define **RCC_APB1RSTR_WWDGRST** ((uint32_t)0x00000800)
- #define **RCC_APB1RSTR_SPI2RST** ((uint32_t)0x00004000)
- #define **RCC_APB1RSTR_SPI3RST** ((uint32_t)0x00008000)
- #define **RCC_APB1RSTR_USART2RST** ((uint32_t)0x00020000)
- #define **RCC_APB1RSTR_USART3RST** ((uint32_t)0x00040000)
- #define **RCC_APB1RSTR_UART4RST** ((uint32_t)0x00080000)
- #define **RCC_APB1RSTR_UART5RST** ((uint32_t)0x00100000)
- #define **RCC_APB1RSTR_I2C1RST** ((uint32_t)0x00200000)
- #define **RCC_APB1RSTR_I2C2RST** ((uint32_t)0x00400000)
- #define **RCC_APB1RSTR_I2C3RST** ((uint32_t)0x00800000)
- #define **RCC_APB1RSTR_CAN1RST** ((uint32_t)0x02000000)
- #define **RCC_APB1RSTR_CAN2RST** ((uint32_t)0x04000000)
- #define **RCC_APB1RSTR_PWRRST** ((uint32_t)0x10000000)
- #define **RCC_APB1RSTR_DACRST** ((uint32_t)0x20000000)
- #define **RCC_APB1RSTR_UART7RST** ((uint32_t)0x40000000)
- #define **RCC_APB1RSTR_UART8RST** ((uint32_t)0x80000000)
- #define **RCC_APB2RSTR_TIM1RST** ((uint32_t)0x00000001)
- #define **RCC_APB2RSTR_TIM8RST** ((uint32_t)0x00000002)
- #define **RCC_APB2RSTR_USART1RST** ((uint32_t)0x00000010)
- #define **RCC_APB2RSTR_USART6RST** ((uint32_t)0x00000020)
- #define **RCC_APB2RSTR_UART9RST** ((uint32_t)0x00000040)
- #define **RCC_APB2RSTR_UART10RST** ((uint32_t)0x00000080)
- #define **RCC_APB2RSTR_ADCRST** ((uint32_t)0x00000100)
- #define **RCC_APB2RSTR_SDIORST** ((uint32_t)0x00000800)
- #define **RCC_APB2RSTR_SPI1RST** ((uint32_t)0x00001000)
- #define **RCC_APB2RSTR_SPI4RST** ((uint32_t)0x00002000)
- #define **RCC_APB2RSTR_SYSCFGRST** ((uint32_t)0x00004000)
- #define **RCC_APB2RSTR_TIM9RST** ((uint32_t)0x00010000)
- #define **RCC_APB2RSTR_TIM10RST** ((uint32_t)0x00020000)
- #define **RCC_APB2RSTR_TIM11RST** ((uint32_t)0x00040000)
- #define **RCC_APB2RSTR_SPI5RST** ((uint32_t)0x00100000)
- #define **RCC_APB2RSTR_SPI6RST** ((uint32_t)0x00200000)
- #define **RCC_APB2RSTR_SAI1RST** ((uint32_t)0x00400000)
- #define **RCC_APB2RSTR_LTDCRST** ((uint32_t)0x04000000)
- #define **RCC_APB2RSTR_SPI1** RCC_APB2RSTR_SPI1RST
- #define **RCC_APB2RSTR_DFSDMRST** RCC_APB2RSTR_DFSDM1RST
- #define **RCC_AHB1ENR_GPIOAEN** ((uint32_t)0x00000001)
- #define **RCC_AHB1ENR_GPIOBEN** ((uint32_t)0x00000002)
- #define **RCC_AHB1ENR_GPIOCEN** ((uint32_t)0x00000004)
- #define **RCC_AHB1ENR_GPIODEN** ((uint32_t)0x00000008)
- #define **RCC_AHB1ENR_GPIOEEN** ((uint32_t)0x00000010)
- #define **RCC_AHB1ENR_GPIOFEN** ((uint32_t)0x00000020)
- #define **RCC_AHB1ENR_GPIOGEN** ((uint32_t)0x00000040)
- #define **RCC_AHB1ENR_GPIOHEN** ((uint32_t)0x00000080)
- #define **RCC_AHB1ENR_GPIOIEN** ((uint32_t)0x00000100)
- #define **RCC_AHB1ENR_GPIOJEN** ((uint32_t)0x00000200)
- #define **RCC_AHB1ENR_GPIOKEN** ((uint32_t)0x00000400)
- #define **RCC_AHB1ENR_CRCEN** ((uint32_t)0x00001000)

- #define **RCC_AHB1ENR_BKPSRAMEN** ((uint32_t)0x00040000)
- #define **RCC_AHB1ENR_CCMMDATARAMEN** ((uint32_t)0x00100000)
- #define **RCC_AHB1ENR_DMA1EN** ((uint32_t)0x00200000)
- #define **RCC_AHB1ENR_DMA2EN** ((uint32_t)0x00400000)
- #define **RCC_AHB1ENR_DMA2DEN** ((uint32_t)0x00800000)
- #define **RCC_AHB1ENR_ETHMACEN** ((uint32_t)0x02000000)
- #define **RCC_AHB1ENR_ETHMACTXEN** ((uint32_t)0x04000000)
- #define **RCC_AHB1ENR_ETHMACRXEN** ((uint32_t)0x08000000)
- #define **RCC_AHB1ENR_ETHMACPTPEN** ((uint32_t)0x10000000)
- #define **RCC_AHB1ENR_OTGHSEN** ((uint32_t)0x20000000)
- #define **RCC_AHB1ENR_OTGHSULPIEN** ((uint32_t)0x40000000)
- #define **RCC_AHB2ENR_DCMIEN** ((uint32_t)0x00000001)
- #define **RCC_AHB2ENR_CRYPTEN** ((uint32_t)0x00000010)
- #define **RCC_AHB2ENR_HASHEN** ((uint32_t)0x00000020)
- #define **RCC_AHB2ENR RNGEN** ((uint32_t)0x00000040)
- #define **RCC_AHB2ENR_OTGFSEN** ((uint32_t)0x00000080)
- #define **RCC_APB1ENR_TIM2EN** ((uint32_t)0x00000001)
- #define **RCC_APB1ENR_TIM3EN** ((uint32_t)0x00000002)
- #define **RCC_APB1ENR_TIM4EN** ((uint32_t)0x00000004)
- #define **RCC_APB1ENR_TIM5EN** ((uint32_t)0x00000008)
- #define **RCC_APB1ENR_TIM6EN** ((uint32_t)0x00000010)
- #define **RCC_APB1ENR_TIM7EN** ((uint32_t)0x00000020)
- #define **RCC_APB1ENR_TIM12EN** ((uint32_t)0x00000040)
- #define **RCC_APB1ENR_TIM13EN** ((uint32_t)0x00000080)
- #define **RCC_APB1ENR_TIM14EN** ((uint32_t)0x00000100)
- #define **RCC_APB1ENR_WWDGEN** ((uint32_t)0x00000800)
- #define **RCC_APB1ENR_SPI2EN** ((uint32_t)0x00004000)
- #define **RCC_APB1ENR_SPI3EN** ((uint32_t)0x00008000)
- #define **RCC_APB1ENR_USART2EN** ((uint32_t)0x00020000)
- #define **RCC_APB1ENR_USART3EN** ((uint32_t)0x00040000)
- #define **RCC_APB1ENR_UART4EN** ((uint32_t)0x00080000)
- #define **RCC_APB1ENR_UART5EN** ((uint32_t)0x00100000)
- #define **RCC_APB1ENR_I2C1EN** ((uint32_t)0x00200000)
- #define **RCC_APB1ENR_I2C2EN** ((uint32_t)0x00400000)
- #define **RCC_APB1ENR_I2C3EN** ((uint32_t)0x00800000)
- #define **RCC_APB1ENR_CAN1EN** ((uint32_t)0x02000000)
- #define **RCC_APB1ENR_CAN2EN** ((uint32_t)0x04000000)
- #define **RCC_APB1ENR_PWREN** ((uint32_t)0x10000000)
- #define **RCC_APB1ENR_DACEN** ((uint32_t)0x20000000)
- #define **RCC_APB1ENR_UART7EN** ((uint32_t)0x40000000)
- #define **RCC_APB1ENR_UART8EN** ((uint32_t)0x80000000)
- #define **RCC_APB2ENR_TIM1EN** ((uint32_t)0x00000001)
- #define **RCC_APB2ENR_TIM8EN** ((uint32_t)0x00000002)
- #define **RCC_APB2ENR_USART1EN** ((uint32_t)0x00000010)
- #define **RCC_APB2ENR_USART6EN** ((uint32_t)0x00000020)
- #define **RCC_APB2ENR_UART9EN** ((uint32_t)0x00000040)
- #define **RCC_APB2ENR_UART10EN** ((uint32_t)0x00000080)
- #define **RCC_APB2ENR_ADC1EN** ((uint32_t)0x00000100)
- #define **RCC_APB2ENR_ADC2EN** ((uint32_t)0x00000200)
- #define **RCC_APB2ENR_ADC3EN** ((uint32_t)0x00000400)
- #define **RCC_APB2ENR_SDIOEN** ((uint32_t)0x00000800)
- #define **RCC_APB2ENR_SPI1EN** ((uint32_t)0x00001000)
- #define **RCC_APB2ENR_SPI4EN** ((uint32_t)0x00002000)
- #define **RCC_APB2ENR_SYSCFGEN** ((uint32_t)0x00004000)
- #define **RCC_APB2ENR_EXTIEN** ((uint32_t)0x00008000)

- #define **RCC_APB2ENR_TIM9EN** ((uint32_t)0x00010000)
- #define **RCC_APB2ENR_TIM10EN** ((uint32_t)0x00020000)
- #define **RCC_APB2ENR_TIM11EN** ((uint32_t)0x00040000)
- #define **RCC_APB2ENR_SPI5EN** ((uint32_t)0x00100000)
- #define **RCC_APB2ENR_SPI6EN** ((uint32_t)0x00200000)
- #define **RCC_APB2ENR_SAI1EN** ((uint32_t)0x00400000)
- #define **RCC_APB2ENR_LTDCEN** ((uint32_t)0x04000000)
- #define **RCC_AHB1LPENR_GPIOALPEN** ((uint32_t)0x00000001)
- #define **RCC_AHB1LPENR_GPIOBLPEN** ((uint32_t)0x00000002)
- #define **RCC_AHB1LPENR_GPIOCLPEN** ((uint32_t)0x00000004)
- #define **RCC_AHB1LPENR_GPIODLPEN** ((uint32_t)0x00000008)
- #define **RCC_AHB1LPENR_GPIOELPEN** ((uint32_t)0x00000010)
- #define **RCC_AHB1LPENR_GPIOFLPEN** ((uint32_t)0x00000020)
- #define **RCC_AHB1LPENR_GPIOGLPEN** ((uint32_t)0x00000040)
- #define **RCC_AHB1LPENR_GPIOHLPEN** ((uint32_t)0x00000080)
- #define **RCC_AHB1LPENR_GPIOILPEN** ((uint32_t)0x00000100)
- #define **RCC_AHB1LPENR_GPIOJLPEN** ((uint32_t)0x00000200)
- #define **RCC_AHB1LPENR_GPIOKLPEN** ((uint32_t)0x00000400)
- #define **RCC_AHB1LPENR_CRCLPEN** ((uint32_t)0x00001000)
- #define **RCC_AHB1LPENR_FLITFLPEN** ((uint32_t)0x00008000)
- #define **RCC_AHB1LPENR_SRAM1LPEN** ((uint32_t)0x00010000)
- #define **RCC_AHB1LPENR_SRAM2LPEN** ((uint32_t)0x00020000)
- #define **RCC_AHB1LPENR_BKPSRAMLPEN** ((uint32_t)0x00040000)
- #define **RCC_AHB1LPENR_SRAM3LPEN** ((uint32_t)0x00080000)
- #define **RCC_AHB1LPENR_DMA1LPEN** ((uint32_t)0x00200000)
- #define **RCC_AHB1LPENR_DMA2LPEN** ((uint32_t)0x00400000)
- #define **RCC_AHB1LPENR_DMA2DLPEN** ((uint32_t)0x00800000)
- #define **RCC_AHB1LPENR_ETHMACLPEN** ((uint32_t)0x02000000)
- #define **RCC_AHB1LPENR_ETHMACTXLPEN** ((uint32_t)0x04000000)
- #define **RCC_AHB1LPENR_ETHMACRXLPEN** ((uint32_t)0x08000000)
- #define **RCC_AHB1LPENR_ETHMACPTPLPEN** ((uint32_t)0x10000000)
- #define **RCC_AHB1LPENR_OTGHSLPEN** ((uint32_t)0x20000000)
- #define **RCC_AHB1LPENR_OTGHSULPILPEN** ((uint32_t)0x40000000)
- #define **RCC_AHB2LPENR_DCMILPEN** ((uint32_t)0x00000001)
- #define **RCC_AHB2LPENR_CRYPLPEN** ((uint32_t)0x00000010)
- #define **RCC_AHB2LPENR_HASHLPEN** ((uint32_t)0x00000020)
- #define **RCC_AHB2LPENR_RNGLPEN** ((uint32_t)0x00000040)
- #define **RCC_AHB2LPENR_OTGFSLPEN** ((uint32_t)0x00000080)
- #define **RCC_APB1LPENR_TIM2LPEN** ((uint32_t)0x00000001)
- #define **RCC_APB1LPENR_TIM3LPEN** ((uint32_t)0x00000002)
- #define **RCC_APB1LPENR_TIM4LPEN** ((uint32_t)0x00000004)
- #define **RCC_APB1LPENR_TIM5LPEN** ((uint32_t)0x00000008)
- #define **RCC_APB1LPENR_TIM6LPEN** ((uint32_t)0x00000010)
- #define **RCC_APB1LPENR_TIM7LPEN** ((uint32_t)0x00000020)
- #define **RCC_APB1LPENR_TIM12LPEN** ((uint32_t)0x00000040)
- #define **RCC_APB1LPENR_TIM13LPEN** ((uint32_t)0x00000080)
- #define **RCC_APB1LPENR_TIM14LPEN** ((uint32_t)0x00000100)
- #define **RCC_APB1LPENR_WWDGLPEN** ((uint32_t)0x00000800)
- #define **RCC_APB1LPENR_SPI2LPEN** ((uint32_t)0x00004000)
- #define **RCC_APB1LPENR_SPI3LPEN** ((uint32_t)0x00008000)
- #define **RCC_APB1LPENR_USART2LPEN** ((uint32_t)0x00020000)
- #define **RCC_APB1LPENR_USART3LPEN** ((uint32_t)0x00040000)
- #define **RCC_APB1LPENR_UART4LPEN** ((uint32_t)0x00080000)
- #define **RCC_APB1LPENR_UART5LPEN** ((uint32_t)0x00100000)
- #define **RCC_APB1LPENR_I2C1LPEN** ((uint32_t)0x00200000)

- #define **RCC_APB1LPENR_I2C2LPEN** ((uint32_t)0x00400000)
- #define **RCC_APB1LPENR_I2C3LPEN** ((uint32_t)0x00800000)
- #define **RCC_APB1LPENR_CAN1LPEN** ((uint32_t)0x02000000)
- #define **RCC_APB1LPENR_CAN2LPEN** ((uint32_t)0x04000000)
- #define **RCC_APB1LPENR_PWRLPEN** ((uint32_t)0x10000000)
- #define **RCC_APB1LPENR_DACLPEN** ((uint32_t)0x20000000)
- #define **RCC_APB1LPENR_UART7LPEN** ((uint32_t)0x40000000)
- #define **RCC_APB1LPENR_UART8LPEN** ((uint32_t)0x80000000)
- #define **RCC_APB2LPENR_TIM1LPEN** ((uint32_t)0x00000001)
- #define **RCC_APB2LPENR_TIM8LPEN** ((uint32_t)0x00000002)
- #define **RCC_APB2LPENR_USART1LPEN** ((uint32_t)0x00000010)
- #define **RCC_APB2LPENR_USART6LPEN** ((uint32_t)0x00000020)
- #define **RCC_APB2LPENR_UART9LPEN** ((uint32_t)0x00000040)
- #define **RCC_APB2LPENR_UART10LPEN** ((uint32_t)0x00000080)
- #define **RCC_APB2LPENR_ADC1LPEN** ((uint32_t)0x00000100)
- #define **RCC_APB2LPENR_ADC2PEN** ((uint32_t)0x00000200)
- #define **RCC_APB2LPENR_ADC3LPEN** ((uint32_t)0x00000400)
- #define **RCC_APB2LPENR_SDIOLPEN** ((uint32_t)0x00000800)
- #define **RCC_APB2LPENR_SPI1LPEN** ((uint32_t)0x00001000)
- #define **RCC_APB2LPENR_SPI4LPEN** ((uint32_t)0x00002000)
- #define **RCC_APB2LPENR_SYSCFGLPEN** ((uint32_t)0x00004000)
- #define **RCC_APB2LPENR_TIM9LPEN** ((uint32_t)0x00010000)
- #define **RCC_APB2LPENR_TIM10LPEN** ((uint32_t)0x00020000)
- #define **RCC_APB2LPENR_TIM11LPEN** ((uint32_t)0x00040000)
- #define **RCC_APB2LPENR_SPI5LPEN** ((uint32_t)0x00100000)
- #define **RCC_APB2LPENR_SPI6LPEN** ((uint32_t)0x00200000)
- #define **RCC_APB2LPENR_SAI1LPEN** ((uint32_t)0x00400000)
- #define **RCC_APB2LPENR_LTDCLPEN** ((uint32_t)0x04000000)
- #define **RCC_BDCR_LSEON** ((uint32_t)0x00000001)
- #define **RCC_BDCR_LSERDY** ((uint32_t)0x00000002)
- #define **RCC_BDCR_LSEBYP** ((uint32_t)0x00000004)
- #define **RCC_BDCR_LSEMOD** ((uint32_t)0x00000008)
- #define **RCC_BDCR_RTCSEL** ((uint32_t)0x00000300)
- #define **RCC_BDCR_RTCSEL_0** ((uint32_t)0x00000100)
- #define **RCC_BDCR_RTCSEL_1** ((uint32_t)0x00000200)
- #define **RCC_BDCR_RTCEN** ((uint32_t)0x00008000)
- #define **RCC_BDCR_BDRST** ((uint32_t)0x00010000)
- #define **RCC_CSR_LSION** ((uint32_t)0x00000001)
- #define **RCC_CSR_LSIRDY** ((uint32_t)0x00000002)
- #define **RCC_CSR_RMVF** ((uint32_t)0x01000000)
- #define **RCC_CSR_BORRSTF** ((uint32_t)0x02000000)
- #define **RCC_CSR_PADRSTF** ((uint32_t)0x04000000)
- #define **RCC_CSR_PORRSTF** ((uint32_t)0x08000000)
- #define **RCC_CSR_SFTRSTF** ((uint32_t)0x10000000)
- #define **RCC_CSR_WDGRSTF** ((uint32_t)0x20000000)
- #define **RCC_CSR_WWDGRSTF** ((uint32_t)0x40000000)
- #define **RCC_CSR_LPWRRSTF** ((uint32_t)0x80000000)
- #define **RCC_SSCGR_MODPER** ((uint32_t)0x00001FFF)
- #define **RCC_SSCGR_INCSTEP** ((uint32_t)0x0FFE000)
- #define **RCC_SSCGR_SPREADSEL** ((uint32_t)0x40000000)
- #define **RCC_SSCGR_SSCGEN** ((uint32_t)0x80000000)
- #define **RCC_PLLI2SCFGR_PLLI2SM** ((uint32_t)0x0000003F)
- #define **RCC_PLLI2SCFGR_PLLI2SM_0** ((uint32_t)0x00000001)
- #define **RCC_PLLI2SCFGR_PLLI2SM_1** ((uint32_t)0x00000002)
- #define **RCC_PLLI2SCFGR_PLLI2SM_2** ((uint32_t)0x00000004)

```
• #define RCC_PLLI2SCFGR_PLLI2SM_3 ((uint32_t)0x00000008)
• #define RCC_PLLI2SCFGR_PLLI2SM_4 ((uint32_t)0x00000010)
• #define RCC_PLLI2SCFGR_PLLI2SM_5 ((uint32_t)0x00000020)
• #define RCC_PLLI2SCFGR_PLLI2SN ((uint32_t)0x00007FC0)
• #define RCC_PLLI2SCFGR_PLLI2SN_0 ((uint32_t)0x00000040)
• #define RCC_PLLI2SCFGR_PLLI2SN_1 ((uint32_t)0x00000080)
• #define RCC_PLLI2SCFGR_PLLI2SN_2 ((uint32_t)0x00000100)
• #define RCC_PLLI2SCFGR_PLLI2SN_3 ((uint32_t)0x00000200)
• #define RCC_PLLI2SCFGR_PLLI2SN_4 ((uint32_t)0x00000400)
• #define RCC_PLLI2SCFGR_PLLI2SN_5 ((uint32_t)0x00000800)
• #define RCC_PLLI2SCFGR_PLLI2SN_6 ((uint32_t)0x00001000)
• #define RCC_PLLI2SCFGR_PLLI2SN_7 ((uint32_t)0x00002000)
• #define RCC_PLLI2SCFGR_PLLI2SN_8 ((uint32_t)0x00004000)
• #define RCC_PLLI2SCFGR_PLLI2SQ ((uint32_t)0x0F000000)
• #define RCC_PLLI2SCFGR_PLLI2SQ_0 ((uint32_t)0x01000000)
• #define RCC_PLLI2SCFGR_PLLI2SQ_1 ((uint32_t)0x02000000)
• #define RCC_PLLI2SCFGR_PLLI2SQ_2 ((uint32_t)0x04000000)
• #define RCC_PLLI2SCFGR_PLLI2SQ_3 ((uint32_t)0x08000000)
• #define RCC_PLLI2SCFGR_PLLI2SR ((uint32_t)0x70000000)
• #define RCC_PLLI2SCFGR_PLLI2SR_0 ((uint32_t)0x10000000)
• #define RCC_PLLI2SCFGR_PLLI2SR_1 ((uint32_t)0x20000000)
• #define RCC_PLLI2SCFGR_PLLI2SR_2 ((uint32_t)0x40000000)
• #define RCC_PLLSAICFGR_PLLSAIN ((uint32_t)0x00007FC0)
• #define RCC_PLLSAICFGR_PLLSAIN_0 ((uint32_t)0x00000040)
• #define RCC_PLLSAICFGR_PLLSAIN_1 ((uint32_t)0x00000080)
• #define RCC_PLLSAICFGR_PLLSAIN_2 ((uint32_t)0x00000100)
• #define RCC_PLLSAICFGR_PLLSAIN_3 ((uint32_t)0x00000200)
• #define RCC_PLLSAICFGR_PLLSAIN_4 ((uint32_t)0x00000400)
• #define RCC_PLLSAICFGR_PLLSAIN_5 ((uint32_t)0x00000800)
• #define RCC_PLLSAICFGR_PLLSAIN_6 ((uint32_t)0x00001000)
• #define RCC_PLLSAICFGR_PLLSAIN_7 ((uint32_t)0x00002000)
• #define RCC_PLLSAICFGR_PLLSAIN_8 ((uint32_t)0x00004000)
• #define RCC_PLLSAICFGR_PLLSAIQ ((uint32_t)0x0F000000)
• #define RCC_PLLSAICFGR_PLLSAIQ_0 ((uint32_t)0x01000000)
• #define RCC_PLLSAICFGR_PLLSAIQ_1 ((uint32_t)0x02000000)
• #define RCC_PLLSAICFGR_PLLSAIQ_2 ((uint32_t)0x04000000)
• #define RCC_PLLSAICFGR_PLLSAIQ_3 ((uint32_t)0x08000000)
• #define RCC_PLLSAICFGR_PLLSAIR ((uint32_t)0x70000000)
• #define RCC_PLLSAICFGR_PLLSAIR_0 ((uint32_t)0x10000000)
• #define RCC_PLLSAICFGR_PLLSAIR_1 ((uint32_t)0x20000000)
• #define RCC_PLLSAICFGR_PLLSAIR_2 ((uint32_t)0x40000000)
• #define RCC_DCKCFGGR_PLLI2SDIVQ ((uint32_t)0x0000001F)
• #define RCC_DCKCFGGR_PLLSAIDIVQ ((uint32_t)0x00001F00)
• #define RCC_DCKCFGGR_PLLSAIDIVR ((uint32_t)0x00030000)
• #define RCC_DCKCFGGR_SAI1ASRC ((uint32_t)0x00300000)
• #define RCC_DCKCFGGR_SAI1ASRC_0 ((uint32_t)0x00100000)
• #define RCC_DCKCFGGR_SAI1ASRC_1 ((uint32_t)0x00200000)
• #define RCC_DCKCFGGR_SAI1BSRC ((uint32_t)0x00C00000)
• #define RCC_DCKCFGGR_SAI1BSRC_0 ((uint32_t)0x00400000)
• #define RCC_DCKCFGGR_SAI1BSRC_1 ((uint32_t)0x00800000)
• #define RCC_DCKCFGGR_TIMPRE ((uint32_t)0x01000000)
• #define RNG_CR_RNGEN ((uint32_t)0x00000004)
• #define RNG_CR_IE ((uint32_t)0x00000008)
• #define RNG_SR_DRDY ((uint32_t)0x00000001)
• #define RNG_SR_CECS ((uint32_t)0x00000002)
```

- #define **RNG_SR_SECS** ((uint32_t)0x00000004)
- #define **RNG_SR_CEIS** ((uint32_t)0x00000020)
- #define **RNG_SR_SEIS** ((uint32_t)0x00000040)
- #define **RTC_TR_PM** ((uint32_t)0x00400000)
- #define **RTC_TR_HT** ((uint32_t)0x00300000)
- #define **RTC_TR_HT_0** ((uint32_t)0x00100000)
- #define **RTC_TR_HT_1** ((uint32_t)0x00200000)
- #define **RTC_TR_HU** ((uint32_t)0x000F0000)
- #define **RTC_TR_HU_0** ((uint32_t)0x00010000)
- #define **RTC_TR_HU_1** ((uint32_t)0x00020000)
- #define **RTC_TR_HU_2** ((uint32_t)0x00040000)
- #define **RTC_TR_HU_3** ((uint32_t)0x00080000)
- #define **RTC_TR_MNT** ((uint32_t)0x00007000)
- #define **RTC_TR_MNT_0** ((uint32_t)0x00001000)
- #define **RTC_TR_MNT_1** ((uint32_t)0x00002000)
- #define **RTC_TR_MNT_2** ((uint32_t)0x00004000)
- #define **RTC_TR_MNU** ((uint32_t)0x00000F00)
- #define **RTC_TR_MNU_0** ((uint32_t)0x00000100)
- #define **RTC_TR_MNU_1** ((uint32_t)0x00000200)
- #define **RTC_TR_MNU_2** ((uint32_t)0x00000400)
- #define **RTC_TR_MNU_3** ((uint32_t)0x00000800)
- #define **RTC_TR_ST** ((uint32_t)0x00000070)
- #define **RTC_TR_ST_0** ((uint32_t)0x00000010)
- #define **RTC_TR_ST_1** ((uint32_t)0x00000020)
- #define **RTC_TR_ST_2** ((uint32_t)0x00000040)
- #define **RTC_TR_SU** ((uint32_t)0x0000000F)
- #define **RTC_TR_SU_0** ((uint32_t)0x00000001)
- #define **RTC_TR_SU_1** ((uint32_t)0x00000002)
- #define **RTC_TR_SU_2** ((uint32_t)0x00000004)
- #define **RTC_TR_SU_3** ((uint32_t)0x00000008)
- #define **RTC_DR_YT** ((uint32_t)0x00F00000)
- #define **RTC_DR_YT_0** ((uint32_t)0x00100000)
- #define **RTC_DR_YT_1** ((uint32_t)0x00200000)
- #define **RTC_DR_YT_2** ((uint32_t)0x00400000)
- #define **RTC_DR_YT_3** ((uint32_t)0x00800000)
- #define **RTC_DR_YU** ((uint32_t)0x000F0000)
- #define **RTC_DR_YU_0** ((uint32_t)0x00010000)
- #define **RTC_DR_YU_1** ((uint32_t)0x00020000)
- #define **RTC_DR_YU_2** ((uint32_t)0x00040000)
- #define **RTC_DR_YU_3** ((uint32_t)0x00080000)
- #define **RTC_DR_WDU** ((uint32_t)0x0000E000)
- #define **RTC_DR_WDU_0** ((uint32_t)0x00002000)
- #define **RTC_DR_WDU_1** ((uint32_t)0x00004000)
- #define **RTC_DR_WDU_2** ((uint32_t)0x00008000)
- #define **RTC_DR_MT** ((uint32_t)0x00001000)
- #define **RTC_DR_MU** ((uint32_t)0x00000F00)
- #define **RTC_DR_MU_0** ((uint32_t)0x00000100)
- #define **RTC_DR_MU_1** ((uint32_t)0x00000200)
- #define **RTC_DR_MU_2** ((uint32_t)0x00000400)
- #define **RTC_DR_MU_3** ((uint32_t)0x00000800)
- #define **RTC_DR_DT** ((uint32_t)0x00000030)
- #define **RTC_DR_DT_0** ((uint32_t)0x00000010)
- #define **RTC_DR_DT_1** ((uint32_t)0x00000020)
- #define **RTC_DR_DU** ((uint32_t)0x0000000F)
- #define **RTC_DR_DU_0** ((uint32_t)0x00000001)

- #define **RTC_DR_DU_1** ((uint32_t)0x00000002)
- #define **RTC_DR_DU_2** ((uint32_t)0x00000004)
- #define **RTC_DR_DU_3** ((uint32_t)0x00000008)
- #define **RTC_CR_COE** ((uint32_t)0x00800000)
- #define **RTC_CR_OSEL** ((uint32_t)0x00600000)
- #define **RTC_CR_OSEL_0** ((uint32_t)0x00200000)
- #define **RTC_CR_OSEL_1** ((uint32_t)0x00400000)
- #define **RTC_CR_POL** ((uint32_t)0x00100000)
- #define **RTC_CR_COSEL** ((uint32_t)0x00080000)
- #define **RTC_CR_BCK** ((uint32_t)0x00040000)
- #define **RTC_CR_SUB1H** ((uint32_t)0x00020000)
- #define **RTC_CR_ADD1H** ((uint32_t)0x00010000)
- #define **RTC_CR_TSIE** ((uint32_t)0x00008000)
- #define **RTC_CR_WUTIE** ((uint32_t)0x00004000)
- #define **RTC_CR_ALRBIE** ((uint32_t)0x00002000)
- #define **RTC_CR_ALRAIE** ((uint32_t)0x00001000)
- #define **RTC_CR_TSE** ((uint32_t)0x00000800)
- #define **RTC_CR_WUTE** ((uint32_t)0x00000400)
- #define **RTC_CR_ALRBE** ((uint32_t)0x00000200)
- #define **RTC_CR_ALRAE** ((uint32_t)0x00000100)
- #define **RTC_CR_DCE** ((uint32_t)0x00000080)
- #define **RTC_CR_FMT** ((uint32_t)0x00000040)
- #define **RTC_CR_BYPSHAD** ((uint32_t)0x00000020)
- #define **RTC_CR_REFCKON** ((uint32_t)0x00000010)
- #define **RTC_CR_TSEDGE** ((uint32_t)0x00000008)
- #define **RTC_CR_WUCKSEL** ((uint32_t)0x00000007)
- #define **RTC_CR_WUCKSEL_0** ((uint32_t)0x00000001)
- #define **RTC_CR_WUCKSEL_1** ((uint32_t)0x00000002)
- #define **RTC_CR_WUCKSEL_2** ((uint32_t)0x00000004)
- #define **RTC_ISR_RECALPF** ((uint32_t)0x00010000)
- #define **RTC_ISR_TAMP1F** ((uint32_t)0x00002000)
- #define **RTC_ISR_TAMP2F** ((uint32_t)0x00004000)
- #define **RTC_ISR_TSOVF** ((uint32_t)0x00001000)
- #define **RTC_ISR_TSF** ((uint32_t)0x00000800)
- #define **RTC_ISR_WUTF** ((uint32_t)0x00000400)
- #define **RTC_ISR_ALRBF** ((uint32_t)0x00000200)
- #define **RTC_ISR_ALRAF** ((uint32_t)0x00000100)
- #define **RTC_ISR_INIT** ((uint32_t)0x00000080)
- #define **RTC_ISR_INITF** ((uint32_t)0x00000040)
- #define **RTC_ISR_RSF** ((uint32_t)0x00000020)
- #define **RTC_ISR_INITS** ((uint32_t)0x00000010)
- #define **RTC_ISR_SHPF** ((uint32_t)0x00000008)
- #define **RTC_ISR_WUTWF** ((uint32_t)0x00000004)
- #define **RTC_ISR_ALRBWF** ((uint32_t)0x00000002)
- #define **RTC_ISR_ALRAWF** ((uint32_t)0x00000001)
- #define **RTC_PRER_PREDIV_A** ((uint32_t)0x007F0000)
- #define **RTC_PRER_PREDIV_S** ((uint32_t)0x00001FFF)
- #define **RTC_WUTR_WUT** ((uint32_t)0x0000FFFF)
- #define **RTC_CALIBR_DCS** ((uint32_t)0x00000080)
- #define **RTC_CALIBR_DC** ((uint32_t)0x0000001F)
- #define **RTC_ALRMAR_MSK4** ((uint32_t)0x80000000)
- #define **RTC_ALRMAR_WDSEL** ((uint32_t)0x40000000)
- #define **RTC_ALRMAR_DT** ((uint32_t)0x30000000)
- #define **RTC_ALRMAR_DT_0** ((uint32_t)0x10000000)
- #define **RTC_ALRMAR_DT_1** ((uint32_t)0x20000000)

- #define **RTC_ALRMAR_DU** ((uint32_t)0x0F000000)
- #define **RTC_ALRMAR_DU_0** ((uint32_t)0x01000000)
- #define **RTC_ALRMAR_DU_1** ((uint32_t)0x02000000)
- #define **RTC_ALRMAR_DU_2** ((uint32_t)0x04000000)
- #define **RTC_ALRMAR_DU_3** ((uint32_t)0x08000000)
- #define **RTC_ALRMAR_MSK3** ((uint32_t)0x00800000)
- #define **RTC_ALRMAR_PM** ((uint32_t)0x00400000)
- #define **RTC_ALRMAR_HT** ((uint32_t)0x00300000)
- #define **RTC_ALRMAR_HT_0** ((uint32_t)0x00100000)
- #define **RTC_ALRMAR_HT_1** ((uint32_t)0x00200000)
- #define **RTC_ALRMAR_HU** ((uint32_t)0x000F0000)
- #define **RTC_ALRMAR_HU_0** ((uint32_t)0x00010000)
- #define **RTC_ALRMAR_HU_1** ((uint32_t)0x00020000)
- #define **RTC_ALRMAR_HU_2** ((uint32_t)0x00040000)
- #define **RTC_ALRMAR_HU_3** ((uint32_t)0x00080000)
- #define **RTC_ALRMAR_MSK2** ((uint32_t)0x00008000)
- #define **RTC_ALRMAR_MNT** ((uint32_t)0x00007000)
- #define **RTC_ALRMAR_MNT_0** ((uint32_t)0x00001000)
- #define **RTC_ALRMAR_MNT_1** ((uint32_t)0x00002000)
- #define **RTC_ALRMAR_MNT_2** ((uint32_t)0x00004000)
- #define **RTC_ALRMAR_MNU** ((uint32_t)0x0000F00)
- #define **RTC_ALRMAR_MNU_0** ((uint32_t)0x00000100)
- #define **RTC_ALRMAR_MNU_1** ((uint32_t)0x00000200)
- #define **RTC_ALRMAR_MNU_2** ((uint32_t)0x00000400)
- #define **RTC_ALRMAR_MNU_3** ((uint32_t)0x00000800)
- #define **RTC_ALRMAR_MSK1** ((uint32_t)0x00000080)
- #define **RTC_ALRMAR_ST** ((uint32_t)0x00000070)
- #define **RTC_ALRMAR_ST_0** ((uint32_t)0x00000010)
- #define **RTC_ALRMAR_ST_1** ((uint32_t)0x00000020)
- #define **RTC_ALRMAR_ST_2** ((uint32_t)0x00000040)
- #define **RTC_ALRMAR_SU** ((uint32_t)0x0000000F)
- #define **RTC_ALRMAR_SU_0** ((uint32_t)0x00000001)
- #define **RTC_ALRMAR_SU_1** ((uint32_t)0x00000002)
- #define **RTC_ALRMAR_SU_2** ((uint32_t)0x00000004)
- #define **RTC_ALRMAR_SU_3** ((uint32_t)0x00000008)
- #define **RTC_ALRMBR_MSK4** ((uint32_t)0x80000000)
- #define **RTC_ALRMBR_WDSEL** ((uint32_t)0x40000000)
- #define **RTC_ALRMBR_DT** ((uint32_t)0x30000000)
- #define **RTC_ALRMBR_DT_0** ((uint32_t)0x10000000)
- #define **RTC_ALRMBR_DT_1** ((uint32_t)0x20000000)
- #define **RTC_ALRMBR_DU** ((uint32_t)0x0F000000)
- #define **RTC_ALRMBR_DU_0** ((uint32_t)0x01000000)
- #define **RTC_ALRMBR_DU_1** ((uint32_t)0x02000000)
- #define **RTC_ALRMBR_DU_2** ((uint32_t)0x04000000)
- #define **RTC_ALRMBR_DU_3** ((uint32_t)0x08000000)
- #define **RTC_ALRMBR_MSK3** ((uint32_t)0x00800000)
- #define **RTC_ALRMBR_PM** ((uint32_t)0x00400000)
- #define **RTC_ALRMBR_HT** ((uint32_t)0x00300000)
- #define **RTC_ALRMBR_HT_0** ((uint32_t)0x00100000)
- #define **RTC_ALRMBR_HT_1** ((uint32_t)0x00200000)
- #define **RTC_ALRMBR_HU** ((uint32_t)0x000F0000)
- #define **RTC_ALRMBR_HU_0** ((uint32_t)0x00010000)
- #define **RTC_ALRMBR_HU_1** ((uint32_t)0x00020000)
- #define **RTC_ALRMBR_HU_2** ((uint32_t)0x00040000)
- #define **RTC_ALRMBR_HU_3** ((uint32_t)0x00080000)

- #define **RTC_ALRMBR_MSK2** ((uint32_t)0x000008000)
- #define **RTC_ALRMBR_MNT** ((uint32_t)0x000007000)
- #define **RTC_ALRMBR_MNT_0** ((uint32_t)0x000001000)
- #define **RTC_ALRMBR_MNT_1** ((uint32_t)0x000002000)
- #define **RTC_ALRMBR_MNT_2** ((uint32_t)0x000004000)
- #define **RTC_ALRMBR_MNU** ((uint32_t)0x00000F00)
- #define **RTC_ALRMBR_MNU_0** ((uint32_t)0x00000100)
- #define **RTC_ALRMBR_MNU_1** ((uint32_t)0x00000200)
- #define **RTC_ALRMBR_MNU_2** ((uint32_t)0x00000400)
- #define **RTC_ALRMBR_MNU_3** ((uint32_t)0x00000800)
- #define **RTC_ALRMBR_MSK1** ((uint32_t)0x00000080)
- #define **RTC_ALRMBR_ST** ((uint32_t)0x00000070)
- #define **RTC_ALRMBR_ST_0** ((uint32_t)0x00000010)
- #define **RTC_ALRMBR_ST_1** ((uint32_t)0x00000020)
- #define **RTC_ALRMBR_ST_2** ((uint32_t)0x00000040)
- #define **RTC_ALRMBR_SU** ((uint32_t)0x0000000F)
- #define **RTC_ALRMBR_SU_0** ((uint32_t)0x00000001)
- #define **RTC_ALRMBR_SU_1** ((uint32_t)0x00000002)
- #define **RTC_ALRMBR_SU_2** ((uint32_t)0x00000004)
- #define **RTC_ALRMBR_SU_3** ((uint32_t)0x00000008)
- #define **RTC_WPR_KEY** ((uint32_t)0x000000FF)
- #define **RTC_SSR_SS** ((uint32_t)0x0000FFFF)
- #define **RTC_SHIFTR_SUBFS** ((uint32_t)0x00007FFF)
- #define **RTC_SHIFTR_ADD1S** ((uint32_t)0x80000000)
- #define **RTC_TSTR_PM** ((uint32_t)0x00400000)
- #define **RTC_TSTR_HT** ((uint32_t)0x00300000)
- #define **RTC_TSTR_HT_0** ((uint32_t)0x00100000)
- #define **RTC_TSTR_HT_1** ((uint32_t)0x00200000)
- #define **RTC_TSTR_HU** ((uint32_t)0x000F0000)
- #define **RTC_TSTR_HU_0** ((uint32_t)0x00010000)
- #define **RTC_TSTR_HU_1** ((uint32_t)0x00020000)
- #define **RTC_TSTR_HU_2** ((uint32_t)0x00040000)
- #define **RTC_TSTR_HU_3** ((uint32_t)0x00080000)
- #define **RTC_TSTR_MNT** ((uint32_t)0x00007000)
- #define **RTC_TSTR_MNT_0** ((uint32_t)0x00001000)
- #define **RTC_TSTR_MNT_1** ((uint32_t)0x00002000)
- #define **RTC_TSTR_MNT_2** ((uint32_t)0x00004000)
- #define **RTC_TSTR_MNU** ((uint32_t)0x00000F00)
- #define **RTC_TSTR_MNU_0** ((uint32_t)0x00000100)
- #define **RTC_TSTR_MNU_1** ((uint32_t)0x00000200)
- #define **RTC_TSTR_MNU_2** ((uint32_t)0x00000400)
- #define **RTC_TSTR_MNU_3** ((uint32_t)0x00000800)
- #define **RTC_TSTR_ST** ((uint32_t)0x00000070)
- #define **RTC_TSTR_ST_0** ((uint32_t)0x00000010)
- #define **RTC_TSTR_ST_1** ((uint32_t)0x00000020)
- #define **RTC_TSTR_ST_2** ((uint32_t)0x00000040)
- #define **RTC_TSTR_SU** ((uint32_t)0x0000000F)
- #define **RTC_TSTR_SU_0** ((uint32_t)0x00000001)
- #define **RTC_TSTR_SU_1** ((uint32_t)0x00000002)
- #define **RTC_TSTR_SU_2** ((uint32_t)0x00000004)
- #define **RTC_TSTR_SU_3** ((uint32_t)0x00000008)
- #define **RTC_TSDR_WDU** ((uint32_t)0x0000E000)
- #define **RTC_TSDR_WDU_0** ((uint32_t)0x00002000)
- #define **RTC_TSDR_WDU_1** ((uint32_t)0x00004000)
- #define **RTC_TSDR_WDU_2** ((uint32_t)0x00008000)

```
• #define RTC_TSDF_MT ((uint32_t)0x00001000)
• #define RTC_TSDF_MU ((uint32_t)0x00000F00)
• #define RTC_TSDF_MU_0 ((uint32_t)0x00000100)
• #define RTC_TSDF_MU_1 ((uint32_t)0x00000200)
• #define RTC_TSDF_MU_2 ((uint32_t)0x00000400)
• #define RTC_TSDF_MU_3 ((uint32_t)0x00000800)
• #define RTC_TSDF_DT ((uint32_t)0x00000030)
• #define RTC_TSDF_DT_0 ((uint32_t)0x00000010)
• #define RTC_TSDF_DT_1 ((uint32_t)0x00000020)
• #define RTC_TSDF_DU ((uint32_t)0x0000000F)
• #define RTC_TSDF_DU_0 ((uint32_t)0x00000001)
• #define RTC_TSDF_DU_1 ((uint32_t)0x00000002)
• #define RTC_TSDF_DU_2 ((uint32_t)0x00000004)
• #define RTC_TSDF_DU_3 ((uint32_t)0x00000008)
• #define RTC_TSSR_SS ((uint32_t)0x0000FFFF)
• #define RTC_CALR_CALP ((uint32_t)0x00008000)
• #define RTC_CALR_CALW8 ((uint32_t)0x00004000)
• #define RTC_CALR_CALW16 ((uint32_t)0x00002000)
• #define RTC_CALR_CALM ((uint32_t)0x000001FF)
• #define RTC_CALR_CALM_0 ((uint32_t)0x00000001)
• #define RTC_CALR_CALM_1 ((uint32_t)0x00000002)
• #define RTC_CALR_CALM_2 ((uint32_t)0x00000004)
• #define RTC_CALR_CALM_3 ((uint32_t)0x00000008)
• #define RTC_CALR_CALM_4 ((uint32_t)0x00000010)
• #define RTC_CALR_CALM_5 ((uint32_t)0x00000020)
• #define RTC_CALR_CALM_6 ((uint32_t)0x00000040)
• #define RTC_CALR_CALM_7 ((uint32_t)0x00000080)
• #define RTC_CALR_CALM_8 ((uint32_t)0x00000100)
• #define RTC_TAFCR_ALARMOUTTYPE ((uint32_t)0x00040000)
• #define RTC_TAFCR_TSINSEL ((uint32_t)0x00020000)
• #define RTC_TAFCR_TAMPINSEL ((uint32_t)0x00010000)
• #define RTC_TAFCR_TAMPPUDIS ((uint32_t)0x00008000)
• #define RTC_TAFCR_TAMPPRCH ((uint32_t)0x00006000)
• #define RTC_TAFCR_TAMPPRCH_0 ((uint32_t)0x00002000)
• #define RTC_TAFCR_TAMPPRCH_1 ((uint32_t)0x00004000)
• #define RTC_TAFCR_TAMPFLT ((uint32_t)0x00001800)
• #define RTC_TAFCR_TAMPFLT_0 ((uint32_t)0x00000800)
• #define RTC_TAFCR_TAMPFLT_1 ((uint32_t)0x00001000)
• #define RTC_TAFCR_TAMPFREQ ((uint32_t)0x00000700)
• #define RTC_TAFCR_TAMPFREQ_0 ((uint32_t)0x00000100)
• #define RTC_TAFCR_TAMPFREQ_1 ((uint32_t)0x00000200)
• #define RTC_TAFCR_TAMPFREQ_2 ((uint32_t)0x00000400)
• #define RTC_TAFCR_TAMPTS ((uint32_t)0x00000080)
• #define RTC_TAFCR_TAMP2TRG ((uint32_t)0x00000010)
• #define RTC_TAFCR_TAMP2E ((uint32_t)0x00000008)
• #define RTC_TAFCR_TAMPIE ((uint32_t)0x00000004)
• #define RTC_TAFCR_TAMP1TRG ((uint32_t)0x00000002)
• #define RTC_TAFCR_TAMP1E ((uint32_t)0x00000001)
• #define RTC_ALRMASSR_MASKSS ((uint32_t)0x0F000000)
• #define RTC_ALRMASSR_MASKSS_0 ((uint32_t)0x01000000)
• #define RTC_ALRMASSR_MASKSS_1 ((uint32_t)0x02000000)
• #define RTC_ALRMASSR_MASKSS_2 ((uint32_t)0x04000000)
• #define RTC_ALRMASSR_MASKSS_3 ((uint32_t)0x08000000)
• #define RTC_ALRMASSR_SS ((uint32_t)0x000007FFF)
• #define RTC_ALRMBSSR_MASKSS ((uint32_t)0x0F000000)
```

```
• #define RTC_ALRMBSSR_MASKSS_0 ((uint32_t)0x01000000)
• #define RTC_ALRMBSSR_MASKSS_1 ((uint32_t)0x02000000)
• #define RTC_ALRMBSSR_MASKSS_2 ((uint32_t)0x04000000)
• #define RTC_ALRMBSSR_MASKSS_3 ((uint32_t)0x08000000)
• #define RTC_ALRMBSSR_SS ((uint32_t)0x00007FFF)
• #define RTC_BKP0R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP1R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP2R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP3R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP4R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP5R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP6R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP7R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP8R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP9R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP10R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP11R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP12R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP13R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP14R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP15R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP16R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP17R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP18R ((uint32_t)0xFFFFFFFF)
• #define RTC_BKP19R ((uint32_t)0xFFFFFFFF)
• #define SAI_GCR_SYNCIN ((uint32_t)0x00000003)
• #define SAI_GCR_SYNCIN_0 ((uint32_t)0x00000001)
• #define SAI_GCR_SYNCIN_1 ((uint32_t)0x00000002)
• #define SAI_GCR_SYNCOUT ((uint32_t)0x00000030)
• #define SAI_GCR_SYNCOUT_0 ((uint32_t)0x00000010)
• #define SAI_GCR_SYNCOUT_1 ((uint32_t)0x00000020)
• #define SAI_xCR1_MODE ((uint32_t)0x00000003)
• #define SAI_xCR1_MODE_0 ((uint32_t)0x00000001)
• #define SAI_xCR1_MODE_1 ((uint32_t)0x00000002)
• #define SAI_xCR1_PRTCFG ((uint32_t)0x0000000C)
• #define SAI_xCR1_PRTCFG_0 ((uint32_t)0x00000004)
• #define SAI_xCR1_PRTCFG_1 ((uint32_t)0x00000008)
• #define SAI_xCR1_DS ((uint32_t)0x000000E0)
• #define SAI_xCR1_DS_0 ((uint32_t)0x00000020)
• #define SAI_xCR1_DS_1 ((uint32_t)0x00000040)
• #define SAI_xCR1_DS_2 ((uint32_t)0x00000080)
• #define SAI_xCR1_LSBFIRST ((uint32_t)0x00000100)
• #define SAI_xCR1_CKSTR ((uint32_t)0x00000200)
• #define SAI_xCR1_SYNCEN ((uint32_t)0x00000C00)
• #define SAI_xCR1_SYNCEN_0 ((uint32_t)0x00000400)
• #define SAI_xCR1_SYNCEN_1 ((uint32_t)0x00000800)
• #define SAI_xCR1_MONO ((uint32_t)0x00001000)
• #define SAI_xCR1_OUTDRIV ((uint32_t)0x00002000)
• #define SAI_xCR1_SAIEN ((uint32_t)0x00010000)
• #define SAI_xCR1_DMAEN ((uint32_t)0x00020000)
• #define SAI_xCR1_NODIV ((uint32_t)0x00080000)
• #define SAI_xCR1_MCKDIV ((uint32_t)0x00780000)
• #define SAI_xCR1_MCKDIV_0 ((uint32_t)0x00080000)
• #define SAI_xCR1_MCKDIV_1 ((uint32_t)0x00100000)
• #define SAI_xCR1_MCKDIV_2 ((uint32_t)0x00200000)
```

```
• #define SAI_xCR1_MCKDIV_3 ((uint32_t)0x00400000)
• #define SAI_xCR2_FTH ((uint32_t)0x00000003)
• #define SAI_xCR2_FTH_0 ((uint32_t)0x00000001)
• #define SAI_xCR2_FTH_1 ((uint32_t)0x00000002)
• #define SAI_xCR2_FFLUSH ((uint32_t)0x00000008)
• #define SAI_xCR2_TRIS ((uint32_t)0x00000010)
• #define SAI_xCR2_MUTE ((uint32_t)0x00000020)
• #define SAI_xCR2_MUTEVAL ((uint32_t)0x00000040)
• #define SAI_xCR2_MUTECNT ((uint32_t)0x00001F80)
• #define SAI_xCR2_MUTECNT_0 ((uint32_t)0x00000080)
• #define SAI_xCR2_MUTECNT_1 ((uint32_t)0x00000100)
• #define SAI_xCR2_MUTECNT_2 ((uint32_t)0x00000200)
• #define SAI_xCR2_MUTECNT_3 ((uint32_t)0x00000400)
• #define SAI_xCR2_MUTECNT_4 ((uint32_t)0x00000800)
• #define SAI_xCR2_MUTECNT_5 ((uint32_t)0x00001000)
• #define SAI_xCR2_CPL ((uint32_t)0x00002000)
• #define SAI_xCR2_COMP ((uint32_t)0x0000C000)
• #define SAI_xCR2_COMP_0 ((uint32_t)0x00004000)
• #define SAI_xCR2_COMP_1 ((uint32_t)0x00008000)
• #define SAI_xFRCR_FRL ((uint32_t)0x000000FF)
• #define SAI_xFRCR_FRL_0 ((uint32_t)0x00000001)
• #define SAI_xFRCR_FRL_1 ((uint32_t)0x00000002)
• #define SAI_xFRCR_FRL_2 ((uint32_t)0x00000004)
• #define SAI_xFRCR_FRL_3 ((uint32_t)0x00000008)
• #define SAI_xFRCR_FRL_4 ((uint32_t)0x00000010)
• #define SAI_xFRCR_FRL_5 ((uint32_t)0x00000020)
• #define SAI_xFRCR_FRL_6 ((uint32_t)0x00000040)
• #define SAI_xFRCR_FRL_7 ((uint32_t)0x00000080)
• #define SAI_xFRCR_FSALL ((uint32_t)0x00007F00)
• #define SAI_xFRCR_FSALL_0 ((uint32_t)0x00000100)
• #define SAI_xFRCR_FSALL_1 ((uint32_t)0x00000200)
• #define SAI_xFRCR_FSALL_2 ((uint32_t)0x00000400)
• #define SAI_xFRCR_FSALL_3 ((uint32_t)0x00000800)
• #define SAI_xFRCR_FSALL_4 ((uint32_t)0x00001000)
• #define SAI_xFRCR_FSALL_5 ((uint32_t)0x00002000)
• #define SAI_xFRCR_FSALL_6 ((uint32_t)0x00004000)
• #define SAI_xFRCR_FSDEF ((uint32_t)0x00010000)
• #define SAI_xFRCR_FSPOL ((uint32_t)0x00020000)
• #define SAI_xFRCR_FSOFF ((uint32_t)0x00040000)
• #define SAI_xFRCR_FSPO SAI_xFRCR_FSPOL
• #define SAI_xSLOTR_FBOFF ((uint32_t)0x0000001F)
• #define SAI_xSLOTR_FBOFF_0 ((uint32_t)0x00000001)
• #define SAI_xSLOTR_FBOFF_1 ((uint32_t)0x00000002)
• #define SAI_xSLOTR_FBOFF_2 ((uint32_t)0x00000004)
• #define SAI_xSLOTR_FBOFF_3 ((uint32_t)0x00000008)
• #define SAI_xSLOTR_FBOFF_4 ((uint32_t)0x00000010)
• #define SAI_xSLOTR_SLOTsz ((uint32_t)0x000000C0)
• #define SAI_xSLOTR_SLOTsz_0 ((uint32_t)0x00000040)
• #define SAI_xSLOTR_SLOTsz_1 ((uint32_t)0x00000080)
• #define SAI_xSLOTR_NBSLOT ((uint32_t)0x00000F00)
• #define SAI_xSLOTR_NBSLOT_0 ((uint32_t)0x00000100)
• #define SAI_xSLOTR_NBSLOT_1 ((uint32_t)0x00000200)
• #define SAI_xSLOTR_NBSLOT_2 ((uint32_t)0x00000400)
• #define SAI_xSLOTR_NBSLOT_3 ((uint32_t)0x00000800)
• #define SAI_xSLOTR_SLOTEN ((uint32_t)0xFFFF0000)
```

- #define SAI_xIMR_OVRUDRIE ((uint32_t)0x00000001)
- #define SAI_xIMR_MUTEDETIE ((uint32_t)0x00000002)
- #define SAI_xIMR_WCKCFGIE ((uint32_t)0x00000004)
- #define SAI_xIMR_FREQIE ((uint32_t)0x00000008)
- #define SAI_xIMR_CNRDYIE ((uint32_t)0x00000010)
- #define SAI_xIMR_AFSDETIE ((uint32_t)0x00000020)
- #define SAI_xIMR_LFSDETIE ((uint32_t)0x00000040)
- #define SAI_xSR_OVRUDR ((uint32_t)0x00000001)
- #define SAI_xSR_MUTEDET ((uint32_t)0x00000002)
- #define SAI_xSR_WCKCFG ((uint32_t)0x00000004)
- #define SAI_xSR_FREQ ((uint32_t)0x00000008)
- #define SAI_xSR_CNRDY ((uint32_t)0x00000010)
- #define SAI_xSR_AFSDET ((uint32_t)0x00000020)
- #define SAI_xSR_LFSDET ((uint32_t)0x00000040)
- #define SAI_xSR_FLVL ((uint32_t)0x00070000)
- #define SAI_xSR_FLVL_0 ((uint32_t)0x00010000)
- #define SAI_xSR_FLVL_1 ((uint32_t)0x00020000)
- #define SAI_xSR_FLVL_2 ((uint32_t)0x00030000)
- #define SAI_xCLRFR_COVRUDR ((uint32_t)0x00000001)
- #define SAI_xCLRFR_CMUTEDET ((uint32_t)0x00000002)
- #define SAI_xCLRFR_CWCKCFG ((uint32_t)0x00000004)
- #define SAI_xCLRFR_CFREQ ((uint32_t)0x00000008)
- #define SAI_xCLRFR_CCNRDY ((uint32_t)0x00000010)
- #define SAI_xCLRFR_CAFSDET ((uint32_t)0x00000020)
- #define SAI_xCLRFR_CLFSDET ((uint32_t)0x00000040)
- #define SAI_xDR_DATA ((uint32_t)0xFFFFFFFF)
- #define SDIO_POWER_PWRCTRL ((uint8_t)0x03)
- #define SDIO_POWER_PWRCTRL_0 ((uint8_t)0x01)
- #define SDIO_CLKCR_CLKDIV ((uint16_t)0x00FF)
- #define SDIO_CLKCR_CLKEN ((uint16_t)0x0100)
- #define SDIO_CLKCR_PWRSAV ((uint16_t)0x0200)
- #define SDIO_CLKCR_BYPASS ((uint16_t)0x0400)
- #define SDIO_CLKCR_WIDBUS ((uint16_t)0x1800)
- #define SDIO_CLKCR_WIDBUS_0 ((uint16_t)0x0800)
- #define SDIO_CLKCR_WIDBUS_1 ((uint16_t)0x1000)
- #define SDIO_CLKCR_NEGEDGE ((uint16_t)0x2000)
- #define SDIO_CLKCR_HWFC_EN ((uint16_t)0x4000)
- #define SDIO_ARG_CMDARG ((uint32_t)0xFFFFFFFF)
- #define SDIO_CMD_CMDINDEX ((uint16_t)0x003F)
- #define SDIO_CMD_WAITRESP ((uint16_t)0x00C0)
- #define SDIO_CMD_WAITRESP_0 ((uint16_t)0x0040)
- #define SDIO_CMD_WAITRESP_1 ((uint16_t)0x0080)
- #define SDIO_CMD_WAITINT ((uint16_t)0x0100)
- #define SDIO_CMD_WAITPEND ((uint16_t)0x0200)
- #define SDIO_CMD_CPSMEN ((uint16_t)0x0400)
- #define SDIO_CMD_SDIOSUSPEND ((uint16_t)0x0800)
- #define SDIO_CMD_ENCMDCOMPL ((uint16_t)0x1000)
- #define SDIO_CMD_NIEN ((uint16_t)0x2000)
- #define SDIO_CMD_CEATACMD ((uint16_t)0x4000)
- #define SDIO_RESPCMD_RESPCMD ((uint8_t)0x3F)
- #define SDIO_RESP0_CARDSTATUS0 ((uint32_t)0xFFFFFFFF)
- #define SDIO_RESP1_CARDSTATUS1 ((uint32_t)0xFFFFFFFF)
- #define SDIO_RESP2_CARDSTATUS2 ((uint32_t)0xFFFFFFFF)
- #define SDIO_RESP3_CARDSTATUS3 ((uint32_t)0xFFFFFFFF)

- #define SDIO_RESP4_CARDSTATUS4 ((uint32_t)0xFFFFFFFF)
- #define SDIO_DTIMER_DATATIME ((uint32_t)0xFFFFFFFF)
- #define SDIO_DLEN_DATALENGTH ((uint32_t)0x01FFFFFF)
- #define SDIO_DCTRL_DTEN ((uint16_t)0x0001)
- #define SDIO_DCTRL_DTDIR ((uint16_t)0x0002)
- #define SDIO_DCTRL_DTMODE ((uint16_t)0x0004)
- #define SDIO_DCTRL_DMAEN ((uint16_t)0x0008)
- #define SDIO_DCTRL_DBLOCKSIZE ((uint16_t)0x00F0)
- #define SDIO_DCTRL_DBLOCKSIZE_0 ((uint16_t)0x0010)
- #define SDIO_DCTRL_DBLOCKSIZE_1 ((uint16_t)0x0020)
- #define SDIO_DCTRL_DBLOCKSIZE_2 ((uint16_t)0x0040)
- #define SDIO_DCTRL_DBLOCKSIZE_3 ((uint16_t)0x0080)
- #define SDIO_DCTRL_RWSTART ((uint16_t)0x0100)
- #define SDIO_DCTRL_RWSTOP ((uint16_t)0x0200)
- #define SDIO_DCTRL_RWMOD ((uint16_t)0x0400)
- #define SDIO_DCTRL_SDIOEN ((uint16_t)0x0800)
- #define SDIO_DCOUNT_DATACOUNT ((uint32_t)0x01FFFFFF)
- #define SDIO_STA_CCRCFAIL ((uint32_t)0x00000001)
- #define SDIO_STA_DCRCFAIL ((uint32_t)0x00000002)
- #define SDIO_STA_CTIMEOUT ((uint32_t)0x00000004)
- #define SDIO_STA_DTIMEOUT ((uint32_t)0x00000008)
- #define SDIO_STA_TXUNDERR ((uint32_t)0x00000010)
- #define SDIO_STA_RXOVERR ((uint32_t)0x00000020)
- #define SDIO_STA_CMDREND ((uint32_t)0x00000040)
- #define SDIO_STA_CMDSENT ((uint32_t)0x00000080)
- #define SDIO_STA_DATAEND ((uint32_t)0x00000100)
- #define SDIO_STA_STBITERR ((uint32_t)0x00000200)
- #define SDIO_STA_DBCKEND ((uint32_t)0x00000400)
- #define SDIO_STA_CMDACT ((uint32_t)0x00000800)
- #define SDIO_STA_TXACT ((uint32_t)0x00001000)
- #define SDIO_STA_RXACT ((uint32_t)0x00002000)
- #define SDIO_STA_TXFIFOHE ((uint32_t)0x00004000)
- #define SDIO_STA_RXFIFOHF ((uint32_t)0x00008000)
- #define SDIO_STA_TXFIFOF ((uint32_t)0x00010000)
- #define SDIO_STA_RXFIFOF ((uint32_t)0x00020000)
- #define SDIO_STA_TXFIFOE ((uint32_t)0x00040000)
- #define SDIO_STA_RXFIFOE ((uint32_t)0x00080000)
- #define SDIO_STA_TXDAVL ((uint32_t)0x00100000)
- #define SDIO_STA_RXDAVL ((uint32_t)0x00200000)
- #define SDIO_STA_SDIOIT ((uint32_t)0x00400000)
- #define SDIO_STA_CEATAEND ((uint32_t)0x00800000)
- #define SDIO_ICR_CCRCFAILC ((uint32_t)0x00000001)
- #define SDIO_ICR_DCRCFAILC ((uint32_t)0x00000002)
- #define SDIO_ICR_CTIMEOUTC ((uint32_t)0x00000004)
- #define SDIO_ICR_DTIMEOUTC ((uint32_t)0x00000008)
- #define SDIO_ICR_TXUNDERRC ((uint32_t)0x00000010)
- #define SDIO_ICR_RXOVERRRC ((uint32_t)0x00000020)
- #define SDIO_ICR_CMDRENDC ((uint32_t)0x00000040)
- #define SDIO_ICR_CMDSENTC ((uint32_t)0x00000080)
- #define SDIO_ICR_DATAENDC ((uint32_t)0x00000100)
- #define SDIO_ICR_STBITERRC ((uint32_t)0x00000200)
- #define SDIO_ICR_DBCKENDC ((uint32_t)0x00000400)
- #define SDIO_ICR_SDIOITC ((uint32_t)0x00400000)
- #define SDIO_ICR_CEATAENDC ((uint32_t)0x00800000)
- #define SDIO_MASK_CCRCFAILIE ((uint32_t)0x00000001)

- #define SDIO_MASK_DCRCFAILIE ((uint32_t)0x00000002)
- #define SDIO_MASK_CTIMEOUTIE ((uint32_t)0x00000004)
- #define SDIO_MASK_DTIMEOUTIE ((uint32_t)0x00000008)
- #define SDIO_MASK_TXUNDERRIE ((uint32_t)0x00000010)
- #define SDIO_MASK_RXOVERRIE ((uint32_t)0x00000020)
- #define SDIO_MASK_CMDRENDIE ((uint32_t)0x00000040)
- #define SDIO_MASK_CMDSENTIE ((uint32_t)0x00000080)
- #define SDIO_MASK_DATAENDIE ((uint32_t)0x00000100)
- #define SDIO_MASK_STBITERRIE ((uint32_t)0x00000200)
- #define SDIO_MASK_DBCKENDIE ((uint32_t)0x00000400)
- #define SDIO_MASK_CMDACTIE ((uint32_t)0x00000800)
- #define SDIO_MASK_TXACTIE ((uint32_t)0x00001000)
- #define SDIO_MASK_RXACTIE ((uint32_t)0x00002000)
- #define SDIO_MASK_TXFIFOHEIE ((uint32_t)0x00004000)
- #define SDIO_MASK_RXFIFOHFIE ((uint32_t)0x00008000)
- #define SDIO_MASK_RXFIFOFIE ((uint32_t)0x00010000)
- #define SDIO_MASK_RXFIFOFIE ((uint32_t)0x00020000)
- #define SDIO_MASK_RXFIFOEIE ((uint32_t)0x00040000)
- #define SDIO_MASK_RXFIFOEIE ((uint32_t)0x00080000)
- #define SDIO_MASK_TXDAVLIE ((uint32_t)0x00100000)
- #define SDIO_MASK_RXDAVLIE ((uint32_t)0x00200000)
- #define SDIO_MASK_SDIOITIE ((uint32_t)0x00400000)
- #define SDIO_MASK_CEATAENDIE ((uint32_t)0x00800000)
- #define SDIO_FIFOCNT_FIFOCOUNT ((uint32_t)0x00FFFFFF)
- #define SDIO_FIFO_FIFODATA ((uint32_t)0xFFFFFFFF)
- #define SPI_CR1_CPHA ((uint16_t)0x0001)
- #define SPI_CR1_CPOL ((uint16_t)0x0002)
- #define SPI_CR1_MSTR ((uint16_t)0x0004)
- #define SPI_CR1_BR ((uint16_t)0x0038)
- #define SPI_CR1_BR_0 ((uint16_t)0x0008)
- #define SPI_CR1_BR_1 ((uint16_t)0x0010)
- #define SPI_CR1_BR_2 ((uint16_t)0x0020)
- #define SPI_CR1_SPE ((uint16_t)0x0040)
- #define SPI_CR1_LSBFIRST ((uint16_t)0x0080)
- #define SPI_CR1_SSI ((uint16_t)0x0100)
- #define SPI_CR1_SSM ((uint16_t)0x0200)
- #define SPI_CR1_RXONLY ((uint16_t)0x0400)
- #define SPI_CR1_DFF ((uint16_t)0x0800)
- #define SPI_CR1_CRCNEXT ((uint16_t)0x1000)
- #define SPI_CR1_CRCEN ((uint16_t)0x2000)
- #define SPI_CR1_BIDIOE ((uint16_t)0x4000)
- #define SPI_CR1_BIDIODE ((uint16_t)0x8000)
- #define SPI_CR2_RXDMAEN ((uint8_t)0x01)
- #define SPI_CR2_TXDMAEN ((uint8_t)0x02)
- #define SPI_CR2_SSOE ((uint8_t)0x04)
- #define SPI_CR2_ERRIE ((uint8_t)0x20)
- #define SPI_CR2_RXNEIE ((uint8_t)0x40)
- #define SPI_CR2_TXEIE ((uint8_t)0x80)
- #define SPI_SR_RXNE ((uint8_t)0x01)
- #define SPI_SR_TXE ((uint8_t)0x02)
- #define SPI_SR_CHSIDE ((uint8_t)0x04)
- #define SPI_SR_UDR ((uint8_t)0x08)
- #define SPI_SR_CRCERR ((uint8_t)0x10)
- #define SPI_SR_MODF ((uint8_t)0x20)
- #define SPI_SR_OVR ((uint8_t)0x40)

- #define SPI_SR_BSY ((uint8_t)0x80)
- #define SPI_DR_DR ((uint16_t)0xFFFF)
- #define SPI_CRCPR_CRCPOL ((uint16_t)0xFFFF)
- #define SPI_RXCRCR_RXCRC ((uint16_t)0xFFFF)
- #define SPI_TXCRCR_TXCRC ((uint16_t)0xFFFF)
- #define SPI_I2SCFGR_CHLEN ((uint16_t)0x0001)
- #define SPI_I2SCFGR_DATLEN ((uint16_t)0x0006)
- #define SPI_I2SCFGR_DATLEN_0 ((uint16_t)0x0002)
- #define SPI_I2SCFGR_DATLEN_1 ((uint16_t)0x0004)
- #define SPI_I2SCFGR_CKPOL ((uint16_t)0x0008)
- #define SPI_I2SCFGR_I2SSTD ((uint16_t)0x0030)
- #define SPI_I2SCFGR_I2SSTD_0 ((uint16_t)0x0010)
- #define SPI_I2SCFGR_I2SSTD_1 ((uint16_t)0x0020)
- #define SPI_I2SCFGR_PCMSYNC ((uint16_t)0x0080)
- #define SPI_I2SCFGR_I2SCFG ((uint16_t)0x0300)
- #define SPI_I2SCFGR_I2SCFG_0 ((uint16_t)0x0100)
- #define SPI_I2SCFGR_I2SCFG_1 ((uint16_t)0x0200)
- #define SPI_I2SCFGR_I2SE ((uint16_t)0x0400)
- #define SPI_I2SCFGR_I2SMOD ((uint16_t)0x0800)
- #define SPI_I2SPR_I2SDIV ((uint16_t)0x00FF)
- #define SPI_I2SPR_ODD ((uint16_t)0x0100)
- #define SPI_I2SPR_MCKOE ((uint16_t)0x0200)
- #define SYSCFG_MEMRMP_MEM_MODE ((uint32_t)0x00000007)
- #define SYSCFG_MEMRMP_MEM_MODE_0 ((uint32_t)0x00000001)
- #define SYSCFG_MEMRMP_MEM_MODE_1 ((uint32_t)0x00000002)
- #define SYSCFG_MEMRMP_MEM_MODE_2 ((uint32_t)0x00000004)
- #define SYSCFG_MEMRMP_FB_MODE ((uint32_t)0x00000100)
- #define SYSCFG_MEMRMP_SWP_FMC ((uint32_t)0x00000C00)
- #define SYSCFG_MEMRMP_SWP_FMC_0 ((uint32_t)0x00000400)
- #define SYSCFG_MEMRMP_SWP_FMC_1 ((uint32_t)0x00000800)
- #define SYSCFG_PMC_ADCxDC2 ((uint32_t)0x00070000)
- #define SYSCFG_PMC_ADC1DC2 ((uint32_t)0x00010000)
- #define SYSCFG_PMC_ADC2DC2 ((uint32_t)0x00020000)
- #define SYSCFG_PMC_ADC3DC2 ((uint32_t)0x00040000)
- #define SYSCFG_PMC_MII_RMII_SEL ((uint32_t)0x00800000)
- #define SYSCFG_PMC_MII_RMII SYSCFG_PMC_MII_RMII_SEL
- #define SYSCFG_EXTICR1_EXTI0 ((uint16_t)0x000F)
- #define SYSCFG_EXTICR1_EXTI1 ((uint16_t)0x00F0)
- #define SYSCFG_EXTICR1_EXTI2 ((uint16_t)0x0F00)
- #define SYSCFG_EXTICR1_EXTI3 ((uint16_t)0xF000)
- #define SYSCFG_EXTICR1_EXTI0_PA ((uint16_t)0x0000)

EXTI0 configuration

- #define SYSCFG_EXTICR1_EXTI0_PB ((uint16_t)0x0001)
- #define SYSCFG_EXTICR1_EXTI0_PC ((uint16_t)0x0002)
- #define SYSCFG_EXTICR1_EXTI0_PD ((uint16_t)0x0003)
- #define SYSCFG_EXTICR1_EXTI0_PE ((uint16_t)0x0004)
- #define SYSCFG_EXTICR1_EXTI0_PF ((uint16_t)0x0005)
- #define SYSCFG_EXTICR1_EXTI0_PG ((uint16_t)0x0006)
- #define SYSCFG_EXTICR1_EXTI0_PH ((uint16_t)0x0007)
- #define SYSCFG_EXTICR1_EXTI0_PI ((uint16_t)0x0008)
- #define SYSCFG_EXTICR1_EXTI0_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR1_EXTI0_PK ((uint16_t)0x000A)
- #define SYSCFG_EXTICR1_EXTI1_PA ((uint16_t)0x0000)

EXTI1 configuration

- #define SYSCFG_EXTICR1_EXTI1_PB ((uint16_t)0x0010)
- #define SYSCFG_EXTICR1_EXTI1_PC ((uint16_t)0x0020)
- #define SYSCFG_EXTICR1_EXTI1_PD ((uint16_t)0x0030)
- #define SYSCFG_EXTICR1_EXTI1_PE ((uint16_t)0x0040)
- #define SYSCFG_EXTICR1_EXTI1_PF ((uint16_t)0x0050)
- #define SYSCFG_EXTICR1_EXTI1_PG ((uint16_t)0x0060)
- #define SYSCFG_EXTICR1_EXTI1_PH ((uint16_t)0x0070)
- #define SYSCFG_EXTICR1_EXTI1_PI ((uint16_t)0x0080)
- #define SYSCFG_EXTICR1_EXTI1_PJ ((uint16_t)0x0090)
- #define SYSCFG_EXTICR1_EXTI1_PK ((uint16_t)0x00A0)
- #define SYSCFG_EXTICR1_EXTI2_PA ((uint16_t)0x0000)

EXTI2 configuration

- #define SYSCFG_EXTICR1_EXTI2_PB ((uint16_t)0x0100)
- #define SYSCFG_EXTICR1_EXTI2_PC ((uint16_t)0x0200)
- #define SYSCFG_EXTICR1_EXTI2_PD ((uint16_t)0x0300)
- #define SYSCFG_EXTICR1_EXTI2_PE ((uint16_t)0x0400)
- #define SYSCFG_EXTICR1_EXTI2_PF ((uint16_t)0x0500)
- #define SYSCFG_EXTICR1_EXTI2_PG ((uint16_t)0x0600)
- #define SYSCFG_EXTICR1_EXTI2_PH ((uint16_t)0x0700)
- #define SYSCFG_EXTICR1_EXTI2_PI ((uint16_t)0x0800)
- #define SYSCFG_EXTICR1_EXTI2_PJ ((uint16_t)0x0900)
- #define SYSCFG_EXTICR1_EXTI2_PK ((uint16_t)0xA000)
- #define SYSCFG_EXTICR1_EXTI3_PA ((uint16_t)0x0000)

EXTI3 configuration

- #define SYSCFG_EXTICR1_EXTI3_PB ((uint16_t)0x1000)
- #define SYSCFG_EXTICR1_EXTI3_PC ((uint16_t)0x2000)
- #define SYSCFG_EXTICR1_EXTI3_PD ((uint16_t)0x3000)
- #define SYSCFG_EXTICR1_EXTI3_PE ((uint16_t)0x4000)
- #define SYSCFG_EXTICR1_EXTI3_PF ((uint16_t)0x5000)
- #define SYSCFG_EXTICR1_EXTI3_PG ((uint16_t)0x6000)
- #define SYSCFG_EXTICR1_EXTI3_PH ((uint16_t)0x7000)
- #define SYSCFG_EXTICR1_EXTI3_PI ((uint16_t)0x8000)
- #define SYSCFG_EXTICR1_EXTI3_PJ ((uint16_t)0x9000)
- #define SYSCFG_EXTICR1_EXTI3_PK ((uint16_t)0xA000)
- #define SYSCFG_EXTICR2_EXTI4 ((uint16_t)0x000F)
- #define SYSCFG_EXTICR2_EXTI5 ((uint16_t)0x00F0)
- #define SYSCFG_EXTICR2_EXTI6 ((uint16_t)0x0F00)
- #define SYSCFG_EXTICR2_EXTI7 ((uint16_t)0xF000)
- #define SYSCFG_EXTICR2_EXTI4_PA ((uint16_t)0x0000)

EXTI4 configuration

- #define SYSCFG_EXTICR2_EXTI4_PB ((uint16_t)0x0001)
- #define SYSCFG_EXTICR2_EXTI4_PC ((uint16_t)0x0002)
- #define SYSCFG_EXTICR2_EXTI4_PD ((uint16_t)0x0003)
- #define SYSCFG_EXTICR2_EXTI4_PE ((uint16_t)0x0004)
- #define SYSCFG_EXTICR2_EXTI4_PF ((uint16_t)0x0005)
- #define SYSCFG_EXTICR2_EXTI4_PG ((uint16_t)0x0006)
- #define SYSCFG_EXTICR2_EXTI4_PH ((uint16_t)0x0007)
- #define SYSCFG_EXTICR2_EXTI4_PI ((uint16_t)0x0008)
- #define SYSCFG_EXTICR2_EXTI4_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR2_EXTI4_PK ((uint16_t)0x000A)

- #define SYSCFG_EXTICR2_EXTI5_PA ((uint16_t)0x0000)
EXTI5 configuration
 - #define SYSCFG_EXTICR2_EXTI5_PB ((uint16_t)0x0010)
 - #define SYSCFG_EXTICR2_EXTI5_PC ((uint16_t)0x0020)
 - #define SYSCFG_EXTICR2_EXTI5_PD ((uint16_t)0x0030)
 - #define SYSCFG_EXTICR2_EXTI5_PE ((uint16_t)0x0040)
 - #define SYSCFG_EXTICR2_EXTI5_PF ((uint16_t)0x0050)
 - #define SYSCFG_EXTICR2_EXTI5_PG ((uint16_t)0x0060)
 - #define SYSCFG_EXTICR2_EXTI5_PH ((uint16_t)0x0070)
 - #define SYSCFG_EXTICR2_EXTI5_PI ((uint16_t)0x0080)
 - #define SYSCFG_EXTICR2_EXTI5_PJ ((uint16_t)0x0090)
 - #define SYSCFG_EXTICR2_EXTI5_PK ((uint16_t)0x00A0)
 - #define SYSCFG_EXTICR2_EXTI6_PA ((uint16_t)0x0000)
- #define SYSCFG_EXTICR2_EXTI6_PB ((uint16_t)0x0100)
EXTI6 configuration
 - #define SYSCFG_EXTICR2_EXTI6_PC ((uint16_t)0x0200)
 - #define SYSCFG_EXTICR2_EXTI6_PD ((uint16_t)0x0300)
 - #define SYSCFG_EXTICR2_EXTI6_PE ((uint16_t)0x0400)
 - #define SYSCFG_EXTICR2_EXTI6_PF ((uint16_t)0x0500)
 - #define SYSCFG_EXTICR2_EXTI6_PG ((uint16_t)0x0600)
 - #define SYSCFG_EXTICR2_EXTI6_PH ((uint16_t)0x0700)
 - #define SYSCFG_EXTICR2_EXTI6_PI ((uint16_t)0x0800)
 - #define SYSCFG_EXTICR2_EXTI6_PJ ((uint16_t)0x0900)
 - #define SYSCFG_EXTICR2_EXTI6_PK ((uint16_t)0xA000)
 - #define SYSCFG_EXTICR2_EXTI7_PA ((uint16_t)0x0000)
- #define SYSCFG_EXTICR2_EXTI7_PB ((uint16_t)0x1000)
EXTI7 configuration
 - #define SYSCFG_EXTICR2_EXTI7_PC ((uint16_t)0x2000)
 - #define SYSCFG_EXTICR2_EXTI7_PD ((uint16_t)0x3000)
 - #define SYSCFG_EXTICR2_EXTI7_PE ((uint16_t)0x4000)
 - #define SYSCFG_EXTICR2_EXTI7_PF ((uint16_t)0x5000)
 - #define SYSCFG_EXTICR2_EXTI7_PG ((uint16_t)0x6000)
 - #define SYSCFG_EXTICR2_EXTI7_PH ((uint16_t)0x7000)
 - #define SYSCFG_EXTICR2_EXTI7_PI ((uint16_t)0x8000)
 - #define SYSCFG_EXTICR2_EXTI7_PJ ((uint16_t)0x9000)
 - #define SYSCFG_EXTICR2_EXTI7_PK ((uint16_t)0xA000)
 - #define SYSCFG_EXTICR3_EXTI8 ((uint16_t)0x000F)
 - #define SYSCFG_EXTICR3_EXTI9 ((uint16_t)0x00F0)
 - #define SYSCFG_EXTICR3_EXTI10 ((uint16_t)0x0F00)
 - #define SYSCFG_EXTICR3_EXTI11 ((uint16_t)0xF000)
 - #define SYSCFG_EXTICR3_EXTI8_PA ((uint16_t)0x0000)
- #define SYSCFG_EXTICR3_EXTI8_PB ((uint16_t)0x0001)
EXTI8 configuration
 - #define SYSCFG_EXTICR3_EXTI8_PC ((uint16_t)0x0002)
 - #define SYSCFG_EXTICR3_EXTI8_PD ((uint16_t)0x0003)
 - #define SYSCFG_EXTICR3_EXTI8_PE ((uint16_t)0x0004)
 - #define SYSCFG_EXTICR3_EXTI8_PF ((uint16_t)0x0005)
 - #define SYSCFG_EXTICR3_EXTI8_PG ((uint16_t)0x0006)
 - #define SYSCFG_EXTICR3_EXTI8_PH ((uint16_t)0x0007)
 - #define SYSCFG_EXTICR3_EXTI8_PI ((uint16_t)0x0008)

- #define SYSCFG_EXTICR3_EXTI8_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR3_EXTI9_PA ((uint16_t)0x0000)

EXTI9 configuration

- #define SYSCFG_EXTICR3_EXTI9_PB ((uint16_t)0x0010)
- #define SYSCFG_EXTICR3_EXTI9_PC ((uint16_t)0x0020)
- #define SYSCFG_EXTICR3_EXTI9_PD ((uint16_t)0x0030)
- #define SYSCFG_EXTICR3_EXTI9_PE ((uint16_t)0x0040)
- #define SYSCFG_EXTICR3_EXTI9_PF ((uint16_t)0x0050)
- #define SYSCFG_EXTICR3_EXTI9_PG ((uint16_t)0x0060)
- #define SYSCFG_EXTICR3_EXTI9_PH ((uint16_t)0x0070)
- #define SYSCFG_EXTICR3_EXTI9_PI ((uint16_t)0x0080)
- #define SYSCFG_EXTICR3_EXTI9_PJ ((uint16_t)0x0090)
- #define SYSCFG_EXTICR3_EXTI10_PA ((uint16_t)0x0000)

EXTI10 configuration

- #define SYSCFG_EXTICR3_EXTI10_PB ((uint16_t)0x0100)
- #define SYSCFG_EXTICR3_EXTI10_PC ((uint16_t)0x0200)
- #define SYSCFG_EXTICR3_EXTI10_PD ((uint16_t)0x0300)
- #define SYSCFG_EXTICR3_EXTI10_PE ((uint16_t)0x0400)
- #define SYSCFG_EXTICR3_EXTI10_PF ((uint16_t)0x0500)
- #define SYSCFG_EXTICR3_EXTI10_PG ((uint16_t)0x0600)
- #define SYSCFG_EXTICR3_EXTI10_PH ((uint16_t)0x0700)
- #define SYSCFG_EXTICR3_EXTI10_PI ((uint16_t)0x0800)
- #define SYSCFG_EXTICR3_EXTI10_PJ ((uint16_t)0x0900)
- #define SYSCFG_EXTICR3_EXTI11_PA ((uint16_t)0x0000)

EXTI11 configuration

- #define SYSCFG_EXTICR3_EXTI11_PB ((uint16_t)0x1000)
- #define SYSCFG_EXTICR3_EXTI11_PC ((uint16_t)0x2000)
- #define SYSCFG_EXTICR3_EXTI11_PD ((uint16_t)0x3000)
- #define SYSCFG_EXTICR3_EXTI11_PE ((uint16_t)0x4000)
- #define SYSCFG_EXTICR3_EXTI11_PF ((uint16_t)0x5000)
- #define SYSCFG_EXTICR3_EXTI11_PG ((uint16_t)0x6000)
- #define SYSCFG_EXTICR3_EXTI11_PH ((uint16_t)0x7000)
- #define SYSCFG_EXTICR3_EXTI11_PI ((uint16_t)0x8000)
- #define SYSCFG_EXTICR3_EXTI11_PJ ((uint16_t)0x9000)
- #define SYSCFG_EXTICR4_EXTI12 ((uint16_t)0x000F)
- #define SYSCFG_EXTICR4_EXTI13 ((uint16_t)0x00F0)
- #define SYSCFG_EXTICR4_EXTI14 ((uint16_t)0x0F00)
- #define SYSCFG_EXTICR4_EXTI15 ((uint16_t)0xF000)
- #define SYSCFG_EXTICR4_EXTI12_PA ((uint16_t)0x0000)

EXTI12 configuration

- #define SYSCFG_EXTICR4_EXTI12_PB ((uint16_t)0x0001)
- #define SYSCFG_EXTICR4_EXTI12_PC ((uint16_t)0x0002)
- #define SYSCFG_EXTICR4_EXTI12_PD ((uint16_t)0x0003)
- #define SYSCFG_EXTICR4_EXTI12_PE ((uint16_t)0x0004)
- #define SYSCFG_EXTICR4_EXTI12_PF ((uint16_t)0x0005)
- #define SYSCFG_EXTICR4_EXTI12_PG ((uint16_t)0x0006)
- #define SYSCFG_EXTICR4_EXTI12_PH ((uint16_t)0x0007)
- #define SYSCFG_EXTICR4_EXTI12_PI ((uint16_t)0x0008)
- #define SYSCFG_EXTICR4_EXTI12_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR4_EXTI13_PA ((uint16_t)0x0000)

EXTI13 configuration

- #define SYSCFG_EXTICR4_EXTI13_PB ((uint16_t)0x0010)
- #define SYSCFG_EXTICR4_EXTI13_PC ((uint16_t)0x0020)
- #define SYSCFG_EXTICR4_EXTI13_PD ((uint16_t)0x0030)
- #define SYSCFG_EXTICR4_EXTI13_PE ((uint16_t)0x0040)
- #define SYSCFG_EXTICR4_EXTI13_PF ((uint16_t)0x0050)
- #define SYSCFG_EXTICR4_EXTI13_PG ((uint16_t)0x0060)
- #define SYSCFG_EXTICR4_EXTI13_PH ((uint16_t)0x0070)
- #define SYSCFG_EXTICR4_EXTI13_PI ((uint16_t)0x0008)
- #define SYSCFG_EXTICR4_EXTI13_PJ ((uint16_t)0x0009)
- #define SYSCFG_EXTICR4_EXTI14_PA ((uint16_t)0x0000)

EXTI14 configuration

- #define SYSCFG_EXTICR4_EXTI14_PB ((uint16_t)0x0100)
- #define SYSCFG_EXTICR4_EXTI14_PC ((uint16_t)0x0200)
- #define SYSCFG_EXTICR4_EXTI14_PD ((uint16_t)0x0300)
- #define SYSCFG_EXTICR4_EXTI14_PE ((uint16_t)0x0400)
- #define SYSCFG_EXTICR4_EXTI14_PF ((uint16_t)0x0500)
- #define SYSCFG_EXTICR4_EXTI14_PG ((uint16_t)0x0600)
- #define SYSCFG_EXTICR4_EXTI14_PH ((uint16_t)0x0700)
- #define SYSCFG_EXTICR4_EXTI14_PI ((uint16_t)0x0800)
- #define SYSCFG_EXTICR4_EXTI14_PJ ((uint16_t)0x0900)
- #define SYSCFG_EXTICR4_EXTI15_PA ((uint16_t)0x0000)

EXTI15 configuration

- #define SYSCFG_EXTICR4_EXTI15_PB ((uint16_t)0x1000)
- #define SYSCFG_EXTICR4_EXTI15_PC ((uint16_t)0x2000)
- #define SYSCFG_EXTICR4_EXTI15_PD ((uint16_t)0x3000)
- #define SYSCFG_EXTICR4_EXTI15_PE ((uint16_t)0x4000)
- #define SYSCFG_EXTICR4_EXTI15_PF ((uint16_t)0x5000)
- #define SYSCFG_EXTICR4_EXTI15_PG ((uint16_t)0x6000)
- #define SYSCFG_EXTICR4_EXTI15_PH ((uint16_t)0x7000)
- #define SYSCFG_EXTICR4_EXTI15_PI ((uint16_t)0x8000)
- #define SYSCFG_EXTICR4_EXTI15_PJ ((uint16_t)0x9000)
- #define SYSCFG_CMPCR_CMP_PD ((uint32_t)0x00000001)
- #define SYSCFG_CMPCR_READY ((uint32_t)0x00000100)
- #define TIM_CR1_CEN ((uint16_t)0x0001)
- #define TIM_CR1_UDIS ((uint16_t)0x0002)
- #define TIM_CR1_URS ((uint16_t)0x0004)
- #define TIM_CR1_OPM ((uint16_t)0x0008)
- #define TIM_CR1_DIR ((uint16_t)0x0010)
- #define TIM_CR1_CMS ((uint16_t)0x0060)
- #define TIM_CR1_CMS_0 ((uint16_t)0x0020)
- #define TIM_CR1_CMS_1 ((uint16_t)0x0040)
- #define TIM_CR1_ARPE ((uint16_t)0x0080)
- #define TIM_CR1_CKD ((uint16_t)0x0300)
- #define TIM_CR1_CKD_0 ((uint16_t)0x0100)
- #define TIM_CR1_CKD_1 ((uint16_t)0x0200)
- #define TIM_CR2_CCPC ((uint16_t)0x0001)
- #define TIM_CR2_CCUS ((uint16_t)0x0004)
- #define TIM_CR2_CCDS ((uint16_t)0x0008)
- #define TIM_CR2_MMS ((uint16_t)0x0070)
- #define TIM_CR2_MMS_0 ((uint16_t)0x0010)
- #define TIM_CR2_MMS_1 ((uint16_t)0x0020)

- #define **TIM_CR2_MMS_2** ((uint16_t)0x0040)
- #define **TIM_CR2_TI1S** ((uint16_t)0x0080)
- #define **TIM_CR2_OIS1** ((uint16_t)0x0100)
- #define **TIM_CR2_OIS1N** ((uint16_t)0x0200)
- #define **TIM_CR2_OIS2** ((uint16_t)0x0400)
- #define **TIM_CR2_OIS2N** ((uint16_t)0x0800)
- #define **TIM_CR2_OIS3** ((uint16_t)0x1000)
- #define **TIM_CR2_OIS3N** ((uint16_t)0x2000)
- #define **TIM_CR2_OIS4** ((uint16_t)0x4000)
- #define **TIM_SMCR_SMS** ((uint16_t)0x0007)
- #define **TIM_SMCR_SMS_0** ((uint16_t)0x0001)
- #define **TIM_SMCR_SMS_1** ((uint16_t)0x0002)
- #define **TIM_SMCR_SMS_2** ((uint16_t)0x0004)
- #define **TIM_SMCR_TS** ((uint16_t)0x0070)
- #define **TIM_SMCR_TS_0** ((uint16_t)0x0010)
- #define **TIM_SMCR_TS_1** ((uint16_t)0x0020)
- #define **TIM_SMCR_TS_2** ((uint16_t)0x0040)
- #define **TIM_SMCR_MSM** ((uint16_t)0x0080)
- #define **TIM_SMCR_ETF** ((uint16_t)0x0F00)
- #define **TIM_SMCR_ETF_0** ((uint16_t)0x0100)
- #define **TIM_SMCR_ETF_1** ((uint16_t)0x0200)
- #define **TIM_SMCR_ETF_2** ((uint16_t)0x0400)
- #define **TIM_SMCR_ETF_3** ((uint16_t)0x0800)
- #define **TIM_SMCR_ETPS** ((uint16_t)0x3000)
- #define **TIM_SMCR_ETPS_0** ((uint16_t)0x1000)
- #define **TIM_SMCR_ETPS_1** ((uint16_t)0x2000)
- #define **TIM_SMCR_ECE** ((uint16_t)0x4000)
- #define **TIM_SMCR_ETP** ((uint16_t)0x8000)
- #define **TIM_DIER_UIE** ((uint16_t)0x0001)
- #define **TIM_DIER_CC1IE** ((uint16_t)0x0002)
- #define **TIM_DIER_CC2IE** ((uint16_t)0x0004)
- #define **TIM_DIER_CC3IE** ((uint16_t)0x0008)
- #define **TIM_DIER_CC4IE** ((uint16_t)0x0010)
- #define **TIM_DIER_COMIE** ((uint16_t)0x0020)
- #define **TIM_DIER_TIE** ((uint16_t)0x0040)
- #define **TIM_DIER_BIE** ((uint16_t)0x0080)
- #define **TIM_DIER_UDE** ((uint16_t)0x0100)
- #define **TIM_DIER_CC1DE** ((uint16_t)0x0200)
- #define **TIM_DIER_CC2DE** ((uint16_t)0x0400)
- #define **TIM_DIER_CC3DE** ((uint16_t)0x0800)
- #define **TIM_DIER_CC4DE** ((uint16_t)0x1000)
- #define **TIM_DIER_COMDE** ((uint16_t)0x2000)
- #define **TIM_DIER_TDE** ((uint16_t)0x4000)
- #define **TIM_SR UIF** ((uint16_t)0x0001)
- #define **TIM_SR_CC1IF** ((uint16_t)0x0002)
- #define **TIM_SR_CC2IF** ((uint16_t)0x0004)
- #define **TIM_SR_CC3IF** ((uint16_t)0x0008)
- #define **TIM_SR_CC4IF** ((uint16_t)0x0010)
- #define **TIM_SR_COMIF** ((uint16_t)0x0020)
- #define **TIM_SR_TIF** ((uint16_t)0x0040)
- #define **TIM_SR_BIF** ((uint16_t)0x0080)
- #define **TIM_SR_CC1OF** ((uint16_t)0x0200)
- #define **TIM_SR_CC2OF** ((uint16_t)0x0400)
- #define **TIM_SR_CC3OF** ((uint16_t)0x0800)
- #define **TIM_SR_CC4OF** ((uint16_t)0x1000)

- #define TIM_EGR_UG ((uint8_t)0x01)
- #define TIM_EGR_CC1G ((uint8_t)0x02)
- #define TIM_EGR_CC2G ((uint8_t)0x04)
- #define TIM_EGR_CC3G ((uint8_t)0x08)
- #define TIM_EGR_CC4G ((uint8_t)0x10)
- #define TIM_EGR_COMG ((uint8_t)0x20)
- #define TIM_EGR_TG ((uint8_t)0x40)
- #define TIM_EGR_BG ((uint8_t)0x80)
- #define TIM_CCMR1_CC1S ((uint16_t)0x0003)
- #define TIM_CCMR1_CC1S_0 ((uint16_t)0x0001)
- #define TIM_CCMR1_CC1S_1 ((uint16_t)0x0002)
- #define TIM_CCMR1_OC1FE ((uint16_t)0x0004)
- #define TIM_CCMR1_OC1PE ((uint16_t)0x0008)
- #define TIM_CCMR1_OC1M ((uint16_t)0x0070)
- #define TIM_CCMR1_OC1M_0 ((uint16_t)0x0010)
- #define TIM_CCMR1_OC1M_1 ((uint16_t)0x0020)
- #define TIM_CCMR1_OC1M_2 ((uint16_t)0x0040)
- #define TIM_CCMR1_OC1CE ((uint16_t)0x0080)
- #define TIM_CCMR1_CC2S ((uint16_t)0x0300)
- #define TIM_CCMR1_CC2S_0 ((uint16_t)0x0100)
- #define TIM_CCMR1_CC2S_1 ((uint16_t)0x0200)
- #define TIM_CCMR1_OC2FE ((uint16_t)0x0400)
- #define TIM_CCMR1_OC2PE ((uint16_t)0x0800)
- #define TIM_CCMR1_OC2M ((uint16_t)0x7000)
- #define TIM_CCMR1_OC2M_0 ((uint16_t)0x1000)
- #define TIM_CCMR1_OC2M_1 ((uint16_t)0x2000)
- #define TIM_CCMR1_OC2M_2 ((uint16_t)0x4000)
- #define TIM_CCMR1_OC2CE ((uint16_t)0x8000)
- #define TIM_CCMR1_IC1PSC ((uint16_t)0x000C)
- #define TIM_CCMR1_IC1PSC_0 ((uint16_t)0x0004)
- #define TIM_CCMR1_IC1PSC_1 ((uint16_t)0x0008)
- #define TIM_CCMR1_IC1F ((uint16_t)0x00F0)
- #define TIM_CCMR1_IC1F_0 ((uint16_t)0x0010)
- #define TIM_CCMR1_IC1F_1 ((uint16_t)0x0020)
- #define TIM_CCMR1_IC1F_2 ((uint16_t)0x0040)
- #define TIM_CCMR1_IC1F_3 ((uint16_t)0x0080)
- #define TIM_CCMR1_IC2PSC ((uint16_t)0x0C00)
- #define TIM_CCMR1_IC2PSC_0 ((uint16_t)0x0400)
- #define TIM_CCMR1_IC2PSC_1 ((uint16_t)0x0800)
- #define TIM_CCMR1_IC2F ((uint16_t)0xF000)
- #define TIM_CCMR1_IC2F_0 ((uint16_t)0x1000)
- #define TIM_CCMR1_IC2F_1 ((uint16_t)0x2000)
- #define TIM_CCMR1_IC2F_2 ((uint16_t)0x4000)
- #define TIM_CCMR1_IC2F_3 ((uint16_t)0x8000)
- #define TIM_CCMR2_CC3S ((uint16_t)0x0003)
- #define TIM_CCMR2_CC3S_0 ((uint16_t)0x0001)
- #define TIM_CCMR2_CC3S_1 ((uint16_t)0x0002)
- #define TIM_CCMR2_OC3FE ((uint16_t)0x0004)
- #define TIM_CCMR2_OC3PE ((uint16_t)0x0008)
- #define TIM_CCMR2_OC3M ((uint16_t)0x0070)
- #define TIM_CCMR2_OC3M_0 ((uint16_t)0x0010)
- #define TIM_CCMR2_OC3M_1 ((uint16_t)0x0020)
- #define TIM_CCMR2_OC3M_2 ((uint16_t)0x0040)
- #define TIM_CCMR2_OC3CE ((uint16_t)0x0080)
- #define TIM_CCMR2_CC4S ((uint16_t)0x0300)

- #define TIM_CCMR2_CC4S_0 ((uint16_t)0x0100)
- #define TIM_CCMR2_CC4S_1 ((uint16_t)0x0200)
- #define TIM_CCMR2_OC4FE ((uint16_t)0x0400)
- #define TIM_CCMR2_OC4PE ((uint16_t)0x0800)
- #define TIM_CCMR2_OC4M ((uint16_t)0x7000)
- #define TIM_CCMR2_OC4M_0 ((uint16_t)0x1000)
- #define TIM_CCMR2_OC4M_1 ((uint16_t)0x2000)
- #define TIM_CCMR2_OC4M_2 ((uint16_t)0x4000)
- #define TIM_CCMR2_OC4CE ((uint16_t)0x8000)
- #define TIM_CCMR2_IC3PSC ((uint16_t)0x000C)
- #define TIM_CCMR2_IC3PSC_0 ((uint16_t)0x0004)
- #define TIM_CCMR2_IC3PSC_1 ((uint16_t)0x0008)
- #define TIM_CCMR2_IC3F ((uint16_t)0x00F0)
- #define TIM_CCMR2_IC3F_0 ((uint16_t)0x0010)
- #define TIM_CCMR2_IC3F_1 ((uint16_t)0x0020)
- #define TIM_CCMR2_IC3F_2 ((uint16_t)0x0040)
- #define TIM_CCMR2_IC3F_3 ((uint16_t)0x0080)
- #define TIM_CCMR2_IC4PSC ((uint16_t)0x0C00)
- #define TIM_CCMR2_IC4PSC_0 ((uint16_t)0x0400)
- #define TIM_CCMR2_IC4PSC_1 ((uint16_t)0x0800)
- #define TIM_CCMR2_IC4F ((uint16_t)0xF000)
- #define TIM_CCMR2_IC4F_0 ((uint16_t)0x1000)
- #define TIM_CCMR2_IC4F_1 ((uint16_t)0x2000)
- #define TIM_CCMR2_IC4F_2 ((uint16_t)0x4000)
- #define TIM_CCMR2_IC4F_3 ((uint16_t)0x8000)
- #define TIM_CCER_CC1E ((uint16_t)0x0001)
- #define TIM_CCER_CC1P ((uint16_t)0x0002)
- #define TIM_CCER_CC1NE ((uint16_t)0x0004)
- #define TIM_CCER_CC1NP ((uint16_t)0x0008)
- #define TIM_CCER_CC2E ((uint16_t)0x0010)
- #define TIM_CCER_CC2P ((uint16_t)0x0020)
- #define TIM_CCER_CC2NE ((uint16_t)0x0040)
- #define TIM_CCER_CC2NP ((uint16_t)0x0080)
- #define TIM_CCER_CC3E ((uint16_t)0x0100)
- #define TIM_CCER_CC3P ((uint16_t)0x0200)
- #define TIM_CCER_CC3NE ((uint16_t)0x0400)
- #define TIM_CCER_CC3NP ((uint16_t)0x0800)
- #define TIM_CCER_CC4E ((uint16_t)0x1000)
- #define TIM_CCER_CC4P ((uint16_t)0x2000)
- #define TIM_CCER_CC4NP ((uint16_t)0x8000)
- #define TIM_CNT_CNT ((uint16_t)0xFFFF)
- #define TIM_PSC_PSC ((uint16_t)0xFFFF)
- #define TIM_ARR_ARR ((uint16_t)0xFFFF)
- #define TIM_RCR REP ((uint8_t)0xFF)
- #define TIM_CCR1_CCR1 ((uint16_t)0xFFFF)
- #define TIM_CCR2_CCR2 ((uint16_t)0xFFFF)
- #define TIM_CCR3_CCR3 ((uint16_t)0xFFFF)
- #define TIM_CCR4_CCR4 ((uint16_t)0xFFFF)
- #define TIM_BDTR_DTG ((uint16_t)0x00FF)
- #define TIM_BDTR_DTG_0 ((uint16_t)0x0001)
- #define TIM_BDTR_DTG_1 ((uint16_t)0x0002)
- #define TIM_BDTR_DTG_2 ((uint16_t)0x0004)
- #define TIM_BDTR_DTG_3 ((uint16_t)0x0008)
- #define TIM_BDTR_DTG_4 ((uint16_t)0x0010)
- #define TIM_BDTR_DTG_5 ((uint16_t)0x0020)

- #define TIM_BDTR_DTG_6 ((uint16_t)0x0040)
- #define TIM_BDTR_DTG_7 ((uint16_t)0x0080)
- #define TIM_BDTR_LOCK ((uint16_t)0x0300)
- #define TIM_BDTR_LOCK_0 ((uint16_t)0x0100)
- #define TIM_BDTR_LOCK_1 ((uint16_t)0x0200)
- #define TIM_BDTR_OSSI ((uint16_t)0x0400)
- #define TIM_BDTR_OSSR ((uint16_t)0x0800)
- #define TIM_BDTR_BKE ((uint16_t)0x1000)
- #define TIM_BDTR_BKP ((uint16_t)0x2000)
- #define TIM_BDTR_AOE ((uint16_t)0x4000)
- #define TIM_BDTR_MOE ((uint16_t)0x8000)
- #define TIM_DCR_DBA ((uint16_t)0x001F)
- #define TIM_DCR_DBA_0 ((uint16_t)0x0001)
- #define TIM_DCR_DBA_1 ((uint16_t)0x0002)
- #define TIM_DCR_DBA_2 ((uint16_t)0x0004)
- #define TIM_DCR_DBA_3 ((uint16_t)0x0008)
- #define TIM_DCR_DBA_4 ((uint16_t)0x0010)
- #define TIM_DCR_DBL ((uint16_t)0x1F00)
- #define TIM_DCR_DBL_0 ((uint16_t)0x0100)
- #define TIM_DCR_DBL_1 ((uint16_t)0x0200)
- #define TIM_DCR_DBL_2 ((uint16_t)0x0400)
- #define TIM_DCR_DBL_3 ((uint16_t)0x0800)
- #define TIM_DCR_DBL_4 ((uint16_t)0x1000)
- #define TIM_DMAR_DMAB ((uint16_t)0xFFFF)
- #define TIM_OR_TI4_RMP ((uint16_t)0x00C0)
- #define TIM_OR_TI4_RMP_0 ((uint16_t)0x0040)
- #define TIM_OR_TI4_RMP_1 ((uint16_t)0x0080)
- #define TIM_OR_ITR1_RMP ((uint16_t)0x0C00)
- #define TIM_OR_ITR1_RMP_0 ((uint16_t)0x0400)
- #define TIM_OR_ITR1_RMP_1 ((uint16_t)0x0800)
- #define USART_SR_PE ((uint16_t)0x0001)
- #define USART_SR_FE ((uint16_t)0x0002)
- #define USART_SR_NE ((uint16_t)0x0004)
- #define USART_SR_ORE ((uint16_t)0x0008)
- #define USART_SR_IDLE ((uint16_t)0x0010)
- #define USART_SR_RXNE ((uint16_t)0x0020)
- #define USART_SR_TC ((uint16_t)0x0040)
- #define USART_SR_TXE ((uint16_t)0x0080)
- #define USART_SR_LBD ((uint16_t)0x0100)
- #define USART_SR_CTS ((uint16_t)0x0200)
- #define USART_DR_DR ((uint16_t)0x01FF)
- #define USART_BRR_DIV_Fraction ((uint16_t)0x000F)
- #define USART_BRR_DIV_Mantissa ((uint16_t)0xFFFF0)
- #define USART_CR1_SBK ((uint16_t)0x0001)
- #define USART_CR1_RWU ((uint16_t)0x0002)
- #define USART_CR1_RE ((uint16_t)0x0004)
- #define USART_CR1_TE ((uint16_t)0x0008)
- #define USART_CR1_IDLEIE ((uint16_t)0x0010)
- #define USART_CR1_RXNEIE ((uint16_t)0x0020)
- #define USART_CR1_TCIE ((uint16_t)0x0040)
- #define USART_CR1_TXEIE ((uint16_t)0x0080)
- #define USART_CR1_PEIE ((uint16_t)0x0100)
- #define USART_CR1_PS ((uint16_t)0x0200)
- #define USART_CR1_PCE ((uint16_t)0x0400)
- #define USART_CR1_WAKE ((uint16_t)0x0800)

- #define USART_CR1_M ((uint16_t)0x1000)
- #define USART_CR1_UE ((uint16_t)0x2000)
- #define USART_CR1_OVER8 ((uint16_t)0x8000)
- #define USART_CR2_ADD ((uint16_t)0x000F)
- #define USART_CR2_LBDL ((uint16_t)0x0020)
- #define USART_CR2_LBDIE ((uint16_t)0x0040)
- #define USART_CR2_LBCL ((uint16_t)0x0100)
- #define USART_CR2_CPHA ((uint16_t)0x0200)
- #define USART_CR2_CPOL ((uint16_t)0x0400)
- #define USART_CR2_CLKEN ((uint16_t)0x0800)
- #define USART_CR2_STOP ((uint16_t)0x3000)
- #define USART_CR2_STOP_0 ((uint16_t)0x1000)
- #define USART_CR2_STOP_1 ((uint16_t)0x2000)
- #define USART_CR2_LINE_N ((uint16_t)0x4000)
- #define USART_CR3_EIE ((uint16_t)0x0001)
- #define USART_CR3_IREN ((uint16_t)0x0002)
- #define USART_CR3_IRLP ((uint16_t)0x0004)
- #define USART_CR3_HDSEL ((uint16_t)0x0008)
- #define USART_CR3_NACK ((uint16_t)0x0010)
- #define USART_CR3_SCEN ((uint16_t)0x0020)
- #define USART_CR3_DMAR ((uint16_t)0x0040)
- #define USART_CR3_DMAT ((uint16_t)0x0080)
- #define USART_CR3_RTSE ((uint16_t)0x0100)
- #define USART_CR3_CTSE ((uint16_t)0x0200)
- #define USART_CR3_CTSIE ((uint16_t)0x0400)
- #define USART_CR3_ONEBIT ((uint16_t)0x0800)
- #define USART_GTPR_PSC ((uint16_t)0x00FF)
- #define USART_GTPR_PSC_0 ((uint16_t)0x0001)
- #define USART_GTPR_PSC_1 ((uint16_t)0x0002)
- #define USART_GTPR_PSC_2 ((uint16_t)0x0004)
- #define USART_GTPR_PSC_3 ((uint16_t)0x0008)
- #define USART_GTPR_PSC_4 ((uint16_t)0x0010)
- #define USART_GTPR_PSC_5 ((uint16_t)0x0020)
- #define USART_GTPR_PSC_6 ((uint16_t)0x0040)
- #define USART_GTPR_PSC_7 ((uint16_t)0x0080)
- #define USART_GTPR_GT ((uint16_t)0xFF00)
- #define WWDG_CR_T ((uint8_t)0x7F)
- #define WWDG_CR_T_0 ((uint8_t)0x01)
- #define WWDG_CR_T_1 ((uint8_t)0x02)
- #define WWDG_CR_T_2 ((uint8_t)0x04)
- #define WWDG_CR_T_3 ((uint8_t)0x08)
- #define WWDG_CR_T_4 ((uint8_t)0x10)
- #define WWDG_CR_T_5 ((uint8_t)0x20)
- #define WWDG_CR_T_6 ((uint8_t)0x40)
- #define WWDG_CR_T0 WWDG_CR_T_0
- #define WWDG_CR_T1 WWDG_CR_T_1
- #define WWDG_CR_T2 WWDG_CR_T_2
- #define WWDG_CR_T3 WWDG_CR_T_3
- #define WWDG_CR_T4 WWDG_CR_T_4
- #define WWDG_CR_T5 WWDG_CR_T_5
- #define WWDG_CR_T6 WWDG_CR_T_6
- #define WWDG_CR_WDGA ((uint8_t)0x80)
- #define WWDG_CFR_W ((uint16_t)0x007F)
- #define WWDG_CFR_W_0 ((uint16_t)0x0001)
- #define WWDG_CFR_W_1 ((uint16_t)0x0002)

```
• #define WWDG_CFR_W_2 ((uint16_t)0x0004)
• #define WWDG_CFR_W_3 ((uint16_t)0x0008)
• #define WWDG_CFR_W_4 ((uint16_t)0x0010)
• #define WWDG_CFR_W_5 ((uint16_t)0x0020)
• #define WWDG_CFR_W_6 ((uint16_t)0x0040)
• #define WWDG_CFR_W0 WWDG_CFR_W_0
• #define WWDG_CFR_W1 WWDG_CFR_W_1
• #define WWDG_CFR_W2 WWDG_CFR_W_2
• #define WWDG_CFR_W3 WWDG_CFR_W_3
• #define WWDG_CFR_W4 WWDG_CFR_W_4
• #define WWDG_CFR_W5 WWDG_CFR_W_5
• #define WWDG_CFR_W6 WWDG_CFR_W_6
• #define WWDG_CFR_WDGTB ((uint16_t)0x0180)
• #define WWDG_CFR_WDGTB_0 ((uint16_t)0x0080)
• #define WWDG_CFR_WDGTB_1 ((uint16_t)0x0100)
• #define WWDG_CFR_WDGTB0 WWDG_CFR_WDGTB_0
• #define WWDG_CFR_WDGTB1 WWDG_CFR_WDGTB_1
• #define WWDG_CFR_EWI ((uint16_t)0x0200)
• #define WWDG_SR_EWIF ((uint8_t)0x01)
• #define DBGMCU_IDCODE_DEV_ID ((uint32_t)0x00000FFF)
• #define DBGMCU_IDCODE_REV_ID ((uint32_t)0xFFFF0000)
• #define DBGMCU_CR_DBG_SLEEP ((uint32_t)0x00000001)
• #define DBGMCU_CR_DBG_STOP ((uint32_t)0x00000002)
• #define DBGMCU_CR_DBG_STANDBY ((uint32_t)0x00000004)
• #define DBGMCU_CR_TRACE_IOEN ((uint32_t)0x00000020)
• #define DBGMCU_CR_TRACE_MODE ((uint32_t)0x000000C0)
• #define DBGMCU_CR_TRACE_MODE_0 ((uint32_t)0x00000040)
• #define DBGMCU_CR_TRACE_MODE_1 ((uint32_t)0x00000080)
• #define DBGMCU_APB1_FZ_DBG_TIM2_STOP ((uint32_t)0x00000001)
• #define DBGMCU_APB1_FZ_DBG_TIM3_STOP ((uint32_t)0x00000002)
• #define DBGMCU_APB1_FZ_DBG_TIM4_STOP ((uint32_t)0x00000004)
• #define DBGMCU_APB1_FZ_DBG_TIM5_STOP ((uint32_t)0x00000008)
• #define DBGMCU_APB1_FZ_DBG_TIM6_STOP ((uint32_t)0x00000010)
• #define DBGMCU_APB1_FZ_DBG_TIM7_STOP ((uint32_t)0x00000020)
• #define DBGMCU_APB1_FZ_DBG_TIM12_STOP ((uint32_t)0x00000040)
• #define DBGMCU_APB1_FZ_DBG_TIM13_STOP ((uint32_t)0x00000080)
• #define DBGMCU_APB1_FZ_DBG_TIM14_STOP ((uint32_t)0x00000100)
• #define DBGMCU_APB1_FZ_DBG_RTC_STOP ((uint32_t)0x00000400)
• #define DBGMCU_APB1_FZ_DBG_WWDG_STOP ((uint32_t)0x00000800)
• #define DBGMCU_APB1_FZ_DBG_IWDG_STOP ((uint32_t)0x00001000)
• #define DBGMCU_APB1_FZ_DBG_I2C1_SMBUS_TIMEOUT ((uint32_t)0x00200000)
• #define DBGMCU_APB1_FZ_DBG_I2C2_SMBUS_TIMEOUT ((uint32_t)0x00400000)
• #define DBGMCU_APB1_FZ_DBG_I2C3_SMBUS_TIMEOUT ((uint32_t)0x00800000)
• #define DBGMCU_APB1_FZ_DBG_CAN1_STOP ((uint32_t)0x02000000)
• #define DBGMCU_APB1_FZ_DBG_CAN2_STOP ((uint32_t)0x04000000)
• #define DBGMCU_APB1_FZ_DBG_IWDEG_STOP DBGMCU_APB1_FZ_DBG_IWDG_STOP
• #define DBGMCU_APB1_FZ_DBG_TIM1_STOP ((uint32_t)0x00000001)
• #define DBGMCU_APB1_FZ_DBG_TIM8_STOP ((uint32_t)0x00000002)
• #define DBGMCU_APB1_FZ_DBG_TIM9_STOP ((uint32_t)0x00010000)
• #define DBGMCU_APB1_FZ_DBG_TIM10_STOP ((uint32_t)0x00020000)
• #define DBGMCU_APB1_FZ_DBG_TIM11_STOP ((uint32_t)0x00040000)
• #define ETH_MACCR_WD ((uint32_t)0x00800000) /* Watchdog disable */
• #define ETH_MACCR_JD ((uint32_t)0x00400000) /* Jabber disable */
• #define ETH_MACCR_IFG ((uint32_t)0x000E0000) /* Inter-frame gap */
```

- #define **ETH_MACCR_IFG_96Bit** ((uint32_t)0x00000000) /* Minimum IFG between frames during transmission is 96Bit */
- #define **ETH_MACCR_IFG_88Bit** ((uint32_t)0x00020000) /* Minimum IFG between frames during transmission is 88Bit */
- #define **ETH_MACCR_IFG_80Bit** ((uint32_t)0x00040000) /* Minimum IFG between frames during transmission is 80Bit */
- #define **ETH_MACCR_IFG_72Bit** ((uint32_t)0x00060000) /* Minimum IFG between frames during transmission is 72Bit */
- #define **ETH_MACCR_IFG_64Bit** ((uint32_t)0x00080000) /* Minimum IFG between frames during transmission is 64Bit */
- #define **ETH_MACCR_IFG_56Bit** ((uint32_t)0x000A0000) /* Minimum IFG between frames during transmission is 56Bit */
- #define **ETH_MACCR_IFG_48Bit** ((uint32_t)0x000C0000) /* Minimum IFG between frames during transmission is 48Bit */
- #define **ETH_MACCR_IFG_40Bit** ((uint32_t)0x000E0000) /* Minimum IFG between frames during transmission is 40Bit */
- #define **ETH_MACCR_CSD** ((uint32_t)0x00010000) /* Carrier sense disable (during transmission) */
- #define **ETH_MACCR_FES** ((uint32_t)0x00004000) /* Fast ethernet speed */
- #define **ETH_MACCR_ROD** ((uint32_t)0x00002000) /* Receive own disable */
- #define **ETH_MACCR_LM** ((uint32_t)0x00001000) /* loopback mode */
- #define **ETH_MACCR_DM** ((uint32_t)0x00000800) /* Duplex mode */
- #define **ETH_MACCR_IPCO** ((uint32_t)0x00000400) /* IP Checksum offload */
- #define **ETH_MACCR_RD** ((uint32_t)0x00000200) /* Retry disable */
- #define **ETH_MACCR_APCS** ((uint32_t)0x00000080) /* Automatic Pad/CRC stripping */
- #define **ETH_MACCR_BL**
- #define **ETH_MACCR_BL_10** ((uint32_t)0x00000000) /* k = min (n, 10) */
- #define **ETH_MACCR_BL_8** ((uint32_t)0x00000020) /* k = min (n, 8) */
- #define **ETH_MACCR_BL_4** ((uint32_t)0x00000040) /* k = min (n, 4) */
- #define **ETH_MACCR_BL_1** ((uint32_t)0x00000060) /* k = min (n, 1) */
- #define **ETH_MACCR_DC** ((uint32_t)0x00000010) /* Defferal check */
- #define **ETH_MACCR_TE** ((uint32_t)0x00000008) /* Transmitter enable */
- #define **ETH_MACCR_RE** ((uint32_t)0x00000004) /* Receiver enable */
- #define **ETH_MACFFR_RA** ((uint32_t)0x80000000) /* Receive all */
- #define **ETH_MACFFR_HPF** ((uint32_t)0x00000400) /* Hash or perfect filter */
- #define **ETH_MACFFR_SAF** ((uint32_t)0x00000200) /* Source address filter enable */
- #define **ETH_MACFFR_SAIF** ((uint32_t)0x00000100) /* SA inverse filtering */
- #define **ETH_MACFFR_PCF** ((uint32_t)0x000000C0) /* Pass control frames: 3 cases */
- #define **ETH_MACFFR_PCF_BlockAll** ((uint32_t)0x00000040) /* MAC filters all control frames from reaching the application */
- #define **ETH_MACFFR_PCF_ForwardAll** ((uint32_t)0x00000080) /* MAC forwards all control frames to application even if they fail the Address Filter */
- #define **ETH_MACFFR_PCF_ForwardPassedAddrFilter** ((uint32_t)0x000000C0) /* MAC forwards control frames that pass the Address Filter. */
- #define **ETH_MACFFR_BFD** ((uint32_t)0x00000020) /* Broadcast frame disable */
- #define **ETH_MACFFR_PAM** ((uint32_t)0x00000010) /* Pass all mutlicast */
- #define **ETH_MACFFR_DAIF** ((uint32_t)0x00000008) /* DA Inverse filtering */
- #define **ETH_MACFFR_HM** ((uint32_t)0x00000004) /* Hash multicast */
- #define **ETH_MACFFR_HU** ((uint32_t)0x00000002) /* Hash unicast */
- #define **ETH_MACFFR_PM** ((uint32_t)0x00000001) /* Promiscuous mode */
- #define **ETH_MACHTHR_HTH** ((uint32_t)0xFFFFFFFF) /* Hash table high */
- #define **ETH_MACHTLR_HTL** ((uint32_t)0xFFFFFFFF) /* Hash table low */
- #define **ETH_MACMIIAR_PA** ((uint32_t)0x0000F800) /* Physical layer address */
- #define **ETH_MACMIIAR_MR** ((uint32_t)0x000007C0) /* MII register in the selected PHY */
- #define **ETH_MACMIIAR_CR** ((uint32_t)0x0000001C) /* CR clock range: 6 cases */
- #define **ETH_MACMIIAR_CR_Div42** ((uint32_t)0x00000000) /* HCLK:60-100 MHz; MDC clock= HCLK/42 */

- #define **ETH_MACMIIAR_CR_Div62** ((uint32_t)0x00000004) /* HCLK:100-150 MHz; MDC clock= HCLK/62 */
 */
- #define **ETH_MACMIIAR_CR_Div16** ((uint32_t)0x00000008) /* HCLK:20-35 MHz; MDC clock= HCLK/16 */
- #define **ETH_MACMIIAR_CR_Div26** ((uint32_t)0x0000000C) /* HCLK:35-60 MHz; MDC clock= HCLK/26 */
- #define **ETH_MACMIIAR_CR_Div102** ((uint32_t)0x00000010) /* HCLK:150-168 MHz; MDC clock= HCLK/102 */
- #define **ETH_MACMIIAR_MW** ((uint32_t)0x00000002) /* MII write */
- #define **ETH_MACMIIAR_MB** ((uint32_t)0x00000001) /* MII busy */
- #define **ETH_MACMIIDR_MD** ((uint32_t)0x0000FFFF) /* MII data: read/write data from/to PHY */
- #define **ETH_MACFCR_PT** ((uint32_t)0xFFFF0000) /* Pause time */
- #define **ETH_MACFCR_ZQPD** ((uint32_t)0x00000080) /* Zero-quanta pause disable */
- #define **ETH_MACFCR_PLT** ((uint32_t)0x00000030) /* Pause low threshold: 4 cases */
- #define **ETH_MACFCR_PLT_Minus4** ((uint32_t)0x00000000) /* Pause time minus 4 slot times */
- #define **ETH_MACFCR_PLT_Minus28** ((uint32_t)0x00000010) /* Pause time minus 28 slot times */
- #define **ETH_MACFCR_PLT_Minus144** ((uint32_t)0x00000020) /* Pause time minus 144 slot times */
- #define **ETH_MACFCR_PLT_Minus256** ((uint32_t)0x00000030) /* Pause time minus 256 slot times */
- #define **ETH_MACFCR_UPFD** ((uint32_t)0x00000008) /* Unicast pause frame detect */
- #define **ETH_MACFCR_RFCE** ((uint32_t)0x00000004) /* Receive flow control enable */
- #define **ETH_MACFCR_TFCE** ((uint32_t)0x00000002) /* Transmit flow control enable */
- #define **ETH_MACFCR_FCBBPA** ((uint32_t)0x00000001) /* Flow control busy/backpressure activate */
- #define **ETH_MACVLANTR_VLANTC** ((uint32_t)0x00010000) /* 12-bit VLAN tag comparison */
- #define **ETH_MACVLANTR_VLANTI** ((uint32_t)0x0000FFFF) /* VLAN tag identifier (for receive frames) */
- #define **ETH_MACRWUFFR_D** ((uint32_t)0xFFFFFFFF) /* Wake-up frame filter register data */
- #define **ETH_MACPMTCSR_WFFRPR** ((uint32_t)0x80000000) /* Wake-Up Frame Filter Register Pointer Reset */
- #define **ETH_MACPMTCSR_GU** ((uint32_t)0x00000200) /* Global Unicast */
- #define **ETH_MACPMTCSR_WFR** ((uint32_t)0x00000040) /* Wake-Up Frame Received */
- #define **ETH_MACPMTCSR_MPR** ((uint32_t)0x00000020) /* Magic Packet Received */
- #define **ETH_MACPMTCSR_WFE** ((uint32_t)0x00000004) /* Wake-Up Frame Enable */
- #define **ETH_MACPMTCSR_MPE** ((uint32_t)0x00000002) /* Magic Packet Enable */
- #define **ETH_MACPMTCSR_PD** ((uint32_t)0x00000001) /* Power Down */
- #define **ETH_MACSR_TSTS** ((uint32_t)0x00000200) /* Time stamp trigger status */
- #define **ETH_MACSR_MMCTS** ((uint32_t)0x00000040) /* MMC transmit status */
- #define **ETH_MACSR_MMMCRS** ((uint32_t)0x00000020) /* MMC receive status */
- #define **ETH_MACSR_MMCS** ((uint32_t)0x00000010) /* MMC status */
- #define **ETH_MACSR_PMTS** ((uint32_t)0x00000008) /* PMT status */
- #define **ETH_MACIMR_TSTIM** ((uint32_t)0x00000020) /* Time stamp trigger interrupt mask */
- #define **ETH_MACIMR_PMTIM** ((uint32_t)0x00000008) /* PMT interrupt mask */
- #define **ETH_MACA0HR_MACA0H** ((uint32_t)0x0000FFFF) /* MAC address0 high */
- #define **ETH_MACA0LR_MACA0L** ((uint32_t)0xFFFFFFFF) /* MAC address0 low */
- #define **ETH_MACA1HR_AE** ((uint32_t)0x80000000) /* Address enable */
- #define **ETH_MACA1HR_SA** ((uint32_t)0x40000000) /* Source address */
- #define **ETH_MACA1HR_MBC** ((uint32_t)0x3F000000) /* Mask byte control: bits to mask for comparison of the MAC Address bytes */
- #define **ETH_MACA1HR_MBC_HBits15_8** ((uint32_t)0x20000000) /* Mask MAC Address high reg bits [15:8] */
- #define **ETH_MACA1HR_MBC_HBits7_0** ((uint32_t)0x10000000) /* Mask MAC Address high reg bits [7:0] */
- #define **ETH_MACA1HR_MBC_LBits31_24** ((uint32_t)0x08000000) /* Mask MAC Address low reg bits [31:24] */
- #define **ETH_MACA1HR_MBC_LBits23_16** ((uint32_t)0x04000000) /* Mask MAC Address low reg bits [23:16] */
- #define **ETH_MACA1HR_MBC_LBits15_8** ((uint32_t)0x02000000) /* Mask MAC Address low reg bits [15:8] */
- #define **ETH_MACA1HR_MBC_LBits7_0** ((uint32_t)0x01000000) /* Mask MAC Address low reg bits [7:0] */

- #define **ETH_MACA1HR_MACA1H** ((uint32_t)0x0000FFFF) /* MAC address1 high */
- #define **ETH_MACA1LR_MACA1L** ((uint32_t)0xFFFFFFFF) /* MAC address1 low */
- #define **ETH_MACA2HR_AE** ((uint32_t)0x80000000) /* Address enable */
- #define **ETH_MACA2HR_SA** ((uint32_t)0x40000000) /* Source address */
- #define **ETH_MACA2HR_MBC** ((uint32_t)0x3F000000) /* Mask byte control */
- #define **ETH_MACA2HR_MBC_HBits15_8** ((uint32_t)0x20000000) /* Mask MAC Address high reg bits [15:8] */
- #define **ETH_MACA2HR_MBC_HBits7_0** ((uint32_t)0x10000000) /* Mask MAC Address high reg bits [7:0] */
- #define **ETH_MACA2HR_MBC_LBits31_24** ((uint32_t)0x08000000) /* Mask MAC Address low reg bits [31:24] */
- #define **ETH_MACA2HR_MBC_LBits23_16** ((uint32_t)0x04000000) /* Mask MAC Address low reg bits [23:16] */
- #define **ETH_MACA2HR_MBC_LBits15_8** ((uint32_t)0x02000000) /* Mask MAC Address low reg bits [15:8] */
- #define **ETH_MACA2HR_MBC_LBits7_0** ((uint32_t)0x01000000) /* Mask MAC Address low reg bits [7:0] */
- #define **ETH_MACA2HR_MACA2H** ((uint32_t)0x0000FFFF) /* MAC address1 high */
- #define **ETH_MACA2LR_MACA2L** ((uint32_t)0xFFFFFFFF) /* MAC address2 low */
- #define **ETH_MACA3HR_AE** ((uint32_t)0x80000000) /* Address enable */
- #define **ETH_MACA3HR_SA** ((uint32_t)0x40000000) /* Source address */
- #define **ETH_MACA3HR_MBC** ((uint32_t)0x3F000000) /* Mask byte control */
- #define **ETH_MACA3HR_MBC_HBits15_8** ((uint32_t)0x20000000) /* Mask MAC Address high reg bits [15:8] */
- #define **ETH_MACA3HR_MBC_HBits7_0** ((uint32_t)0x10000000) /* Mask MAC Address high reg bits [7:0] */
- #define **ETH_MACA3HR_MBC_LBits31_24** ((uint32_t)0x08000000) /* Mask MAC Address low reg bits [31:24] */
- #define **ETH_MACA3HR_MBC_LBits23_16** ((uint32_t)0x04000000) /* Mask MAC Address low reg bits [23:16] */
- #define **ETH_MACA3HR_MBC_LBits15_8** ((uint32_t)0x02000000) /* Mask MAC Address low reg bits [15:8] */
- #define **ETH_MACA3HR_MBC_LBits7_0** ((uint32_t)0x01000000) /* Mask MAC Address low reg bits [7:0] */
- #define **ETH_MACA3HR_MACA3H** ((uint32_t)0x0000FFFF) /* MAC address3 high */
- #define **ETH_MACA3LR_MACA3L** ((uint32_t)0xFFFFFFFF) /* MAC address3 low */
- #define **ETH_MMCCR_MCFHP** ((uint32_t)0x00000020) /* MMC counter Full-Half preset */
- #define **ETH_MMCCR_MCP** ((uint32_t)0x00000010) /* MMC counter preset */
- #define **ETH_MMCCR_MCF** ((uint32_t)0x00000008) /* MMC Counter Freeze */
- #define **ETH_MMCCR_ROR** ((uint32_t)0x00000004) /* **Reset** on Read */
- #define **ETH_MMCCR_CSR** ((uint32_t)0x00000002) /* Counter Stop Rollover */
- #define **ETH_MMCCR_CR** ((uint32_t)0x00000001) /* Counters **Reset** */
- #define **ETH_MMCRIR_RGUFS** ((uint32_t)0x00020000) /* **Set** when Rx good unicast frames counter reaches half the maximum value */
- #define **ETH_MMCRIR_RFAES** ((uint32_t)0x00000040) /* **Set** when Rx alignment error counter reaches half the maximum value */
- #define **ETH_MMCRIR_RFCES** ((uint32_t)0x00000020) /* **Set** when Rx crc error counter reaches half the maximum value */
- #define **ETH_MMCTIR_TGFS** ((uint32_t)0x00200000) /* **Set** when Tx good frame count counter reaches half the maximum value */
- #define **ETH_MMCTIR_TGFMSCS** ((uint32_t)0x00008000) /* **Set** when Tx good multi col counter reaches half the maximum value */
- #define **ETH_MMCTIR_TGFSCS** ((uint32_t)0x00004000) /* **Set** when Tx good single col counter reaches half the maximum value */
- #define **ETH_MMCRIMR_RGUFM** ((uint32_t)0x00020000) /* Mask the interrupt when Rx good unicast frames counter reaches half the maximum value */
- #define **ETH_MMCRIMR_RFAEM** ((uint32_t)0x00000040) /* Mask the interrupt when when Rx alignment error counter reaches half the maximum value */

- #define **ETH_MMCRIMR_RFCEM** ((uint32_t)0x00000020) /* Mask the interrupt when Rx crc error counter reaches half the maximum value */
- #define **ETH_MMCTIMR_TGFM** ((uint32_t)0x00200000) /* Mask the interrupt when Tx good frame count counter reaches half the maximum value */
- #define **ETH_MMCTIMR_TGFMSCM** ((uint32_t)0x00008000) /* Mask the interrupt when Tx good multi col counter reaches half the maximum value */
- #define **ETH_MMCTIMR_TGFSCM** ((uint32_t)0x00004000) /* Mask the interrupt when Tx good single col counter reaches half the maximum value */
- #define **ETH_MMCTGFSCCR_TGFSCC** ((uint32_t)0xFFFFFFFF) /* Number of successfully transmitted frames after a single collision in Half-duplex mode. */
- #define **ETH_MMCTGFMSCCR_TGFMSCC** ((uint32_t)0xFFFFFFFF) /* Number of successfully transmitted frames after more than a single collision in Half-duplex mode. */
- #define **ETH_MMCTGFCR_TGFC** ((uint32_t)0xFFFFFFFF) /* Number of good frames transmitted. */
- #define **ETH_MMCRFCECR_RFCEC** ((uint32_t)0xFFFFFFFF) /* Number of frames received with **CRC** error. */
- #define **ETH_MMCRFAECR_RFAEC** ((uint32_t)0xFFFFFFFF) /* Number of frames received with alignment (dribble) error */
- #define **ETH_MMCRGUFCR_RGUFC** ((uint32_t)0xFFFFFFFF) /* Number of good unicast frames received. */
- #define **ETH_PTPTSCR_TSCNT** ((uint32_t)0x00030000) /* Time stamp clock node type */
- #define **ETH_PTPTSSR_TSSMRME** ((uint32_t)0x00008000) /* Time stamp snapshot for message relevant to master enable */
- #define **ETH_PTPTSSR_TSSEME** ((uint32_t)0x00004000) /* Time stamp snapshot for event message enable */
- #define **ETH_PTPTSSR_TSSIPV4FE** ((uint32_t)0x00002000) /* Time stamp snapshot for IPv4 frames enable */
- #define **ETH_PTPTSSR_TSSIPV6FE** ((uint32_t)0x00001000) /* Time stamp snapshot for IPv6 frames enable */
- #define **ETH_PTPTSSR_TSSPTPOEFE** ((uint32_t)0x00000800) /* Time stamp snapshot for PTP over ethernet frames enable */
- #define **ETH_PTPTSSR_TSPTPPSV2E** ((uint32_t)0x00000400) /* Time stamp PTP packet snooping for version2 format enable */
- #define **ETH_PTPTSSR_TSSSR** ((uint32_t)0x00000200) /* Time stamp Sub-seconds rollover */
- #define **ETH_PTPTSSR_TSSARFE** ((uint32_t)0x00000100) /* Time stamp snapshot for all received frames enable */
- #define **ETH_PTPTSCR_TSARU** ((uint32_t)0x00000020) /* Addend register update */
- #define **ETH_PTPTSCR_TSITE** ((uint32_t)0x00000010) /* Time stamp interrupt trigger enable */
- #define **ETH_PTPTSCR_TSSTU** ((uint32_t)0x00000008) /* Time stamp update */
- #define **ETH_PTPTSCR_TSSTI** ((uint32_t)0x00000004) /* Time stamp initialize */
- #define **ETH_PTPTSCR_TSFCU** ((uint32_t)0x00000002) /* Time stamp fine or coarse update */
- #define **ETH_PTPTSCR_TSE** ((uint32_t)0x00000001) /* Time stamp enable */
- #define **ETH_PTSSIR_STSSI** ((uint32_t)0x000000FF) /* System time Sub-second increment value */
- #define **ETH_PTPTSHR_STS** ((uint32_t)0xFFFFFFFF) /* System Time second */
- #define **ETH_PTPTSLR_STPNS** ((uint32_t)0x80000000) /* System Time Positive or negative time */
- #define **ETH_PTPTSLR_STSS** ((uint32_t)0x7FFFFFFF) /* System Time sub-seconds */
- #define **ETH_PTPTSHUR_TSUS** ((uint32_t)0xFFFFFFFF) /* Time stamp update seconds */
- #define **ETH_PTPTSLUR_TSUPNS** ((uint32_t)0x80000000) /* Time stamp update Positive or negative time */
- #define **ETH_PTPTSLUR_TSUSS** ((uint32_t)0x7FFFFFFF) /* Time stamp update sub-seconds */
- #define **ETH_PTPTSAR_TSA** ((uint32_t)0xFFFFFFFF) /* Time stamp addend */
- #define **ETH_PTPTTHR_TTSH** ((uint32_t)0xFFFFFFFF) /* Target time stamp high */
- #define **ETH_PTPTTLR_TTSL** ((uint32_t)0xFFFFFFFF) /* Target time stamp low */
- #define **ETH_PTPTSSR_TSTTR** ((uint32_t)0x00000020) /* Time stamp target time reached */
- #define **ETH_PTPTSSR_TSSO** ((uint32_t)0x00000010) /* Time stamp seconds overflow */
- #define **ETH_DMABMR_AAB** ((uint32_t)0x02000000) /* Address-Aligned beats */
- #define **ETH_DMABMR_FPM** ((uint32_t)0x01000000) /* 4xPBL mode */

- #define **ETH_DMABMR_USP** ((uint32_t)0x00800000) /* Use separate PBL */
- #define **ETH_DMABMR_RDP** ((uint32_t)0x007E0000) /* RxDMA PBL */
- #define **ETH_DMABMR_RDP_1Beat** ((uint32_t)0x00020000) /* maximum number of beats to be transferred in one RxDMA transaction is 1 */
- #define **ETH_DMABMR_RDP_2Beat** ((uint32_t)0x00040000) /* maximum number of beats to be transferred in one RxDMA transaction is 2 */
- #define **ETH_DMABMR_RDP_4Beat** ((uint32_t)0x00080000) /* maximum number of beats to be transferred in one RxDMA transaction is 4 */
- #define **ETH_DMABMR_RDP_8Beat** ((uint32_t)0x00100000) /* maximum number of beats to be transferred in one RxDMA transaction is 8 */
- #define **ETH_DMABMR_RDP_16Beat** ((uint32_t)0x00200000) /* maximum number of beats to be transferred in one RxDMA transaction is 16 */
- #define **ETH_DMABMR_RDP_32Beat** ((uint32_t)0x00400000) /* maximum number of beats to be transferred in one RxDMA transaction is 32 */
- #define **ETH_DMABMR_RDP_4xPBL_4Beat** ((uint32_t)0x01020000) /* maximum number of beats to be transferred in one RxDMA transaction is 4 */
- #define **ETH_DMABMR_RDP_4xPBL_8Beat** ((uint32_t)0x01040000) /* maximum number of beats to be transferred in one RxDMA transaction is 8 */
- #define **ETH_DMABMR_RDP_4xPBL_16Beat** ((uint32_t)0x01080000) /* maximum number of beats to be transferred in one RxDMA transaction is 16 */
- #define **ETH_DMABMR_RDP_4xPBL_32Beat** ((uint32_t)0x01100000) /* maximum number of beats to be transferred in one RxDMA transaction is 32 */
- #define **ETH_DMABMR_RDP_4xPBL_64Beat** ((uint32_t)0x01200000) /* maximum number of beats to be transferred in one RxDMA transaction is 64 */
- #define **ETH_DMABMR_RDP_4xPBL_128Beat** ((uint32_t)0x01400000) /* maximum number of beats to be transferred in one RxDMA transaction is 128 */
- #define **ETH_DMABMR_FB** ((uint32_t)0x00010000) /* Fixed Burst */
- #define **ETH_DMABMR_RTPR** ((uint32_t)0x0000C000) /* Rx Tx priority ratio */
- #define **ETH_DMABMR_RTPR_1_1** ((uint32_t)0x00000000) /* Rx Tx priority ratio */
- #define **ETH_DMABMR_RTPR_2_1** ((uint32_t)0x00004000) /* Rx Tx priority ratio */
- #define **ETH_DMABMR_RTPR_3_1** ((uint32_t)0x00008000) /* Rx Tx priority ratio */
- #define **ETH_DMABMR_RTPR_4_1** ((uint32_t)0x0000C000) /* Rx Tx priority ratio */
- #define **ETH_DMABMR_PBL** ((uint32_t)0x00003F00) /* Programmable burst length */
- #define **ETH_DMABMR_PBL_1Beat** ((uint32_t)0x00000100) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 1 */
- #define **ETH_DMABMR_PBL_2Beat** ((uint32_t)0x00000200) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 2 */
- #define **ETH_DMABMR_PBL_4Beat** ((uint32_t)0x00000400) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 4 */
- #define **ETH_DMABMR_PBL_8Beat** ((uint32_t)0x00000800) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 8 */
- #define **ETH_DMABMR_PBL_16Beat** ((uint32_t)0x00001000) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 16 */
- #define **ETH_DMABMR_PBL_32Beat** ((uint32_t)0x00002000) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 32 */
- #define **ETH_DMABMR_PBL_4xPBL_4Beat** ((uint32_t)0x01000100) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 4 */
- #define **ETH_DMABMR_PBL_4xPBL_8Beat** ((uint32_t)0x01000200) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 8 */
- #define **ETH_DMABMR_PBL_4xPBL_16Beat** ((uint32_t)0x01000400) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 16 */
- #define **ETH_DMABMR_PBL_4xPBL_32Beat** ((uint32_t)0x01000800) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 32 */
- #define **ETH_DMABMR_PBL_4xPBL_64Beat** ((uint32_t)0x01001000) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 64 */

- #define **ETH_DMABMR_PBL_4xPBL_128Beat** ((uint32_t)0x01002000) /* maximum number of beats to be transferred in one TxDMA (or both) transaction is 128 */
- #define **ETH_DMABMR_EDE** ((uint32_t)0x00000080) /* Enhanced Descriptor Enable */
- #define **ETH_DMABMR_DSL** ((uint32_t)0x0000007C) /* Descriptor Skip Length */
- #define **ETH_DMABMR_DA** ((uint32_t)0x00000002) /* DMA arbitration scheme */
- #define **ETH_DMABMR_SR** ((uint32_t)0x00000001) /* Software reset */
- #define **ETH_DMATPDR_TPD** ((uint32_t)0xFFFFFFF) /* Transmit poll demand */
- #define **ETH_DMARPDR_RPD** ((uint32_t)0xFFFFFFFF) /* Receive poll demand */
- #define **ETH_DMARDLAR_SRL** ((uint32_t)0xFFFFFFFF) /* Start of receive list */
- #define **ETH_DMATDLAR_STL** ((uint32_t)0xFFFFFFFF) /* Start of transmit list */
- #define **ETH_DMASR_TSTS** ((uint32_t)0x20000000) /* Time-stamp trigger status */
- #define **ETH_DMASR_PMTS** ((uint32_t)0x10000000) /* PMT status */
- #define **ETH_DMASR_MMCS** ((uint32_t)0x08000000) /* MMC status */
- #define **ETH_DMASR_EBS** ((uint32_t)0x03800000) /* Error bits status */
- #define **ETH_DMASR_EBS_DescAccess** ((uint32_t)0x02000000) /* Error bits 0-data buffer, 1-desc. access */
- #define **ETH_DMASR_EBS_ReadTransf** ((uint32_t)0x01000000) /* Error bits 0-write trnsf, 1-read transfr */
- #define **ETH_DMASR_EBS_DataTransfTx** ((uint32_t)0x00800000) /* Error bits 0-Rx DMA, 1-Tx DMA */
- #define **ETH_DMASR_TPS** ((uint32_t)0x00700000) /* Transmit process state */
- #define **ETH_DMASR_TPS_Stopped** ((uint32_t)0x00000000) /* Stopped - Reset or Stop Tx Command issued */
- #define **ETH_DMASR_TPS_Fetching** ((uint32_t)0x00100000) /* Running - fetching the Tx descriptor */
- #define **ETH_DMASR_TPS_Waiting** ((uint32_t)0x00200000) /* Running - waiting for status */
- #define **ETH_DMASR_TPS_Reading** ((uint32_t)0x00300000) /* Running - reading the data from host memory */
- #define **ETH_DMASR_TPS_Suspended** ((uint32_t)0x00600000) /* Suspended - Tx Descriptor unavailable */
- #define **ETH_DMASR_TPS_Closing** ((uint32_t)0x00700000) /* Running - closing Rx descriptor */
- #define **ETH_DMASR_RPS** ((uint32_t)0x000E0000) /* Receive process state */
- #define **ETH_DMASR_RPS_Stopped** ((uint32_t)0x00000000) /* Stopped - Reset or Stop Rx Command issued */
- #define **ETH_DMASR_RPS_Fetching** ((uint32_t)0x00020000) /* Running - fetching the Rx descriptor */
- #define **ETH_DMASR_RPS_Waiting** ((uint32_t)0x00060000) /* Running - waiting for packet */
- #define **ETH_DMASR_RPS_Suspended** ((uint32_t)0x00080000) /* Suspended - Rx Descriptor unavailable */
- #define **ETH_DMASR_RPS_Closing** ((uint32_t)0x000A0000) /* Running - closing descriptor */
- #define **ETH_DMASR_RPS_Queueing** ((uint32_t)0x000E0000) /* Running - queuing the receive frame into host memory */
- #define **ETH_DMASR_NIS** ((uint32_t)0x00010000) /* Normal interrupt summary */
- #define **ETH_DMASR_AIS** ((uint32_t)0x00008000) /* Abnormal interrupt summary */
- #define **ETH_DMASR_ERS** ((uint32_t)0x00004000) /* Early receive status */
- #define **ETH_DMASR_FBES** ((uint32_t)0x00002000) /* Fatal bus error status */
- #define **ETH_DMASR_ETS** ((uint32_t)0x00000400) /* Early transmit status */
- #define **ETH_DMASR_RWTS** ((uint32_t)0x00000200) /* Receive watchdog timeout status */
- #define **ETH_DMASR_RPSS** ((uint32_t)0x00000100) /* Receive process stopped status */
- #define **ETH_DMASR_RBUS** ((uint32_t)0x00000080) /* Receive buffer unavailable status */
- #define **ETH_DMASR_RS** ((uint32_t)0x00000040) /* Receive status */
- #define **ETH_DMASR_TUS** ((uint32_t)0x00000020) /* Transmit underflow status */
- #define **ETH_DMASR_ROS** ((uint32_t)0x00000010) /* Receive overflow status */
- #define **ETH_DMASR_TJTS** ((uint32_t)0x00000008) /* Transmit jabber timeout status */
- #define **ETH_DMASR_TBUS** ((uint32_t)0x00000004) /* Transmit buffer unavailable status */
- #define **ETH_DMASR_TPSS** ((uint32_t)0x00000002) /* Transmit process stopped status */
- #define **ETH_DMASR_TS** ((uint32_t)0x00000001) /* Transmit status */
- #define **ETH_DMAOMR_DTCEFD** ((uint32_t)0x04000000) /* Disable Dropping of TCP/IP checksum error frames */

- #define **ETH_DMAOMR_RSF** ((uint32_t)0x02000000) /* Receive store and forward */
- #define **ETH_DMAOMR_DFRF** ((uint32_t)0x01000000) /* Disable flushing of received frames */
- #define **ETH_DMAOMR_TSF** ((uint32_t)0x00200000) /* Transmit store and forward */
- #define **ETH_DMAOMR_FTF** ((uint32_t)0x00100000) /* Flush transmit FIFO */
- #define **ETH_DMAOMR_TTC** ((uint32_t)0x0001C000) /* Transmit threshold control */
- #define **ETH_DMAOMR_TTC_64Bytes** ((uint32_t)0x00000000) /* threshold level of the MTL Transmit FIFO is 64 Bytes */
- #define **ETH_DMAOMR_TTC_128Bytes** ((uint32_t)0x00004000) /* threshold level of the MTL Transmit FIFO is 128 Bytes */
- #define **ETH_DMAOMR_TTC_192Bytes** ((uint32_t)0x00008000) /* threshold level of the MTL Transmit FIFO is 192 Bytes */
- #define **ETH_DMAOMR_TTC_256Bytes** ((uint32_t)0x0000C000) /* threshold level of the MTL Transmit FIFO is 256 Bytes */
- #define **ETH_DMAOMR_TTC_40Bytes** ((uint32_t)0x00010000) /* threshold level of the MTL Transmit FIFO is 40 Bytes */
- #define **ETH_DMAOMR_TTC_32Bytes** ((uint32_t)0x00014000) /* threshold level of the MTL Transmit FIFO is 32 Bytes */
- #define **ETH_DMAOMR_TTC_24Bytes** ((uint32_t)0x00018000) /* threshold level of the MTL Transmit FIFO is 24 Bytes */
- #define **ETH_DMAOMR_TTC_16Bytes** ((uint32_t)0x0001C000) /* threshold level of the MTL Transmit FIFO is 16 Bytes */
- #define **ETH_DMAOMR_ST** ((uint32_t)0x00002000) /* Start/stop transmission command */
- #define **ETH_DMAOMR_FEF** ((uint32_t)0x00000080) /* Forward error frames */
- #define **ETH_DMAOMR_FUGF** ((uint32_t)0x00000040) /* Forward undersized good frames */
- #define **ETH_DMAOMR_RTC** ((uint32_t)0x00000018) /* receive threshold control */
- #define **ETH_DMAOMR_RTC_64Bytes** ((uint32_t)0x00000000) /* threshold level of the MTL Receive FIFO is 64 Bytes */
- #define **ETH_DMAOMR_RTC_32Bytes** ((uint32_t)0x00000008) /* threshold level of the MTL Receive FIFO is 32 Bytes */
- #define **ETH_DMAOMR_RTC_96Bytes** ((uint32_t)0x00000010) /* threshold level of the MTL Receive FIFO is 96 Bytes */
- #define **ETH_DMAOMR_RTC_128Bytes** ((uint32_t)0x00000018) /* threshold level of the MTL Receive FIFO is 128 Bytes */
- #define **ETH_DMAOMR_OSF** ((uint32_t)0x00000004) /* operate on second frame */
- #define **ETH_DMAOMR_SR** ((uint32_t)0x00000002) /* Start/stop receive */
- #define **ETH_DMAIER_NISE** ((uint32_t)0x00010000) /* Normal interrupt summary enable */
- #define **ETH_DMAIER_AISE** ((uint32_t)0x00008000) /* Abnormal interrupt summary enable */
- #define **ETH_DMAIER_ERIE** ((uint32_t)0x00004000) /* Early receive interrupt enable */
- #define **ETH_DMAIER_FBEIE** ((uint32_t)0x00002000) /* Fatal bus error interrupt enable */
- #define **ETH_DMAIER_ETIE** ((uint32_t)0x00000400) /* Early transmit interrupt enable */
- #define **ETH_DMAIER_RWTIE** ((uint32_t)0x00000200) /* Receive watchdog timeout interrupt enable */
- #define **ETH_DMAIER_RPSIE** ((uint32_t)0x00000100) /* Receive process stopped interrupt enable */
- #define **ETH_DMAIER_RBUIE** ((uint32_t)0x00000080) /* Receive buffer unavailable interrupt enable */
- #define **ETH_DMAIER_RIE** ((uint32_t)0x00000040) /* Receive interrupt enable */
- #define **ETH_DMAIER_TUIE** ((uint32_t)0x00000020) /* Transmit Underflow interrupt enable */
- #define **ETH_DMAIER_ROIE** ((uint32_t)0x00000010) /* Receive Overflow interrupt enable */
- #define **ETH_DMAIER_TJTIE** ((uint32_t)0x00000008) /* Transmit jabber timeout interrupt enable */
- #define **ETH_DMAIER_TBUIE** ((uint32_t)0x00000004) /* Transmit buffer unavailable interrupt enable */
- #define **ETH_DMAIER_TPSIE** ((uint32_t)0x00000002) /* Transmit process stopped interrupt enable */
- #define **ETH_DMAIER_TIE** ((uint32_t)0x00000001) /* Transmit interrupt enable */
- #define **ETH_DMAMFBOCR_OFOC** ((uint32_t)0x10000000) /* Overflow bit for FIFO overflow counter */
- #define **ETH_DMAMFBOCR_MFA** ((uint32_t)0x0FFE0000) /* Number of frames missed by the application */
- #define **ETH_DMAMFBOCR_OMFC** ((uint32_t)0x00010000) /* Overflow bit for missed frame counter */
- #define **ETH_DMAMFBOCR_MFC** ((uint32_t)0x0000FFFF) /* Number of frames missed by the controller */

- #define **ETH_DMACHTDR_HTDAP** ((uint32_t)0xFFFFFFFF) /* Host transmit descriptor address pointer */
- #define **ETH_DMACHRDR_HRDAP** ((uint32_t)0xFFFFFFFF) /* Host receive descriptor address pointer */
- #define **ETH_DMACHTBAR_HTBAP** ((uint32_t)0xFFFFFFFF) /* Host transmit buffer address pointer */
- #define **ETH_DMACHRBAR_HRBAP** ((uint32_t)0xFFFFFFFF) /* Host receive buffer address pointer */
- #define **SET_BIT**(REG, BIT) ((REG) |= (BIT))
- #define **CLEAR_BIT**(REG, BIT) ((REG) &= ~ (BIT))
- #define **READ_BIT**(REG, BIT) ((REG) & (BIT))
- #define **CLEAR_REG**(REG) ((REG) = (0x0))
- #define **WRITE_REG**(REG, VAL) ((REG) = (VAL))
- #define **READ_REG**(REG) ((REG))
- #define **MODIFY_REG**(REG, CLEARMASK, SETMASK) **WRITE_REG**((REG), (((**READ_REG**(REG)) & (~(CLEARMASK))) | (SETMASK)))

Typedefs

- typedef enum **IRQn IRQn_Type**
STM32F4XX Interrupt Number Definition, according to the selected device in [Library configuration section](#).
- typedef int32_t **s32**
- typedef int16_t **s16**
- typedef int8_t **s8**
- typedef const int32_t **sc32**
- typedef const int16_t **sc16**
- typedef const int8_t **sc8**
- typedef **_IO** int32_t **vs32**
- typedef **_IO** int16_t **vs16**
- typedef **_IO** int8_t **vs8**
- typedef **_I** int32_t **vsc32**
- typedef **_I** int16_t **vsc16**
- typedef **_I** int8_t **vsc8**
- typedef uint32_t **u32**
- typedef uint16_t **u16**
- typedef uint8_t **u8**
- typedef const uint32_t **uc32**
- typedef const uint16_t **uc16**
- typedef const uint8_t **uc8**
- typedef **_IO** uint32_t **vu32**
- typedef **_IO** uint16_t **vu16**
- typedef **_IO** uint8_t **vu8**
- typedef **_I** uint32_t **vuc32**
- typedef **_I** uint16_t **vuc16**
- typedef **_I** uint8_t **vuc8**
- typedef enum **FlagStatus ITStatus**

Enumerations

- enum **IRQn** {
 NonMaskableInt_IRQn = -14 , **MemoryManagement_IRQn** = -12 , **BusFault_IRQn** = -11 , **UsageFault_IRQn** = -10 ,
 SVCall_IRQn = -5 , **DebugMonitor_IRQn** = -4 , **PendSV_IRQn** = -2 , **SysTick_IRQn** = -1 ,
 WWDG_IRQn = 0 , **PVD_IRQn** = 1 , **TAMP_STAMP_IRQn** = 2 , **RTC_WKUP_IRQn** = 3 ,
 FLASH_IRQn = 4 , **RCC_IRQn** = 5 , **EXTI0_IRQn** = 6 , **EXTI1_IRQn** = 7 ,
 EXTI2_IRQn = 8 , **EXTI3_IRQn** = 9 , **EXTI4_IRQn** = 10 , **DMA1_Stream0_IRQn** = 11 ,
 DMA1_Stream1_IRQn = 12 , **DMA1_Stream2_IRQn** = 13 , **DMA1_Stream3_IRQn** = 14 , **DMA1_Stream4_IRQn**

```
= 15 ,
DMA1_Stream5_IRQn = 16 , DMA1_Stream6_IRQn = 17 , ADC_IRQn = 18 , EXTI9_5_IRQn = 23 ,
TIM1_BRK_TIM9_IRQn = 24 , TIM1_UP_TIM10_IRQn = 25 , TIM1_TRG_COM_TIM11_IRQn = 26 ,
TIM1_CC_IRQn = 27 ,
TIM2_IRQn = 28 , TIM3_IRQn = 29 , TIM4_IRQn = 30 , I2C1_EV_IRQn = 31 ,
I2C1_ER_IRQn = 32 , I2C2_EV_IRQn = 33 , I2C2_ER_IRQn = 34 , SPI1_IRQn = 35 ,
SPI2_IRQn = 36 , USART1_IRQn = 37 , USART2_IRQn = 38 , EXTI15_10_IRQn = 40 ,
RTC_Alarm_IRQn = 41 , OTG_FS_WKUP_IRQn = 42 , DMA1_Stream7_IRQn = 47 , SDIO_IRQn = 49 ,
TIM5_IRQn = 50 , SPI3_IRQn = 51 , DMA2_Stream0_IRQn = 56 , DMA2_Stream1_IRQn = 57 ,
DMA2_Stream2_IRQn = 58 , DMA2_Stream3_IRQn = 59 , DMA2_Stream4_IRQn = 60 , OTG_FS_IRQn = 67 ,
DMA2_Stream5_IRQn = 68 , DMA2_Stream6_IRQn = 69 , DMA2_Stream7_IRQn = 70 , USART6_IRQn = 71 ,
I2C3_EV_IRQn = 72 , I2C3_ER_IRQn = 73 , FPU_IRQn = 81 , SPI4_IRQn = 84 }
```

STM32F4XX Interrupt Number Definition, according to the selected device in [Library configuration section](#).

- enum **FlagStatus** { **RESET** = 0 , **SET** = !**RESET** , **Reset** , **Set** }
- enum **FunctionalState** { **DISABLE** = 0 , **ENABLE** = !**DISABLE** }
- enum **ErrorStatus** { **ERROR** = 0 , **SUCCESS** = !**ERROR** }

7.13.1 Detailed Description

CMSIS Cortex-M4 Device Peripheral Access Layer Header File. This file contains all the peripheral register's definitions, bits definitions and memory mapping for STM32F4xx devices.

Author

MCD Application Team

Version

V1.8.0

Date

09-November-2016

The file is the unique include file that the application programmer is using in the C source code, usually in main.c. This file contains:

- Configuration section that allows to select:
 - The device used in the target application
 - To use or not the peripheral's drivers in application code(i.e. code will be based on direct access to peripheral's registers rather than drivers API), this option is controlled by "#define USE_STDPERIPH_DRIVER"
 - To change few application-specific parameters such as the HSE crystal frequency
- Data structures and the address mapping for all peripherals
- Peripherals registers declarations and bits definition
- Macros to access peripheral's registers hardware

Attention

© COPYRIGHT 2016 STMicroelectronics

Licensed under MCD-ST Liberty SW License Agreement V2, (the "License"); You may not use this file except in compliance with the License. You may obtain a copy of the License at:

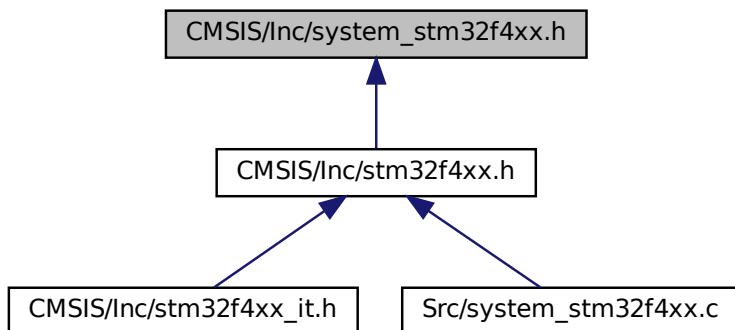
http://www.st.com/software_license_agreement_liberty_v2

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

7.14 CMSIS/Inc/system_stm32f4xx.h File Reference

CMSIS Cortex-M4 Device System Source File for STM32F4xx devices.

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



Functions

- void [SystemInit](#) (void)
Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void [SystemCoreClockUpdate](#) (void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Variables

- uint32_t [SystemCoreClock](#)

7.14.1 Detailed Description

CMSIS Cortex-M4 Device System Source File for STM32F4xx devices.

Author

MCD Application Team

Version

V1.8.0

Date

09-November-2016

Attention

© COPYRIGHT 2016 STMicroelectronics

Licensed under MCD-ST Liberty SW License Agreement V2, (the "License"); You may not use this file except in compliance with the License. You may obtain a copy of the License at:

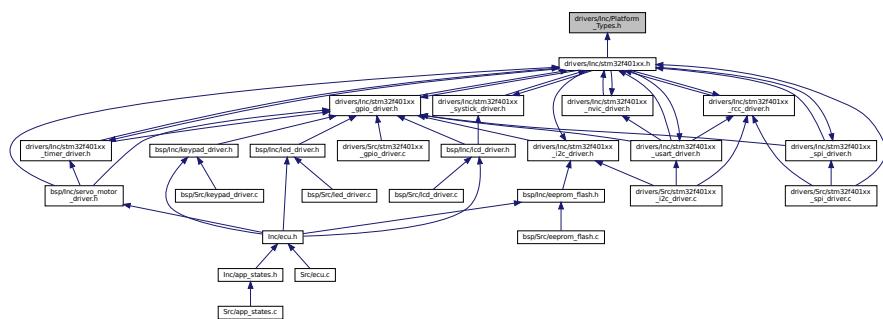
http://www.st.com/software_license_agreement_liberty_v2

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

7.15 drivers/Inc/Platform_Types.h File Reference

Defines data types and macros for STM32F401xx microcontrollers.

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



Macros

- `#define CPU_TYPE_8 8`
CPU Type Definitions.
- `#define CPU_TYPE_16 16`
- `#define CPU_TYPE_32 32`
- `#define CPU_TYPE_64 64`
- `#define MSB_FIRST 0`
Bit Order Definitions.
- `#define LSB_FIRST 1`
- `#define HIGH_BYTE_FIRST 0`
Byte Order Definitions.
- `#define LOW_BYTE_FIRST 1`
- `#define CPU_TYPE CPU_TYPE_32`
Selected CPU Characteristics.
- `#define CPU_BIT_ORDER LSB_FIRST`
- `#define CPU_BYTE_ORDER LOW_BYTE_FIRST`
- `#define TRUE 1`
Boolean Constants.
- `#define FALSE 0`
- `#define SET TRUE`
- `#define RESET FALSE`
- `#define FLAG_SET SET`
- `#define FLAG_RESET RESET`

Typedefs

- `typedef unsigned char boolean`
Standard Data Types.
- `typedef signed char sint8`
- `typedef unsigned char uint8`
- `typedef signed short sint16`
- `typedef unsigned short uint16`
- `typedef signed long sint32`
- `typedef signed long long sint64`
- `typedef unsigned long uint32`
- `typedef unsigned long long uint64`
- `typedef unsigned long uint8_least`
Least Type Definitions.
- `typedef unsigned long uint16_least`
- `typedef unsigned long uint32_least`
- `typedef signed long sint8_least`
- `typedef signed long sint16_least`
- `typedef signed long sint32_least`
- `typedef float float32`
Floating-Point Types.
- `typedef double float64`
- `typedef void * VoidPtr`
Pointer Types.
- `typedef const void * ConstVoidPtr`
- `typedef volatile unsigned char vuint8_t`
Volatile Types.
- `typedef volatile unsigned short vuint16_t`
- `typedef volatile unsigned long vuint32_t`

7.15.1 Detailed Description

Defines data types and macros for STM32F401xx microcontrollers.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This header file provides standard data types, CPU-specific types, and macros for use with STM32F401xx series microcontrollers. It includes definitions for CPU bit and byte order, as well as various data type definitions suited for the platform.

Note

This file is specifically tailored for the STM32F401xx family. Ensure compatibility with other microcontroller series before use.

Copyright

Copyright (c) 2024 Mohamed Ali Haoufa

7.16 drivers/Inc/stm32f401xx.h File Reference

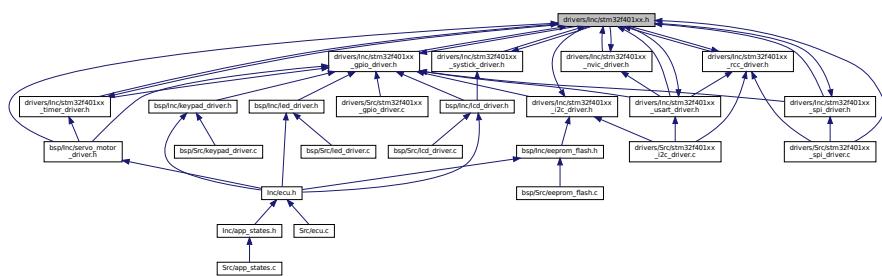
Header file containing all the necessary information about the STM32F401xx MCU.

```
#include <stdlib.h>
#include <stdint.h>
#include <stdio.h>
#include "Platform_Types.h"
#include "stm32f401xx_gpio_driver.h"
#include "stm32f401xx_rcc_driver.h"
#include "stm32f401xx_usart_driver.h"
#include "stm32f401xx_i2c_driver.h"
#include "stm32f401xx_spi_driver.h"
#include "stm32f401xx_nvic_driver.h"
#include "stm32f401xx_systick_driver.h"
#include "stm32f401xx_timer_driver.h"
```

Include dependency graph for stm32f401xx.h:



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



Classes

- struct **NVIC_TypeDef**
 - struct **SCB_TypeDef**
 - struct **STK_TypeDef**
 - struct **GPIO_TypeDef**
 - General Purpose I/O.*
 - struct **RCC_TypeDef**
 - Reset and Clock Control.*
 - struct **EXTI_TypeDef**
 - External Interrupt/Event Controller.*
 - struct **SYSCFG_RegDef_t**
 - struct **USART_TypeDef**
 - Universal Synchronous Asynchronous Receiver Transmitter.*
 - struct **SPI_TypeDef**
 - Serial Peripheral Interface.*
 - struct **I2C_TypeDef**
 - Inter-integrated Circuit Interface.*
 - struct **CRC_TypeDef**
 - CRC calculation unit.*
 - struct **TIM1_TypeDef**

Macros

- `#define FLASH_MEMORY_BASE 0x08000000UL`
Base address of Flash memory.
 - `#define SYSTEM_MEMORY_BASE 0x1FFFFF000UL`
Base address of System memory.
 - `#define SRAM_MEMORY_BASE 0x20000000UL`
Base address of SRAM memory.
 - `#define PERIPHERALS_BASE 0x40000000UL`
Base address of Peripheral registers.
 - `#define CORTEX_M4_INTERNAL_BASE 0xE0000000UL`
Base address of Cortex-M4 internal peripherals.
 - `#define NVIC_BASE 0xE000E100UL`
Base address of NVIC (Nested Vectored Interrupt Controller)
 - `#define SCB_BASE 0xE000ED00UL`

- #define **STK_BASE** 0xE000E010UL
Base address of System Control Block (SCB)
- #define **RCC_BASE** 0x40023800UL
Base address of SysTick Timer.
- #define **CRC_BASE** 0x40023000UL
Base address of RCC (Reset and Clock Control)
- #define **GPIOA_BASE** 0x40020000UL
Base address of GPIOA (General Purpose I/O port A)
- #define **GPIOB_BASE** 0x40020400UL
Base address of GPIOB (General Purpose I/O port B)
- #define **GPIOC_BASE** 0x40020800UL
Base address of GPIOC (General Purpose I/O port C)
- #define **GPIOD_BASE** 0x40020C00UL
Base address of GPIOD (General Purpose I/O port D)
- #define **GPIOE_BASE** 0x40021000UL
Base address of GPIOE (General Purpose I/O port E)
- #define **EXTI_BASE** 0x40013C00UL
Base address of EXTI (External Interrupt/Event Controller)
- #define **TIM1_BASE** 0x40010000UL
Base address of TIM1 (Timer 1)
- #define **TIM2_BASE** 0x40000000UL
Base address of TIM2 (Timer 2)
- #define **USART1_BASE** 0x40011000UL
Base address of USART1 (Universal Synchronous/Asynchronous Receiver Transmitter 1)
- #define **USART6_BASE** 0x40011400UL
Base address of USART6 (Universal Synchronous/Asynchronous Receiver Transmitter 6)
- #define **SPI1_BASE** 0x40013000UL
Base address of SPI1 (Serial Peripheral Interface 1)
- #define **SYSCFG_BASE** 0x40013800UL
Base address of SYSCFG (System Configuration Controller)
- #define **USART2_BASE** 0x40004400UL
Base address of USART2 (Universal Synchronous/Asynchronous Receiver Transmitter 2)
- #define **SPI2_BASE** 0x40003800UL
Base address of SPI2 (Serial Peripheral Interface 2)
- #define **I2C1_BASE** 0x40005400UL
Base address of I2C1 (Inter-Integrated Circuit 1)
- #define **I2C2_BASE** 0x40005800UL
Base address of I2C2 (Inter-Integrated Circuit 2)
- #define **NVIC ((NVIC_TypeDef*)NVIC_BASE)**
- #define **SCB ((SCB_TypeDef*)SCB_BASE)**
- #define **STK ((STK_TypeDef*)STK_BASE)**
- #define **GPIOA ((GPIO_TypeDef*)GPIOA_BASE)**
- #define **GPIOB ((GPIO_TypeDef*)GPIOB_BASE)**
- #define **GPIOC ((GPIO_TypeDef*)GPIOC_BASE)**
- #define **GPIOD ((GPIO_TypeDef*)GPIOD_BASE)**
- #define **GPIOE ((GPIO_TypeDef*)GPIOE_BASE)**
- #define **RCC ((RCC_TypeDef*)RCC_BASE)**
- #define **EXTI ((EXTI_TypeDef*)EXTI_BASE)**
- #define **SYSCFG ((SYSCFG_RegDef_t*)SYSCFG_BASE)**
- #define **TIM1 ((TIM1_TypeDef*)TIM1_BASE)**
- #define **TIM2 ((TIM1_TypeDef*)TIM2_BASE)**

- #define USART1 ((USART_TypeDef*)USART1_BASE)
- #define USART2 ((USART_TypeDef*)USART2_BASE)
- #define USART6 ((USART_TypeDef*)USART6_BASE)
- #define SPI1 ((SPI_TypeDef*)SPI1_BASE)
- #define SPI2 ((SPI_TypeDef*)SPI2_BASE)
- #define I2C1 ((I2C_TypeDef*)I2C1_BASE)
- #define I2C2 ((I2C_TypeDef*)I2C2_BASE)
- #define CRC ((CRC_TypeDef*)CRC_BASE)
- #define I2C_CR1_PE_Pos (0U)

Position and mask for the Peripheral Enable bit in the I2C_CR1 register.

- #define I2C_CR1_PE_Msk (0x1UL << I2C_CR1_PE_Pos)
- #define I2C_CR1_PE I2C_CR1_PE_Msk
- #define I2C_CR1_SMBUS_Pos (1U)

Position and mask for the SMBus Mode bit in the I2C_CR1 register.

- #define I2C_CR1_SMBUS_Msk (0x1UL << I2C_CR1_SMBUS_Pos)
- #define I2C_CR1_SMBUS I2C_CR1_SMBUS_Msk
- #define I2C_CR1_SMBTYPE_Pos (3U)

Position and mask for the SMBus Type bit in the I2C_CR1 register.

- #define I2C_CR1_SMBTYPE_Msk (0x1UL << I2C_CR1_SMBTYPE_Pos)
- #define I2C_CR1_SMBTYPE I2C_CR1_SMBTYPE_Msk
- #define I2C_CR1_ENARP_Pos (4U)

Position and mask for the ARP Enable bit in the I2C_CR1 register.

- #define I2C_CR1_ENARP_Msk (0x1UL << I2C_CR1_ENARP_Pos)
- #define I2C_CR1_ENARP I2C_CR1_ENARP_Msk
- #define I2C_CR1_ENPEC_Pos (5U)

Position and mask for the PEC Enable bit in the I2C_CR1 register.

- #define I2C_CR1_ENPEC_Msk (0x1UL << I2C_CR1_ENPEC_Pos)
- #define I2C_CR1_ENPEC I2C_CR1_ENPEC_Msk
- #define I2C_CR1_ENGC_Pos (6U)

Position and mask for the General Call Enable bit in the I2C_CR1 register.

- #define I2C_CR1_ENGC_Msk (0x1UL << I2C_CR1_ENGC_Pos)
- #define I2C_CR1_ENGC I2C_CR1_ENGC_Msk
- #define I2C_CR1_NOSTRETCH_Pos (7U)

Position and mask for the Clock Stretching Disable bit in the I2C_CR1 register.

- #define I2C_CR1_NOSTRETCH_Msk (0x1UL << I2C_CR1_NOSTRETCH_Pos)
- #define I2C_CR1_NOSTRETCH I2C_CR1_NOSTRETCH_Msk
- #define I2C_CR1_START_Pos (8U)

Position and mask for the Start Generation bit in the I2C_CR1 register.

- #define I2C_CR1_START_Msk (0x1UL << I2C_CR1_START_Pos)
- #define I2C_CR1_START I2C_CR1_START_Msk
- #define I2C_CR1_STOP_Pos (9U)

Position and mask for the Stop Generation bit in the I2C_CR1 register.

- #define I2C_CR1_STOP_Msk (0x1UL << I2C_CR1_STOP_Pos)
- #define I2C_CR1_STOP I2C_CR1_STOP_Msk
- #define I2C_CR1_ACK_Pos (10U)

Position and mask for the Acknowledge Enable bit in the I2C_CR1 register.

- #define I2C_CR1_ACK_Msk (0x1UL << I2C_CR1_ACK_Pos)
- #define I2C_CR1_ACK I2C_CR1_ACK_Msk
- #define I2C_CR1_POS_Pos (11U)

Position and mask for the Acknowledge/PEC Position bit in the I2C_CR1 register.

- #define I2C_CR1_POS_Msk (0x1UL << I2C_CR1_POS_Pos)
- #define I2C_CR1_POS I2C_CR1_POS_Msk
- #define I2C_CR1_PEC_Pos (12U)

Position and mask for the Packet Error Checking bit in the I2C_CR1 register.

- #define I2C_CR1_PEC_Msk (0x1UL << I2C_CR1_PEC_Pos)
- #define I2C_CR1_PEC I2C_CR1_PEC_Msk
- #define I2C_CR1_ALERT_Pos (13U)

Position and mask for the SMBus Alert bit in the I2C_CR1 register.

- #define I2C_CR1_ALERT_Msk (0x1UL << I2C_CR1_ALERT_Pos)
- #define I2C_CR1_ALERT I2C_CR1_ALERT_Msk
- #define I2C_CR1_SWRST_Pos (15U)

Position and mask for the Software Reset bit in the I2C_CR1 register.

- #define I2C_CR1_SWRST_Msk (0x1UL << I2C_CR1_SWRST_Pos)
- #define I2C_CR1_SWRST I2C_CR1_SWRST_Msk
- #define I2C_CR2_FREQ_Pos (0U)

Position and mask for the Peripheral Clock Frequency bits in the I2C_CR2 register.

- #define I2C_CR2_FREQ_Msk (0x3FUL << I2C_CR2_FREQ_Pos)
- #define I2C_CR2_FREQ I2C_CR2_FREQ_Msk
- #define I2C_CR2_ITERREN_Pos (8U)

Position and mask for the Error Interrupt Enable bit in the I2C_CR2 register.

- #define I2C_CR2_ITERREN_Msk (0x1UL << I2C_CR2_ITERREN_Pos)
- #define I2C_CR2_ITERREN I2C_CR2_ITERREN_Msk
- #define I2C_CR2_IteVTEN_Pos (9U)

Position and mask for the Event Interrupt Enable bit in the I2C_CR2 register.

- #define I2C_CR2_IteVTEN_Msk (0x1UL << I2C_CR2_IteVTEN_Pos)
- #define I2C_CR2_IteVTEN I2C_CR2_IteVTEN_Msk
- #define I2C_CR2_ITBUFEN_Pos (10U)

Position and mask for the Buffer Interrupt Enable bit in the I2C_CR2 register.

- #define I2C_CR2_ITBUFEN_Msk (0x1UL << I2C_CR2_ITBUFEN_Pos)
- #define I2C_CR2_ITBUFEN I2C_CR2_ITBUFEN_Msk
- #define I2C_CR2_DMAEN_Pos (11U)

Position and mask for the DMA Requests Enable bit in the I2C_CR2 register.

- #define I2C_CR2_DMAEN_Msk (0x1UL << I2C_CR2_DMAEN_Pos)
- #define I2C_CR2_DMAEN I2C_CR2_DMAEN_Msk
- #define I2C_CR2_LAST_Pos (12U)

Position and mask for the Last DMA Transfer bit in the I2C_CR2 register.

- #define I2C_CR2_LAST_Msk (0x1UL << I2C_CR2_LAST_Pos)
- #define I2C_CR2_LAST I2C_CR2_LAST_Msk
- #define I2C_OAR2_ENDUAL_Pos (0U)

Bit definitions for the I2C_OAR2 register.

- #define I2C_OAR2_ENDUAL_Msk (0x1UL << I2C_OAR2_ENDUAL_Pos)
- #define I2C_OAR2_ENDUAL I2C_OAR2_ENDUAL_Msk
- #define I2C_OAR2_ADD2_Pos (1U)
- #define I2C_OAR2_ADD2_Msk (0x1UL << I2C_OAR2_ADD2_Pos)
- #define I2C_OAR2_ADD2 I2C_OAR2_ADD2_Msk
- #define I2C_SR1_SB_Pos (0U)

Bit definitions for the I2C_SR1 register.

- #define I2C_SR1_SB_Msk (0x1UL << I2C_SR1_SB_Pos)
- #define I2C_SR1_SB I2C_SR1_SB_Msk
- #define I2C_SR1_ADDR_Pos (1U)
- #define I2C_SR1_ADDR_Msk (0x1UL << I2C_SR1_ADDR_Pos)
- #define I2C_SR1_ADDR I2C_SR1_ADDR_Msk
- #define I2C_SR1_BTF_Pos (2U)
- #define I2C_SR1_BTF_Msk (0x1UL << I2C_SR1_BTF_Pos)
- #define I2C_SR1_BTF I2C_SR1_BTF_Msk
- #define I2C_SR1_ADD10_Pos (3U)

- #define I2C_SR1_ADD10_Msk (0x1UL << I2C_SR1_ADD10_Pos)
- #define I2C_SR1_ADD10 I2C_SR1_ADD10_Msk
- #define I2C_SR1_STOPF_Pos (4U)
- #define I2C_SR1_STOPF_Msk (0x1UL << I2C_SR1_STOPF_Pos)
- #define I2C_SR1_STOPF I2C_SR1_STOPF_Msk
- #define I2C_SR1_RXNE_Pos (6U)
- #define I2C_SR1_RXNE_Msk (0x1UL << I2C_SR1_RXNE_Pos)
- #define I2C_SR1_RXNE I2C_SR1_RXNE_Msk
- #define I2C_SR1_TXE_Pos (7U)
- #define I2C_SR1_TXE_Msk (0x1UL << I2C_SR1_TXE_Pos)
- #define I2C_SR1_TXE I2C_SR1_TXE_Msk
- #define I2C_SR1_BERR_Pos (8U)
- #define I2C_SR1_BERR_Msk (0x1UL << I2C_SR1_BERR_Pos)
- #define I2C_SR1_BERR I2C_SR1_BERR_Msk
- #define I2C_SR1_ARLO_Pos (9U)
- #define I2C_SR1_ARLO_Msk (0x1UL << I2C_SR1_ARLO_Pos)
- #define I2C_SR1_ARLO I2C_SR1_ARLO_Msk
- #define I2C_SR1_AF_Pos (10U)
- #define I2C_SR1_AF_Msk (0x1UL << I2C_SR1_AF_Pos)
- #define I2C_SR1_AF I2C_SR1_AF_Msk
- #define I2C_SR1_OVR_Pos (11U)
- #define I2C_SR1_OVR_Msk (0x1UL << I2C_SR1_OVR_Pos)
- #define I2C_SR1_OVR I2C_SR1_OVR_Msk
- #define I2C_SR1_PECERR_Pos (12U)
- #define I2C_SR1_PECERR_Msk (0x1UL << I2C_SR1_PECERR_Pos)
- #define I2C_SR1_PECERR I2C_SR1_PECERR_Msk
- #define I2C_SR1_TIMEOUT_Pos (14U)
- #define I2C_SR1_TIMEOUT_Msk (0x1UL << I2C_SR1_TIMEOUT_Pos)
- #define I2C_SR1_TIMEOUT I2C_SR1_TIMEOUT_Msk
- #define I2C_SR1_SMBALERT_Pos (15U)
- #define I2C_SR1_SMBALERT_Msk (0x1UL << I2C_SR1_SMBALERT_Pos)
- #define I2C_SR1_SMBALERT I2C_SR1_SMBALERT_Msk
- #define I2C_SR2_MSL_Pos (0U)

Bit definitions for the I2C_SR2 register.

- #define I2C_SR2_MSL_Msk (0x1UL << I2C_SR2_MSL_Pos)
- #define I2C_SR2_MSL I2C_SR2_MSL_Msk
- #define I2C_SR2_BUSY_Pos (1U)
- #define I2C_SR2_BUSY_Msk (0x1UL << I2C_SR2_BUSY_Pos)
- #define I2C_SR2_BUSY I2C_SR2_BUSY_Msk
- #define I2C_SR2_TRA_Pos (2U)
- #define I2C_SR2_TRA_Msk (0x1UL << I2C_SR2_TRA_Pos)
- #define I2C_SR2_TRA I2C_SR2_TRA_Msk
- #define I2C_SR2_GENCALL_Pos (4U)
- #define I2C_SR2_GENCALL_Msk (0x1UL << I2C_SR2_GENCALL_Pos)
- #define I2C_SR2_GENCALL I2C_SR2_GENCALL_Msk
- #define I2C_SR2_SMBDEFAULT_Pos (5U)
- #define I2C_SR2_SMBDEFAULT_Msk (0x1UL << I2C_SR2_SMBDEFAULT_Pos)
- #define I2C_SR2_SMBDEFAULT I2C_SR2_SMBDEFAULT_Msk
- #define I2C_SR2_SMBHOST_Pos (6U)
- #define I2C_SR2_SMBHOST_Msk (0x1UL << I2C_SR2_SMBHOST_Pos)
- #define I2C_SR2_SMBHOST I2C_SR2_SMBHOST_Msk
- #define I2C_SR2_DUALF_Pos (7U)
- #define I2C_SR2_DUALF_Msk (0x1UL << I2C_SR2_DUALF_Pos)
- #define I2C_SR2_DUALF I2C_SR2_DUALF_Msk

- #define I2C_SR2_PEC_Pos (8U)
- #define I2C_SR2_PEC_Msk (0xFFUL << I2C_SR2_PEC_Pos)
- #define I2C_SR2_PEC I2C_SR2_PEC_Msk
- #define I2C_CCR_CCR_Pos (0U)
Bit definitions for the I2C_CCR register.
- #define I2C_CCR_CCR_Msk (0xFFFFUL << I2C_CCR_CCR_Pos)
- #define I2C_CCR_CCR_I2C_CCR_CCR_Msk
- #define I2C_CCR_DUTY_Pos (14U)
- #define I2C_CCR_DUTY_Msk (0x1UL << I2C_CCR_DUTY_Pos)
- #define I2C_CCR_DUTY_I2C_CCR_DUTY_Msk
- #define I2C_TRISE_TRISE_Pos (0U)
Bit definitions for the I2C_TRISE register.
- #define I2C_TRISE_TRISE_Msk (0x3FUL << I2C_TRISE_TRISE_Pos)
- #define I2C_TRISE_TRISE_I2C_TRISE_TRISE_Msk
- #define RCC_CR_HSION 0
- #define RCC_CR_HSIRDY 1
- #define RCC_CR_HSITRIM 3
- #define RCC_CR_HSICAL 8
- #define RCC_CR_HSEON 16
- #define RCC_CR_HSERDY 17
- #define RCC_CR_HSEBYP 18
- #define RCC_CR_CSSON 19
- #define RCC_CR_PLLON 24
- #define RCC_CR_PLLRDY 25
- #define RCC_CR_PLLI2SON 26
- #define RCC_CR_PLLI2SRDY 27
- #define RCC_CR_PLLSAION 28
- #define RCC_CR_PLLSAIRDY 29
- #define RCC_PLLCFG_R_PLLM 0
- #define RCC_PLLCFG_R_PLLN 6
- #define RCC_PLLCFG_R_PLLP 16
- #define RCC_PLLCFG_R_PLLSRC 22
- #define RCC_PLLCFG_R_PLLQ 24
- #define RCC_CFGR_SW 0
- #define RCC_CFGR_SWS 2
- #define RCC_CFGR_HPREF 4
- #define RCC_CFGR_PPREF1 10
- #define RCC_CFGR_PPREF2 13
- #define RCC_CFGR_RTCPPREF 16
- #define RCC_CFGR_MCO1 21
- #define RCC_CFGR_I2SSRC 23
- #define RCC_CFGR_MCO1PPREF 24
- #define RCC_CFGR_MCO2PPREF 27
- #define RCC_CFGR_MCO2 30
- #define RCC_CIR_LSIRDYF 0
- #define RCC_CIR_LSERDYF 1
- #define RCC_CIR_HSIRDYF 2
- #define RCC_CIR_HSERDYF 3
- #define RCC_CIR_PLLRDYF 4
- #define RCC_CIR_PLLI2SRDYF 5
- #define RCC_CIR_PLLSAIRDYF 6
- #define RCC_CIR_CSSF 7
- #define RCC_CIR_LSIRDYIE 8
- #define RCC_CIR_LSERDYIE 9

- #define RCC_CIR_HSIRDYIE 10
- #define RCC_CIR_HSERDYIE 11
- #define RCC_CIR_PLLRDYIE 12
- #define RCC_CIR_PLLI2SRDYIE 13
- #define RCC_CIR_PLLSAIRDYIE 14
- #define RCC_CIR_LSIRDYC 16
- #define RCC_CIR_LSERDYC 17
- #define RCC_CIR_HSIRDYC 18
- #define RCC_CIR_HSERDYC 19
- #define RCC_CIR_PLLRDYC 20
- #define RCC_CIR_PLLI2SRDYC 21
- #define RCC_CIR_PLLSAIRDYC 22
- #define RCC_AHB1RSTR_GPIOA 0
- #define RCC_AHB1RSTR_GPIOB 1
- #define RCC_AHB1RSTR_GPIOC 2
- #define RCC_AHB1RSTR_GPIOD 3
- #define RCC_AHB1RSTR_GPIOE 4
- #define RCC_AHB1RSTR_GPIOF 5
- #define RCC_AHB1RSTR_GPIOG 6
- #define RCC_AHB1RSTR_GPIOH 7
- #define RCC_AHB1RSTR_GPIOI 8
- #define RCC_AHB1RSTR_CRC 12
- #define RCC_AHB1RSTR_DMA1 21
- #define RCC_AHB1RSTR_DMA2 22
- #define RCC_AHB1RSTR_ETHMAC 25
- #define RCC_AHB1RSTR_OTGHS 29
- #define RCC_AHB1RSTR_OTGHSULPI 30
- #define RCC_AHB2RSTR_DCMI 0
- #define RCC_AHB2RSTR_CRYP 4
- #define RCC_AHB2RSTR_HASH 5
- #define RCC_AHB2RSTR_RNG 6
- #define RCC_AHB2RSTR_OTGFS 7
- #define RCC_AHB3RSTR_FSMC 0
- #define RCC_APB1RSTR_TIM2 0
- #define RCC_APB1RSTR_TIM3 1
- #define RCC_APB1RSTR_TIM4 2
- #define RCC_APB1RSTR_TIM5 3
- #define RCC_APB1RSTR_TIM6 4
- #define RCC_APB1RSTR_TIM7 5
- #define RCC_APB1RSTR_TIM12 6
- #define RCC_APB1RSTR_TIM13 7
- #define RCC_APB1RSTR_TIM14 8
- #define RCC_APB1RSTR_WWDG 11
- #define RCC_APB1RSTR_SPI2 14
- #define RCC_APB1RSTR_SPI3 15
- #define RCC_APB1RSTR_USART2 17
- #define RCC_APB1RSTR_USART3 18
- #define RCC_APB1RSTR_UART4 19
- #define RCC_APB1RSTR_UART5 20
- #define RCC_APB1RSTR_I2C1 21
- #define RCC_APB1RSTR_I2C2 22
- #define RCC_APB1RSTR_I2C3 23
- #define RCC_APB1RSTR_CAN1 25
- #define RCC_APB1RSTR_CAN2 26
- #define RCC_APB1RSTR_PWR 28

- #define RCC_APB1RSTR_DAC 29
- #define RCC_APB1RSTR_UART7 30
- #define RCC_APB1RSTR_UART8 31
- #define RCC_APB2RSTR_TIM1 0
- #define RCC_APB2RSTR_TIM8 1
- #define RCC_APB2RSTR_USART1 4
- #define RCC_APB2RSTR_USART6 5
- #define RCC_APB2RSTR_ADC 8
- #define RCC_APB2RSTR_SDIO 11
- #define RCC_APB2RSTR_SPI1 12
- #define RCC_APB2RSTR_SYSCFG 14
- #define RCC_APB2RSTR_TIM9 16
- #define RCC_APB2RSTR_TIM10 17
- #define RCC_APB2RSTR_TIM11 18
- #define RCC_AHB1LPENR_GPIOALPEN 0
- #define RCC_AHB1LPENR_GPIOBLPEN 1
- #define RCC_AHB1LPENR_GPIOCLPEN 2
- #define RCC_AHB1LPENR_GPIODLPEN 3
- #define RCC_AHB1LPENR_GPIOELPEN 4
- #define RCC_AHB1LPENR_GPIOFLPEN 5
- #define RCC_AHB1LPENR_GPIOGLPEN 6
- #define RCC_AHB1LPENR_GPIOHLPEN 7
- #define RCC_AHB1LPENR_GPIOILPEN 8
- #define RCC_AHB1LPENR_CRCEN 12
- #define RCC_AHB1LPENR_DMA1LPEN 21
- #define RCC_AHB1LPENR_DMA2LPEN 22
- #define RCC_AHB1LPENR_ETHMACLPEN 25
- #define RCC_AHB1LPENR_ETHMACTXLPEN 26
- #define RCC_AHB1LPENR_ETHMACRXLPEN 27
- #define RCC_AHB1LPENR_ETHMACPTPLPEN 28
- #define RCC_AHB1LPENR_OTGHSLPEN 29
- #define RCC_AHB1LPENR_OTGHSHULPI 30
- #define RCC_AHB2LPENR_DCMILPEN 0
- #define RCC_AHB2LPENR_CRYPLPEN 4
- #define RCC_AHB2LPENR_HASHLPEN 5
- #define RCC_AHB2LPENR_RNGLPEN 6
- #define RCC_AHB2LPENR_OTGFSLPEN 7
- #define RCC_AHB3LPENR_FSMCLPEN 0
- #define RCC_APB1LPENR_TIM2LPEN 0
- #define RCC_APB1LPENR_TIM3LPEN 1
- #define RCC_APB1LPENR_TIM4LPEN 2
- #define RCC_APB1LPENR_TIM5LPEN 3
- #define RCC_APB1LPENR_TIM6LPEN 4
- #define RCC_APB1LPENR_TIM7LPEN 5
- #define RCC_APB1LPENR_TIM12LPEN 6
- #define RCC_APB1LPENR_TIM13LPEN 7
- #define RCC_APB1LPENR_TIM14LPEN 8
- #define RCC_APB1LPENR_WWDGLPEN 11
- #define RCC_APB1LPENR_SPI2LPEN 14
- #define RCC_APB1LPENR_SPI3LPEN 15
- #define RCC_APB1LPENR_USART2LPEN 17
- #define RCC_APB1LPENR_USART3LPEN 18
- #define RCC_APB1LPENR_UART4LPEN 19
- #define RCC_APB1LPENR_UART5LPEN 20
- #define RCC_APB1LPENR_I2C1LPEN 21

- #define RCC_APB1LPENR_I2C2LPEN 22
- #define RCC_APB1LPENR_I2C3LPEN 23
- #define RCC_APB1LPENR_CAN1LPEN 25
- #define RCC_APB1LPENR_CAN2LPEN 26
- #define RCC_APB1LPENR_PWRLPEN 28
- #define RCC_APB1LPENR_DACLPEN 29
- #define RCC_APB1LPENR_UART7LPEN 30
- #define RCC_APB1LPENR_UART8LPEN 31
- #define RCC_APB2LPENR_TIM1LPEN 0
- #define RCC_APB2LPENR_TIM8LPEN 1
- #define RCC_APB2LPENR_USART1LPEN 4
- #define RCC_APB2LPENR_USART6LPEN 5
- #define RCC_APB2LPENR_ADCLPEN 8
- #define RCC_APB2LPENR_SDIOLPEN 11
- #define RCC_APB2LPENR_SPI1LPEN 12
- #define RCC_APB2LPENR_SYSCFGLPEN 14
- #define RCC_APB2LPENR_TIM9LPEN 16
- #define RCC_APB2LPENR_TIM10LPEN 17
- #define RCC_APB2LPENR_TIM11LPEN 18
- #define RCC_BDCR_LSEON 0
- #define RCC_BDCR_LSERDY 1
- #define RCC_BDCR_LSEBYP 2
- #define RCC_BDCR_RTCSEL 8
- #define RCC_BDCR_RTCEN 15
- #define RCC_BDCR_BDRST 16
- #define RCC_CSR_LSION 0
- #define RCC_CSR_LSIRDY 1
- #define RCC_CSR_RMVF 24
- #define RCC_CSR_OBLRSTF 25
- #define RCC_CSR_PINRSTF 26
- #define RCC_CSR_PORRSTF 27
- #define RCC_CSR_SFTRSTF 28
- #define RCC_CSR_IWDGRSTF 29
- #define RCC_CSR_WWDGRSTF 30
- #define RCC_CSR_LPWRRTSF 31
- #define RCC_SSCGR_MODPER 0
- #define RCC_SSCGR_INCSTEP 13
- #define RCC_SSCGR_SPREADSEL 15
- #define RCC_SSCGR_SSCGEN 31
- #define RCC_PLLI2SCFGR_PLLI2SN 6
- #define RCC_PLLI2SCFGR_PLLI2SR 28
- #define GPIO_BASEADDR_TO_CODE(x)

Macro to convert GPIO base address to port code.

7.16.1 Detailed Description

Header file containing all the necessary information about the STM32F401xx MCU.

Author

Mohamed Ali Haoufa

This file provides definitions, memory maps, and register structures for the STM32F401xx series of microcontrollers. It includes base addresses for various memory regions and peripheral registers.

Version

1.0

Date

2024-09-21

Attention

This file is designed specifically for the STM32F401xx series microcontrollers. Ensure that all references and addresses are compliant with the specific microcontroller variant being used.

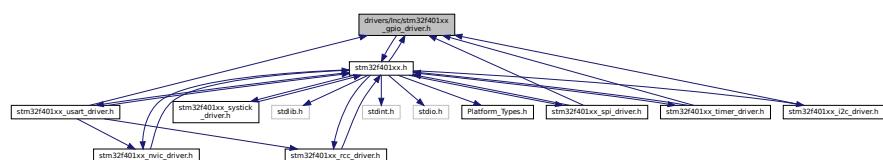
Copyright

(c) 2024

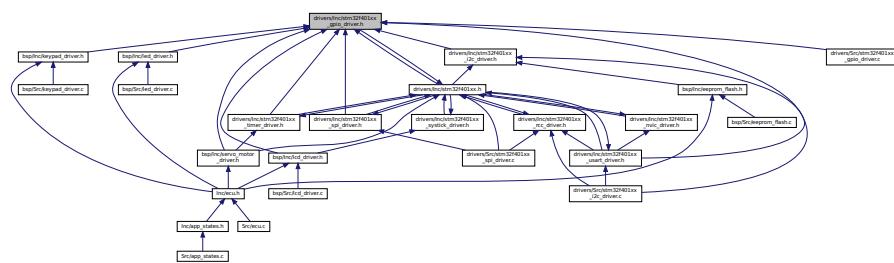
7.17 drivers/Inc/stm32f401xx gpio driver.h File Reference

Header file for GPIO driver for STM32F401xx microcontroller.

```
#include "stm32f401xx.h"
Include dependency graph for stm32f401xx_gpio_driver.h:
```



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



Classes

- struct GPIO_PinConfig t

Configuration structure for GPIO pins.

Macros

- #define **GPIO_PIN_0** 0
- #define **GPIO_PIN_1** 1
- #define **GPIO_PIN_2** 2
- #define **GPIO_PIN_3** 3
- #define **GPIO_PIN_4** 4
- #define **GPIO_PIN_5** 5
- #define **GPIO_PIN_6** 6
- #define **GPIO_PIN_7** 7
- #define **GPIO_PIN_8** 8
- #define **GPIO_PIN_9** 9
- #define **GPIO_PIN_10** 10
- #define **GPIO_PIN_11** 11
- #define **GPIO_PIN_12** 12
- #define **GPIO_PIN_13** 13
- #define **GPIO_PIN_14** 14
- #define **GPIO_PIN_15** 15
- #define **GPIO_PIN_ALL** ((**uint16**)0xFFFF)
- #define **GPIO_MODE_INPUT_FLO** 0x00000001U
- #define **GPIO_MODE_INPUT_PU** 0x00000002U
- #define **GPIO_MODE_INPUT_PD** 0x00000003U
- #define **GPIO_MODE_OUTPUT_PP** 0x00000004U
- #define **GPIO_MODE_OUTPUT_OD** 0x00000005U
- #define **GPIO_MODE_OUTPUT_AF_PP** 0x00000006U
- #define **GPIO_MODE_OUTPUT_AF_OD** 0x00000007U
- #define **GPIO_MODE_AF_INPUT** 0x00000008U
- #define **GPIO_SPEED_10M** 0x00000001U
- #define **GPIO_SPEED_2M** 0x00000002U
- #define **GPIO_SPEED_50M** 0x00000003U
- #define **GPIO_PIN_SET** 1
- #define **GPIO_PIN_RESET** 0
- #define **GPIO_RETURN_LOCK_OK** 1
- #define **GPIO_RETURN_LOCK_ERROR** 0
- #define **GPIO_MODE_IN** 0
- #define **GPIO_MODE_OUT** 1
- #define **GPIO_MODE_ALTFN** 2
- #define **GPIO_MODE_ANALOG** 3
- #define **GPIO_MODE_IT_FT** 4
- #define **GPIO_MODE_IT_RT** 5
- #define **GPIO_MODE_IT_RFT** 6
- #define **GPIO_SPEED_LOW** 0
- #define **GPIO_SPEED_MEDIUM** 1
- #define **GPIO_SPEED_FAST** 2
- #define **GPIO_SPEED_HIGH** 3
- #define **GPIO_NO_PUPD** 0
- #define **GPIO_PIN_PU** 1
- #define **GPIO_PIN_PD** 2
- #define **GPIO_OP_TYPE_PP** 0
- #define **GPIO_OP_TYPE_OD** 1

Functions

- void `MCAL_GPIO_Init (GPIO_TypeDef *GPIOx, GPIO_PinConfig_t *PinConfig)`
Initializes the specified GPIO pin according to the provided configuration.
- void `MCAL_GPIO_DeInit (GPIO_TypeDef *GPIOx)`
Resets the specified GPIO port.
- `uint8 MCAL_GPIO_ReadPin (GPIO_TypeDef *GPIOx, uint16 PinNumber)`
Reads the value of a specified GPIO pin.
- `uint16 MCAL_GPIO_ReadPort (GPIO_TypeDef *GPIOx)`
Reads the value of the entire GPIO port.
- void `MCAL_GPIO_WritePin (GPIO_TypeDef *GPIOx, uint16 PinNumber, uint8 Value)`
Writes a value to a specified GPIO pin.
- void `MCAL_GPIO_WritePort (GPIO_TypeDef *GPIOx, uint16 Value)`
Writes a value to the entire GPIO port.
- void `MCAL_GPIO_TogglePin (GPIO_TypeDef *GPIOx, uint16 PinNumber)`
Toggles the state of a specified GPIO pin.
- void `MCAL_GPIO_TogglePort (GPIO_TypeDef *GPIOx)`
Toggles the value of a specific GPIO Port.
- `uint8 MCAL_GPIO_LockPin (GPIO_TypeDef *GPIOx, uint16 PinNumber)`
Locks the configuration of a specific GPIO pin to prevent further changes.

7.17.1 Detailed Description

Header file for GPIO driver for STM32F401xx microcontroller.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file provides the definitions, macros, and function prototypes for configuring and controlling GPIO pins on the STM32F401xx microcontroller.

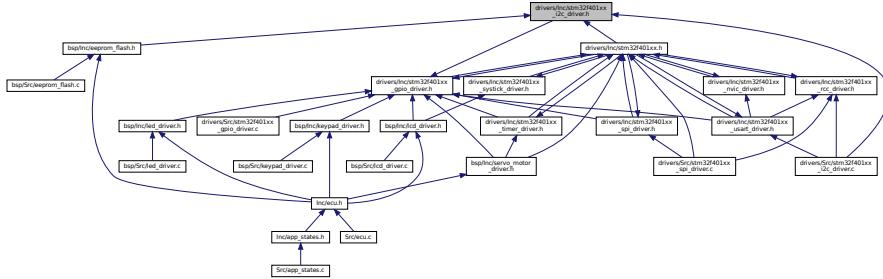
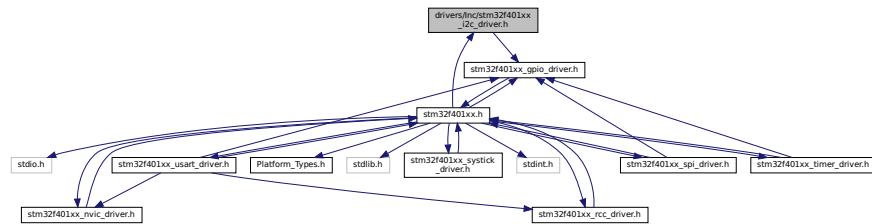
Copyright

Copyright (c) 2024 Mohamed Ali Haoufa

7.18 drivers/Inc/stm32f401xx_i2c_driver.h File Reference

Header file for STM32F401xx I2C driver.

```
#include "stm32f401xx_gpio_driver.h"
Include dependency graph for stm32f401xx_i2c_driver.h:
```



Classes

- struct [S_I2C_Slave_address](#)
Structure for I2C slave address configuration.
- struct [S_I2C_Config_t](#)
Structure for I2C configuration.

Macros

- #define [I2C_SCK_SM_50K](#) (0x500000UL)
- #define [I2C_SCK_SM_100K](#) (1000000UL)
- #define [I2C_SCK_FM_200K](#) (2000000UL)
- #define [I2C_SCK_FM_400K](#) (4000000UL)
- #define [I2C_StretchMode_enabled](#) 0x00000000U
- #define [I2C_StretchMode_disabled](#) I2C_CR1_NOSTRETCH
- #define [I2C_Mode_I2C](#) 0
- #define [I2C_Mode_SMBus](#) I2C_CR1_SMBUS
- #define [I2C_Addressing_Slave_7bits](#) 0
- #define [I2C_Addressing_Slave_10bits](#) (uint16_t)(1<<15)
- #define [I2C_Ack_Control_Enable](#) I2C_CR1_ACK
- #define [I2C_Ack_Control_Disable](#) 0
- #define [I2C_ENGC_Enable](#) I2C_CR1_ENGC
- #define [I2C_ENGC_Disable](#) 0x00

Enumerations

- enum `Slave_State` {
 `I2C_EV_Stop` , `I2C_Error_AF` , `I2C_Ev_Address_Matched` , `I2C_Ev_Data_Req` , `I2C_Ev_Data_RCV` }

Enumeration for I2C slave states.
- enum `StopCondition` { `WithStop` , `WithoutStop` }

Enumeration for I2C stop condition.
- enum `Repeated_Start` { `Start` , `Repeated` }

Enumeration for I2C repeated start condition.
- enum `Functional_State` { `DISABLE` , `ENABLE` }

Enumeration for functional state.
- enum `FlagStatus` { `RESET` = 0 , `SET` = !`RESET` , `Reset` , `Set` }

Enumeration for flag status.
- enum `Status` {
 `I2C_Flag_Busy` , `EV5` , `EV6` , `EV7` , `EV8` , `EV8_1` , `Master_Byte_Transmitting` = (`uint32_t`)(0x00070080) }

Enumeration for I2C status flags.
- enum `I2C_Direction` { `I2C_Direction_Transmitter` , `I2C_Direction_Receiver` }

Enumeration for I2C direction.

Functions

- void `MCAL_I2C_Init` (`I2C_TypeDef` *`I2Cx`, `S_I2C_Config_t` *`I2C_Init_Struct`)

Initializes the I2C peripheral.
- void `MCAL_I2C_Deinit` (`I2C_TypeDef` *`I2Cx`)

Deinitializes the I2C peripheral.
- void `MCAL_I2C_Set_GPIO` (`I2C_TypeDef` *`I2Cx`)

Sets the GPIO configuration for I2C.
- void `MCAL_I2C_Master_Tx` (`I2C_TypeDef` *`I2Cx`, `uint16_t` `SlaveAdd`, `uint8_t` *`dataout`, `uint32_t` `datalen`, `StopCondition` `Stop`, `Repeated_Start` `start`)

Transmits data from the master to the slave.
- void `MCAL_I2C_Master_Rx` (`I2C_TypeDef` *`I2Cx`, `uint16_t` `SlaveAdd`, `uint8_t` *`dataout`, `uint32_t` `datalen`, `StopCondition` `Stop`, `Repeated_Start` `start`)

Receives data from the master.
- void `MCAL_I2C_Slave_SendData` (`I2C_TypeDef` *`I2Cx`, `uint8_t` `data`)

Sends data from the slave.
- `uint8_t MCAL_I2C_SlaveReceiveData` (`I2C_TypeDef` *`I2Cx`)

Receives data in slave mode.
- void `I2C_Generate_Start` (`I2C_TypeDef` *`I2Cx`, `Functional_State` `state`, `Repeated_Start` `start`)

Generates a start condition on the I2C bus.
- `FlagStatus I2C_Get_FlagStatus` (`I2C_TypeDef` *`I2Cx`, `Status` `flag`)

Gets the status of a specific flag.
- void `I2C_SendAddress` (`I2C_TypeDef` *`I2Cx`, `uint16_t` `SlaveAddress`, `I2C_Direction` `I2C_Direction`)

Sends an address to the I2C bus.
- void `I2C_Generate_Stop` (`I2C_TypeDef` *`I2Cx`, `Functional_State` `NewState`)

Generates a stop condition on the I2C bus.
- void `I2C_ACKnowledgeConfig` (`I2C_TypeDef` *`I2Cx`, `Functional_State` `NewState`)

Configures the I2C acknowledge feature.

7.18.1 Detailed Description

Header file for STM32F401xx I2C driver.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file provides the interface for initializing and controlling the I2C peripheral on the STM32F401xx microcontroller. It includes definitions for I2C configurations, enumerations for I2C status and control, and function prototypes for I2C operations.

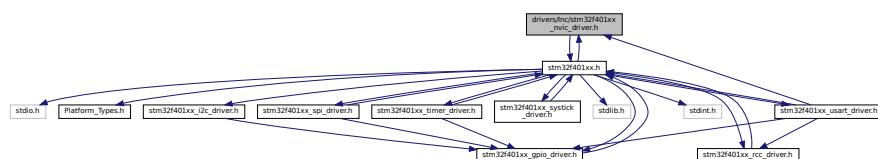
Copyright

Copyright (c) 2023 Fam Ayman. All rights reserved.

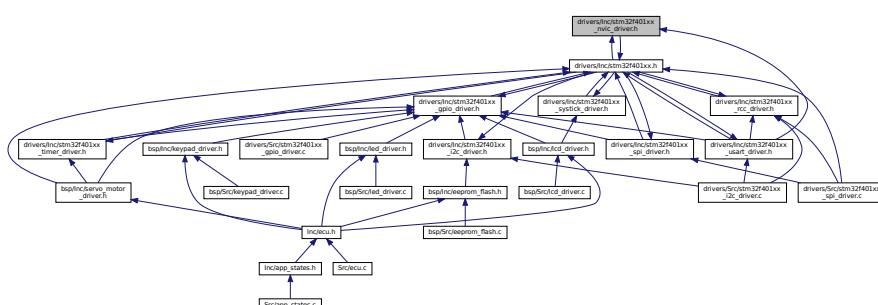
7.19 drivers/Inc/stm32f401xx_nvic_driver.h File Reference

Header file for NVIC (Nested Vectored Interrupt Controller) driver for STM32F401xx microcontroller.

```
#include "stm32f401xx.h"
Include dependency graph for stm32f401xx_nvic_driver.h:
```



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



Macros

- `#define NVIC_ISER0 (*(volatile uint32_t *)(<NVIC_BASE> + 0x0))`
Interrupt Set Enable Register 0 (ISER0) Used to enable interrupts 0 to 31. Write a 1 to the corresponding bit position to enable an interrupt.
- `#define NVIC_ISER1 (*(volatile uint32_t *)(<NVIC_BASE> + 0x4))`
Interrupt Set Enable Register 1 (ISER1) Used to enable interrupts 32 to 63. Write a 1 to the corresponding bit position to enable an interrupt.
- `#define NVIC_ISER2 (*(volatile uint32_t *)(<NVIC_BASE> + 0x8))`
Interrupt Set Enable Register 2 (ISER2) Used to enable interrupts 64 to 95. Write a 1 to the corresponding bit position to enable an interrupt.
- `#define NVIC_ICER0 (*(volatile uint32_t *)(<NVIC_BASE> + 0x80))`
Interrupt Clear Enable Register 0 (ICER0) Used to disable interrupts 0 to 31. Write a 1 to the corresponding bit position to disable an interrupt.
- `#define NVIC_ICER1 (*(volatile uint32_t *)(<NVIC_BASE> + 0x84))`
Interrupt Clear Enable Register 1 (ICER1) Used to disable interrupts 32 to 63. Write a 1 to the corresponding bit position to disable an interrupt.
- `#define NVIC_ICER2 (*(volatile uint32_t *)(<NVIC_BASE> + 0x88))`
Interrupt Clear Enable Register 2 (ICER2) Used to disable interrupts 64 to 95. Write a 1 to the corresponding bit position to disable an interrupt.
- `#define SCB_VECTKEY 0x05FA0000UL`
Vector key for setting priority grouping in NVIC.
- `#define SCB_VECTKEY_MASK 0xFFFF0000UL`
Mask for vector key.
- `#define NVIC_PRIGROUP_SET_MASK 0x700UL`
Mask for setting NVIC priority grouping.
- `#define NVIC_PRIGROUP_CLEAR_MASK 0xFFFFF8FFUL`
Mask for clearing NVIC priority grouping.
- `#define NVIC_INTERRUPT_ACTIVE 1UL`
Status indicating an active interrupt.
- `#define NVIC_INTERRUPT_INACTIVE 0UL`
Status indicating an inactive interrupt.
- `#define NVIC_PRIO_16GRP_0SUBGRP 0x300U`
Priority grouping configuration for 16 groups and 0 subgroups.
- `#define NVIC_PRIO_8GRP_2SUBGRP 0x400U`
Priority grouping configuration for 8 groups and 2 subgroups.
- `#define NVIC_PRIO_4GRP_4SUBGRP 0x500U`
Priority grouping configuration for 4 groups and 4 subgroups.
- `#define NVIC_PRIO_2GRP_8SUBGRP 0x600U`
Priority grouping configuration for 2 groups and 8 subgroups.
- `#define NVIC_PRIO_0GRP_8SUBGRP 0x700U`
Priority grouping configuration for 0 groups and 8 subgroups.
- `#define NVIC_PRIO_0000 0x00U`
Priority level 0.
- `#define NVIC_PRIO_0001 0x10U`
Priority level 1.
- `#define NVIC_PRIO_0010 0x20U`
Priority level 2.
- `#define NVIC_PRIO_0011 0x30U`
Priority level 3.
- `#define NVIC_PRIO_0100 0x40U`
Priority level 4.

- #define **NVIC_PRIO_0101** 0x50U
Priority level 5.
- #define **NVIC_PRIO_0110** 0x60U
Priority level 6.
- #define **NVIC_PRIO_0111** 0x70U
Priority level 7.
- #define **NVIC_PRIO_1000** 0x80U
Priority level 8.
- #define **NVIC_PRIO_1001** 0x90U
Priority level 9.
- #define **NVIC_PRIO_1010** 0xA0U
Priority level 10.
- #define **NVIC_PRIO_1011** 0xB0U
Priority level 11.
- #define **NVIC_PRIO_1100** 0xC0U
Priority level 12.
- #define **NVIC_PRIO_1101** 0xD0U
Priority level 13.
- #define **NVIC_PRIO_1110** 0xE0U
Priority level 14.
- #define **NVIC_PRIO_1111** 0xF0U
Priority level 15.
- #define **EXTI0_IRQ** 6
External Interrupt Request 0.
- #define **EXTI1_IRQ** 7
External Interrupt Request 1.
- #define **EXTI2_IRQ** 8
External Interrupt Request 2.
- #define **EXTI3_IRQ** 9
External Interrupt Request 3.
- #define **EXTI4_IRQ** 10
External Interrupt Request 4.
- #define **EXTI5_IRQ** 23
External Interrupt Request 5.
- #define **EXTI6_IRQ** 23
External Interrupt Request 6.
- #define **EXTI7_IRQ** 23
External Interrupt Request 7.
- #define **EXTI8_IRQ** 23
External Interrupt Request 8.
- #define **EXTI9_IRQ** 23
External Interrupt Request 9.
- #define **EXTI10_IRQ** 40
External Interrupt Request 10.
- #define **EXTI11_IRQ** 40
External Interrupt Request 11.
- #define **EXTI12_IRQ** 40
External Interrupt Request 12.
- #define **EXTI13_IRQ** 40
External Interrupt Request 13.
- #define **EXTI14_IRQ** 40

- External Interrupt Request 14.*
- #define **EXTI15_IRQ** 40
- External Interrupt Request 15.*
- #define **USART1_IRQ** 37
- USART1 Interrupt Request.*
- #define **USART2_IRQ** 38
- USART2 Interrupt Request.*
- #define **USART6_IRQ** 71
- USART6 Interrupt Request.*
- #define **SPI1_IRQ** 35
- SPI1 Interrupt Request.*
- #define **SPI2_IRQ** 36
- SPI2 Interrupt Request.*
- #define **I2C1_EV_IRQ** 31
- I2C1 Event Interrupt Request.*
- #define **I2C1_ER_IRQ** 32
- I2C1 Error Interrupt Request.*
- #define **I2C2_EV_IRQ** 33
- I2C2 Event Interrupt Request.*
- #define **I2C2_ER_IRQ** 34
- I2C2 Error Interrupt Request.*
- #define **NVIC IRQ6_EXTI0_Enable** (NVIC_ISERO |= (1<<6))
 - #define **NVIC IRQ7_EXTI1_Enable** (NVIC_ISERO |= (1<<7))
 - #define **NVIC IRQ8_EXTI2_Enable** (NVIC_ISERO |= (1<<8))
 - #define **NVIC IRQ9_EXTI3_Enable** (NVIC_ISERO |= (1<<9))
 - #define **NVIC IRQ10_EXTI4_Enable** (NVIC_ISERO |= (1<<10))
 - #define **NVIC IRQ23_EXTI5_9_Enable** (NVIC_ISERO |= (1<<23))
 - #define **NVIC IRQ40_EXTI10_15_Enable** (NVIC_ICER1 |= (1<<8))
 - #define **NVIC IRQ6_EXTI0_Disable** (NVIC_ICERO |= (1<<6))
 - #define **NVIC IRQ7_EXTI1_Disable** (NVIC_ICERO |= (1<<7))
 - #define **NVIC IRQ8_EXTI2_Disable** (NVIC_ICERO |= (1<<8))
 - #define **NVIC IRQ9_EXTI3_Disable** (NVIC_ICERO |= (1<<9))
 - #define **NVIC IRQ10_EXTI4_Disable** (NVIC_ICERO |= (1<<10))
 - #define **NVIC IRQ23_EXTI5_9_Disable** (NVIC_ICERO |= (1<<23))
 - #define **NVIC IRQ40_EXTI10_15_Disable** (NVIC_ICER1 |= (1<<8))
 - #define **NVIC IRQ37_USART1_Enable** (NVIC_ISER1 |= 1<<(USART1_IRQ - 32))
 - #define **NVIC IRQ38_USART2_Enable** (NVIC_ISER1 |= 1<<(USART2_IRQ - 32))
 - #define **NVIC IRQ39_USART3_Enable** (NVIC_ISER1 |= 1<<(USART3_IRQ - 32))
 - #define **NVIC IRQ37_USART1_Disable** (NVIC_ICER1 |= 1<<(USART1_IRQ - 32))
 - #define **NVIC IRQ38_USART2_Disable** (NVIC_ICER1 |= 1<<(USART2_IRQ - 32))
 - #define **NVIC IRQ39_USART3_Disable** (NVIC_ICER1 |= 1<<(USART3_IRQ - 32))
 - #define **NVIC IRQ35_SPI1_Enable** (NVIC_ISER1 |= 1<<(SPI1_IRQ - 32))
 - #define **NVIC IRQ36_SPI2_Enable** (NVIC_ISER1 |= 1<<(SPI2_IRQ - 32))
 - #define **NVIC IRQ35_SPI1_Disable** (NVIC_ICER1 &= ~(1<<(SPI1_IRQ - 32)))
 - #define **NVIC IRQ36_SPI2_Disable** (NVIC_ICER1 &= ~(1<<(SPI2_IRQ - 32)))
 - #define **NVIC IRQ31_I2C1_EV_Enable** (NVIC_ISER0 |= 1<<(I2C1_EV_IRQ))
 - #define **NVIC IRQ32_I2C1_ER_Enable** (NVIC_ISER1 |= 1<<(I2C1_ER_IRQ - 32))
 - #define **NVIC IRQ33_I2C2_EV_Enable** (NVIC_ISER1 |= 1<<(I2C2_EV_IRQ - 32))
 - #define **NVIC IRQ34_I2C2_ER_Enable** (NVIC_ISER1 |= 1<<(I2C2_ER_IRQ - 32))
 - #define **NVIC IRQ31_I2C1_EV_Disable** (NVIC_ICER0 &= ~(1<<(I2C1_EV_IRQ)))
 - #define **NVIC IRQ32_I2C1_ER_Disable** (NVIC_ICER1 &= ~(1<<(I2C1_ER_IRQ - 32)))
 - #define **NVIC IRQ33_I2C2_EV_Disable** (NVIC_ICER1 &= ~(1<<(I2C2_EV_IRQ - 32)))
 - #define **NVIC IRQ34_I2C2_ER_Disable** (NVIC_ICER1 &= ~(1<<(I2C2_ER_IRQ - 32)))

Functions

- void `MCAL_NVIC_SetPriorityGrouping (uint32 priority_grouping)`
Set the priority grouping for the NVIC.
- `uint32 MCAL_NVIC_GetPriorityGrouping (void)`
Get the current priority grouping configuration of the NVIC.
- void `MCAL_NVIC_EnableIRQ (uint8 IRQn)`
Enable a specified IRQ.
- void `MCAL_NVIC_DisableIRQ (uint8 IRQn)`
Disable a specified IRQ.
- `uint8 MCAL_NVIC_GetPendingIRQ (uint8 IRQn)`
Get the pending status of a specified IRQ.
- void `MCAL_NVIC_SetPendingIRQ (uint8 IRQn)`
Set the pending status of a specified IRQ.
- void `MCAL_NVIC_ClearPendingIRQ (uint8 IRQn)`
Clear the pending status of a specified IRQ.
- `uint8 MCAL_NVIC_GetActive (uint8 IRQn)`
Get the active status of a specified IRQ.
- void `MCAL_NVIC_SetPriority (uint8 IRQn, uint8 priority)`
Set the priority of a specified IRQ.
- `uint8 MCAL_NVIC_GetPriority (uint8 IRQn)`
Retrieves the priority level of a specified IRQn.
- void `MCAL_NVIC_SystemReset (void)`
Performs a system reset.

7.19.1 Detailed Description

Header file for NVIC (Nested Vectored Interrupt Controller) driver for STM32F401xx microcontroller.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This header file provides definitions, macros, and function prototypes for interacting with the NVIC of the STM32F401xx microcontroller. It includes functions for configuring interrupt priority, enabling/disabling interrupts, and managing interrupt status.

Copyright

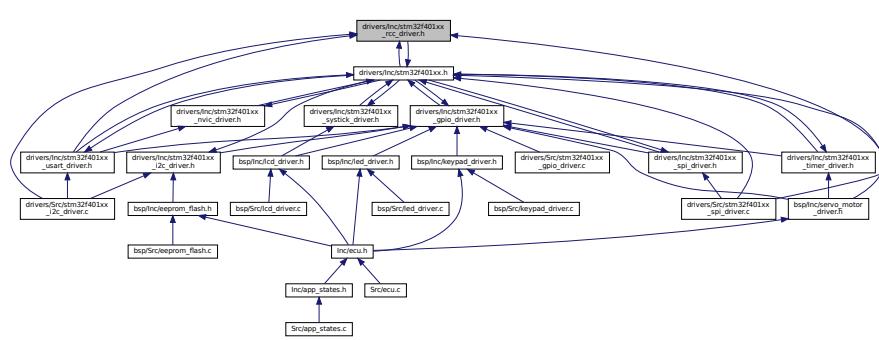
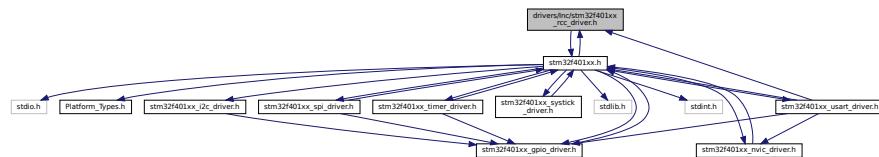
Copyright (c) 2024

7.20 drivers/Inc/stm32f401xx_rcc_driver.h File Reference

Header file for RCC (Reset and Clock Control) driver for STM32F401xx MCU.

```
#include "stm32f401xx.h"
```

Include dependency graph for stm32f401xx_rcc_driver.h:



Macros

- #define **RCC_PLLCFGR_PLLSRC_HSE** ((uint32_t)0x00400000)
- #define **HSI_VALUE** 16000000U
- #define **HSE_VALUE** 8000000U
- #define **RCC_GPIOA** ((uint8_t)0x00)
- #define **RCC_GPIOB** ((uint8_t)0x01)
- #define **RCC_GPIOC** ((uint8_t)0x02)
- #define **RCC_GPIOD** ((uint8_t)0x03)
- #define **RCC_GPIOE** ((uint8_t)0x04)
- #define **RCC_USART1** ((uint8_t)0x08)
- #define **RCC_USART2** ((uint8_t)0x09)
- #define **RCC_USART6** ((uint8_t)0xA)
- #define **RCC_SPI1** ((uint8_t)0xB)
- #define **RCC_SPI2** ((uint8_t)0xC)
- #define **RCC_I2C1** ((uint8_t)0xD)
- #define **RCC_I2C2** ((uint8_t)0xE)
- #define **RCC_CRC** ((uint8_t)0xF)
- #define **RCC_TIM2** ((uint8_t)0x10)
- #define **RCC_SYSCFG** ((uint8_t)0x11)
- #define **HSI_RC_CLK** 8000000UL
- #define **HSE_CLK** 8000000UL
- #define **RCC_SELECT_HSI** ((uint8_t)0x00)
- #define **RCC_SELECT_HSE** ((uint8_t)0x01)
- #define **RCC_SELECT_PLL** ((uint8_t)0x02)

Functions

- void [MCAL_RCC_Select_Clock](#) (uint8_t clock)
Select the clock source for the MCU.
- void [MCAL_RCC_Enable_Peripheral](#) (uint8_t peripheral)
Enable the clock for a specific peripheral.
- void [MCAL_RCC_Reset_Peripheral](#) (uint8_t peripheral)
Reset a specific peripheral.
- uint32_t [MCAL_RCC_GetSYS_CLKFreq](#) (void)
Get the frequency of the system clock.
- uint32_t [MCAL_RCC_GetHCLKFreq](#) (void)
Get the frequency of the AHB bus clock.
- uint32_t [MCAL_RCC_GetPCLK1Freq](#) (void)
Get the frequency of the APB1 bus clock.
- uint32_t [MCAL_RCC_GetPCLK2Freq](#) (void)
Get the frequency of the APB2 bus clock.
- uint32_t [RCC_GetPLLOutputClock](#) (void)
Get the frequency of the PLL (Phase-Locked Loop) output clock.

7.20.1 Detailed Description

Header file for RCC (Reset and Clock Control) driver for STM32F401xx MCU.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This header file defines the macros, constants, and function prototypes for managing the Reset and Clock Control (RCC) peripheral of the STM32F401xx microcontroller. It includes clock source selection, peripheral enablement, and clock frequency retrieval.

Copyright

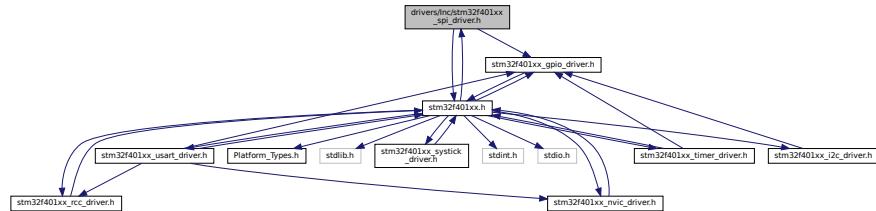
Copyright (c) 2024 Mohamed Ali Haoufa

7.21 drivers/Inc/stm32f401xx_spi_driver.h File Reference

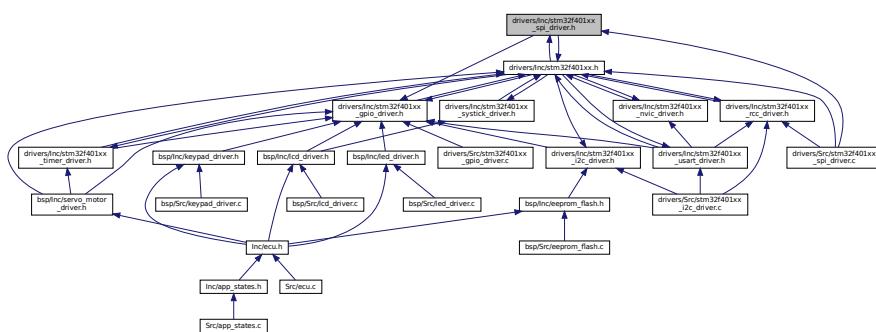
This file contains definitions and function prototypes for the STM32F401xx SPI driver.

```
#include "stm32f401xx_gpio_driver.h"
#include "stm32f401xx.h"
```

Include dependency graph for stm32f401xx_spi_driver.h:



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



Classes

- struct [S_IRQ_SRC](#)
Structure to identify the source of SPI interrupts.
- struct [S_SPI_Config_t](#)
Configuration structure for SPI.

Macros

- #define [SPI_Mode_Master](#) (0x1<<2)
- #define [SPI_Mode_Slave](#) (0x00000000U)
- #define [SPI_Direction_2lines](#) (0x00000000U)
- #define [SPI_Direction_2lines_RX_Only](#) (0x1<<10)
- #define [SPI_Direction_1line_Receive_only](#) (0x1<<15)
- #define [SPI_Direction_1line_Transmit_only](#) ((0x1<<15)|(0x1<<14))
- #define [SPI_Payload_Length_8bit](#) (0x00000000U)
- #define [SPI_Payload_Length_16bit](#) (0x1<<11)
- #define [SPI_Frame_Format_MSB](#) (0x00000000U)

- #define SPI_Frame_Format_LSB (0x1<<7)
- #define SPI_Clock_Polarity_Low_Idle (0x00000000U)
- #define SPI_Clock_Polarity_High_Idle (0x1<<1)
- #define SPI_Clock_Phase_Leading (0x00000000U)
- #define SPI_Clock_Phase_Trailing (0x1)
- #define SPI_NSS_Hard_Slave (0x00000000U)
- #define SPI_NSS_Hard_Master_SS_Output_Enable (0x1<<2)
- #define SPI_NSS_Hard_Master_SS_No_output (0x00000000U)
- #define SPI_NSS_Internal_Soft_Reset (0x1<<9)
- #define SPI_NSS_Internal_Soft_Set ((0x1<<9)|(0x1<<8))
- #define SPI_Prescaler_By2 (0x00000000U)
- #define SPI_Prescaler_By4 (0b001<<3)
- #define SPI_Prescaler_By8 (0b010<<3)
- #define SPI_Prescaler_By16 (0b011<<3)
- #define SPI_Prescaler_By32 (0b100<<3)
- #define SPI_Prescaler_By64 (0b101<<3)
- #define SPI_Prescaler_By128 (0b110<<3)
- #define SPI_Prescaler_By256 (0b111<<3)
- #define SPI_IRQ_Enable_None (uint32_t)(0)
- #define SPI_IRQ_Enable_Tx_Only (uint32_t)(1<<7)
- #define SPI_IRQ_Enable_Rx_Only (uint32_t)(1<<6)
- #define SPI_IRQ_Enable_Err (uint32_t)(1<<5)

Enumerations

- enum SPI_PollingMechanism { Enabled , Disabled }
- Enumeration for SPI polling mechanisms.*

Functions

- void MCAL_SPI_Init (SPI_TypeDef *SPIx, S_SPI_Config_t *Config)
Initializes the SPI peripheral according to the specified configuration.
- void MCAL_SPI_DeInit (SPI_TypeDef *SPIx)
Deinitializes the SPI peripheral.
- void MCAL_SPI_GPIO_Set_Pins (SPI_TypeDef *SPIx)
Configures the GPIO pins used for SPI communication.
- void MCAL_SPI_SendData (SPI_TypeDef *SPIx, uint16_t *pTXBuffer, enum SPI_PollingMechanism Polling_En)
Sends data through SPI.
- void MCAL_SPI_ReceiveData (SPI_TypeDef *SPIx, uint16_t *pRXBuffer, enum SPI_PollingMechanism Polling_En)
Receives data from SPI.
- void MCAL_SPI_Tx_Rx (SPI_TypeDef *SPIx, uint16_t *TX_RX_pBuffer, enum SPI_PollingMechanism Polling_En)
Transmits and receives data through SPI.

7.21.1 Detailed Description

This file contains definitions and function prototypes for the STM32F401xx SPI driver.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

Copyright

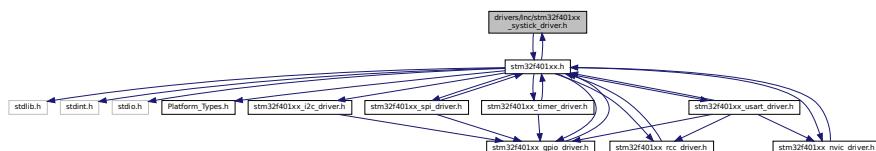
Copyright (c) 2024

This header file provides the definitions and function prototypes required for configuring and using the SPI (Serial Peripheral Interface) peripheral on the STM32F401xx microcontroller. It includes macros for configuration, a structure for SPI configuration, and function prototypes for SPI initialization, data transmission, and reception.

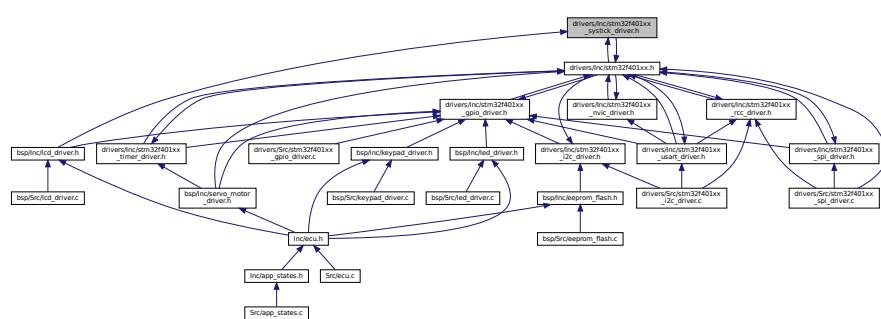
7.22 drivers/Inc/stm32f401xx systick driver.h File Reference

Header file for configuring and controlling the SysTick timer in the STM32F401xx MCU. This file contains the necessary structures, macros, and function prototypes for SysTick initialization, configuration, and handling delays.

```
#include "stm32f401xx.h"
Include dependency graph for stm32f401xx_systick_driver.h:
```



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



Classes

- struct **STK_config_t**

SysTick Configuration Structure Definition.

Macros

- #define **STK_INTERRUPT_MASK** 0x02UL
Interrupt enable/disable mask for the SysTick timer.
- #define **STK_CLK_MASK** 0x04UL
Clock source mask for the SysTick timer.
- #define **STK_RELOAD_MASK** 0x00FFFFFFUL
Reload value mask for the SysTick timer.
- #define **STK_FCPU** 8000000UL
CPU frequency definition for the SysTick timer.
- #define **STK_INTERRUPT_ENABLED** 0x02UL
- #define **STK_INTERRUPT_DISABLED** 0x00UL
- #define **STK_CLK_AHB** 0x04UL
- #define **STK_CLK_AHB_8** 0x00UL
- #define **STK_PERIODIC_MODE** 0x01U
- #define **STK_ONE_SHOT_MODE** 0x00U

Functions

- void **MCAL_STK_Config** (**STK_config_t** *_cfg)
Configures the SysTick timer with the specified parameters.
- void **MCAL_STK_SetReload** (**uint32** value)
Sets the reload value for the SysTick timer.
- void **MCAL_STK_SetCallback** (**void(*pfCallback)(void)**)
Sets the callback function for the SysTick timer interrupt.
- void **MCAL_STK_StartTimer** ()
Starts the SysTick timer.
- void **MCAL_STK_StopTimer** ()
Stops the SysTick timer.
- void **MCAL_STK_Delay** (**uint32** delay_ticks)
Delays the system by a specified number of ticks using the SysTick timer.
- void **MCAL_STK_Delay1ms** (**uint32** delay_ms)
Delays the system by a specified number of milliseconds.

7.22.1 Detailed Description

Header file for configuring and controlling the SysTick timer in the STM32F401xx MCU. This file contains the necessary structures, macros, and function prototypes for SysTick initialization, configuration, and handling delays.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file is part of the STM32F401xx peripheral driver library, focusing on the SysTick timer. The driver allows users to configure SysTick as a periodic or one-shot timer, handle interrupt configuration, and generate delays.

Copyright

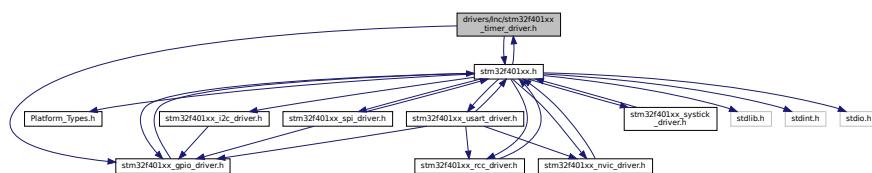
Copyright (c) 2024 Mohamed Ali Haoufa. All rights reserved. This software is provided "as is," without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement.

7.23 drivers/Inc/stm32f401xx_timer_driver.h File Reference

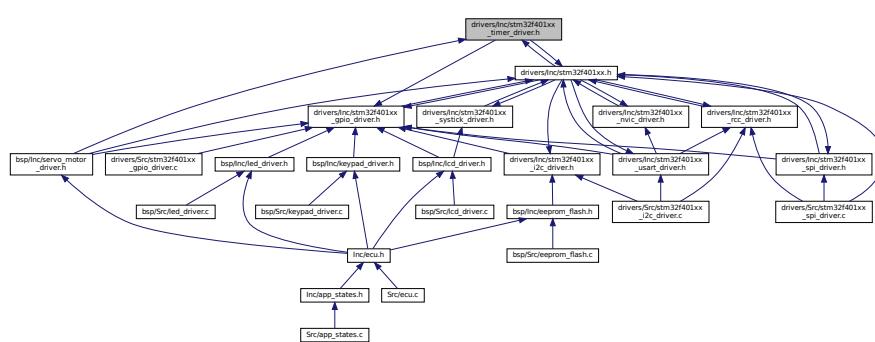
Header file providing timer functionalities for the STM32F401xx MCU. This file contains the necessary definitions, macros, and functions to manage Timer 2 (TIM2) in the STM32F401xx microcontroller.

```
#include "stm32f401xx.h"
#include "stm32f401xx_gpio_driver.h"
```

Include dependency graph for stm32f401xx_timer_driver.h:



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



Macros

- `#define RCC_APB1ENR *(volatile uint32 *) (RCC_BASE + 0x40)`
Enable the APB1 peripheral clock for TIM2.
- `#define RCC_APB2ENR *(volatile uint32 *) (RCC_BASE + 0x44)`
Enable the APB2 peripheral clock for TIM2.
- `#define TIM2_TIMER_BASE 0x40000000`
TIM2_Register TIM2 Register Base Addresses and Offsets.
- `#define TIM2_CR1 *(volatile uint32 *) (TIM2_TIMER_BASE + 0x00)`
Control register 1 (CR1) for TIM2.
- `#define TIM2_DIER *(volatile uint32 *) (TIM2_TIMER_BASE + 0x0C)`
DMA/interrupt enable register (DIER) for TIM2.
- `#define TIM2_SR *(volatile uint32 *) (TIM2_TIMER_BASE + 0x10)`
Status register (SR) for TIM2.
- `#define TIM2_CNT *(volatile uint32 *) (TIM2_TIMER_BASE + 0x24)`
Counter register (CNT) for TIM2.
- `#define TIM2_PSC *(volatile uint32 *) (TIM2_TIMER_BASE + 0x28)`
Prescaler register (PSC) for TIM2.
- `#define TIM2_ARR *(volatile uint32 *) (TIM2_TIMER_BASE + 0x2C)`
Auto-reload register (ARR) for TIM2.

Functions

- `void Timer2_init (void)`
Initialize Timer 2 (TIM2) for basic timing functionalities.
- `void dus (int us)`
Delay execution for a specified number of microseconds.
- `void dms (int ms)`
Delay execution for a specified number of milliseconds.

7.23.1 Detailed Description

Header file providing timer functionalities for the STM32F401xx MCU. This file contains the necessary definitions, macros, and functions to manage Timer 2 (TIM2) in the STM32F401xx microcontroller.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file is part of the STM32F401xx peripheral driver library, which offers low-level control of the MCU's timers, specifically Timer 2 (TIM2). It provides initialization functions for TIM2, along with delay functionalities in microseconds and milliseconds.

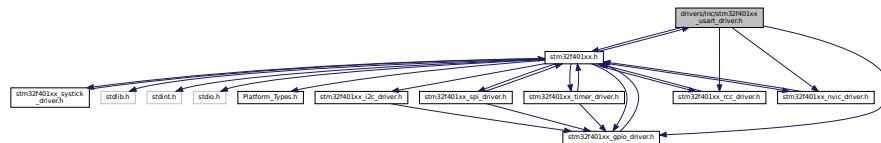
Copyright

Copyright (c) 2024 Mohamed Ali Haoufa. All rights reserved. This software is provided "as is," without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement.

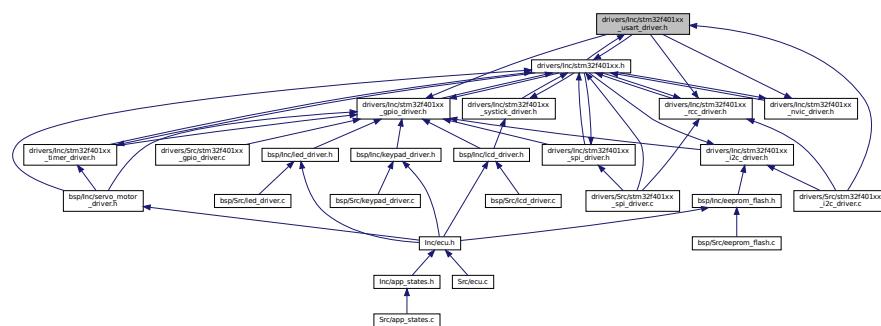
7.24 drivers/Inc/stm32f401xx_usart_driver.h File Reference

USART driver header file for STM32F401 series microcontrollers.

```
#include "stm32f401xx.h"
#include "stm32f401xx_gpio_driver.h"
#include "stm32f401xx_rcc_driver.h"
#include "stm32f401xx_nvic_driver.h"
Include dependency graph for stm32f401xx_usart_driver.h:
```



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



Classes

- struct `USART_cfg_t`

Configuration structure for USART (Universal Synchronous Asynchronous Receiver Transmitter) peripheral.

Macros

- #define `UART_Mode_RX` ((`uint32_t`)(`1UL << 2`))
USART Mode Definitions.
- #define `UART_Mode_TX` ((`uint32_t`)(`1UL << 3`))
- #define `UART_Mode_TX_RX` ((`uint32_t`)(`1UL << 2 | 1UL << 3`)))
- #define `UART_BaudRate_2400` 2400
Baud Rate Definitions.
- #define `UART_BaudRate_9600` 9600
- #define `UART_BaudRate_19200` 19200
- #define `UART_BaudRate_57600` 57600
- #define `UART_BaudRate_115200` 115200
- #define `UART_BaudRate_230400` 230400
- #define `UART_BaudRate_460800` 460800

- #define **UART_BaudRate_921600** 921600
- #define **UART_BaudRate_2250000** 2250000
- #define **UART_BaudRate_4500000** 4500000
- #define **UART_Payload_Length_8B** ((uint32_t)(0))

Payload Length Definitions.
- #define **UART_Payload_Length_9B** ((uint32_t)(1UL << 12))
- #define **UART_Parity_NONE** ((uint32_t)(0))

Parity Definitions.
- #define **UART_Parity_EVEN** ((uint32_t)(1UL << 10))
- #define **UART_Parity_ODD** ((uint32_t)(1UL << 10) | (1UL << 9))
- #define **UART_StopBits_half** ((uint32_t)(1 << 12))

Stop Bits Definitions.
- #define **UART_StopBits_1** ((uint32_t)(0))
- #define **UART_StopBits_1_half** ((uint32_t)(3 << 12))
- #define **UART_StopBits_2** ((uint32_t)(2 << 12))
- #define **UART_HwFlowCtl_NONE** ((uint32_t)(0))

Hardware Flow Control Definitions.
- #define **UART_HwFlowCtl_RTS** ((uint32_t)(1UL << 8))
- #define **UART_HwFlowCtl_CTS** ((uint32_t)(1UL << 9))
- #define **UART_HwFlowCtl_RTS_CTS** ((uint32_t)(1UL << 8 | 1UL << 9))
- #define **UART_IRQ_Enable_NONE** ((uint32_t)(0))

IRQ Enable Definitions.
- #define **UART_IRQ_Enable_TXE** ((uint32_t)(1UL << 7))
- #define **UART_IRQ_Enable_TC** ((uint32_t)(1UL << 6))
- #define **UART_IRQ_Enable_RXNE** ((uint32_t)(1UL << 5))
- #define **UART_IRQ_Enable_PE** ((uint32_t)(1UL << 8))

Enumerations

- enum **Polling_Mechanism** { **enable** , **disable** }
- Enumeration for Polling Mechanism.*

Functions

- void **MCAL_USART_Init** (**USART_TypeDef** *USARTx, **USART_cfg_t** *USART_cfg)

Initializes UART (Asynchronous mode only).
- void **MCAL_USART_DeInit** (**USART_TypeDef** *USARTx)

Deinitializes UART (Asynchronous mode only).
- void **MCAL_USART_GPIO_Set_Pins** (**USART_TypeDef** *USARTx)

Initializes GPIO pins for USART.
- void **MCAL_USART_SendData** (**USART_TypeDef** *USARTx, **uint16_t** *pTxBuffer, **Polling_Mechanism** PollingEn)

Sends buffer via USART.
- void **MCAL_USART_SendString** (**USART_TypeDef** *USARTx, **uint8_t** *str, **uint8_t** str_len)

Sends a string via USART.
- void **MCAL_USART_ReceiveData** (**USART_TypeDef** *USARTx, **uint16_t** *pRxBuffer, **Polling_Mechanism** PollingEn)

Receives buffer from USART.
- void **MCAL_USART_ReceiveBuffer** (**USART_TypeDef** *USARTx, **uint16_t** *pRxBuffer, **uint8_t** length)

Receives buffer from USART with specified length.
- void **MCAL_USART_Wait_TC** (**USART_TypeDef** *USARTx)

Waits until transmission is completed by polling on TC flag.

7.24.1 Detailed Description

USART driver header file for STM32F401 series microcontrollers.

Author

Mohamed Ali Haoufa

Version

1.0

Date

2024-09-21

This file contains the configuration and function prototypes for the USART peripheral. It includes macros for configuration, API declarations, and enumeration types.

Copyright

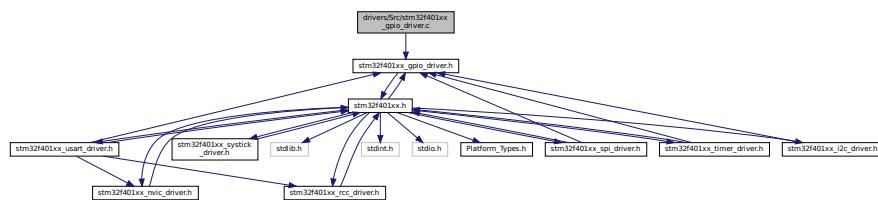
Copyright (c) 2024

7.25 drivers/Src/stm32f401xx_gpio_driver.c File Reference

GPIO driver for STM32F401xx microcontroller.

```
#include "stm32f401xx_gpio_driver.h"
```

Include dependency graph for stm32f401xx_gpio_driver.c:



Functions

- void **MCAL_GPIO_Init** (**GPIO_TypeDef** *GPIOx, **GPIO_PinConfig_t** *PinConfig)
Initializes the specified GPIO pin according to the provided configuration.
- void **MCAL_GPIO_DeInit** (**GPIO_TypeDef** *GPIOx)
Resets the specified GPIO port.
- uint8 **MCAL_GPIO_ReadPin** (**GPIO_TypeDef** *GPIOx, uint16 PinNumber)
Reads the value of a specific GPIO pin.
- uint16 **MCAL_GPIO_ReadPort** (**GPIO_TypeDef** *GPIOx)
Reads the value of a specific GPIO port.
- void **MCAL_GPIO_WritePin** (**GPIO_TypeDef** *GPIOx, uint16 PinNumber, uint8 Value)
Writes a value to a specific GPIO pin.
- void **MCAL_GPIO_WritePort** (**GPIO_TypeDef** *GPIOx, uint16 Value)
Writes a value to the entire GPIO port.
- void **MCAL_GPIO_TogglePin** (**GPIO_TypeDef** *GPIOx, uint16 PinNumber)
Toggles the value of a specific GPIO pin.
- void **MCAL_GPIO_TogglePort** (**GPIO_TypeDef** *GPIOx)
Toggles the value of a specific GPIO Port.
- uint8 **MCAL_GPIO_LockPin** (**GPIO_TypeDef** *GPIOx, uint16 PinNumber)
Locks the configuration of a specific GPIO pin.

7.25.1 Detailed Description

GPIO driver for STM32F401xx microcontroller.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file contains the implementations for GPIO operations for the STM32F401xx microcontroller. It includes initialization, de-initialization, and basic operations for GPIO pins and ports.

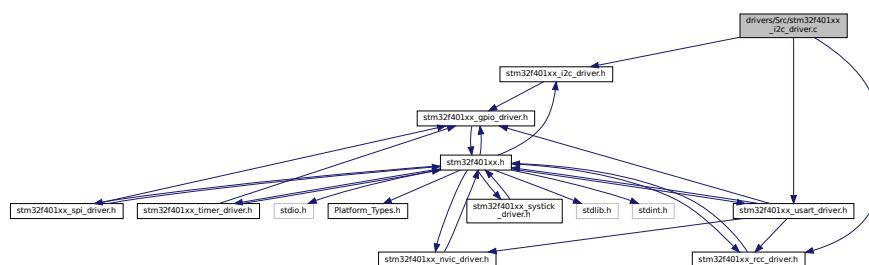
Copyright

Copyright (c) 2024

7.26 drivers/Src/stm32f401xx_i2c_driver.c File Reference

I2C Peripheral Driver Implementation for STM32F103C6.

```
#include "stm32f401xx_i2c_driver.h"
#include "stm32f401xx_usart_driver.h"
#include "stm32f401xx_rcc_driver.h"
Include dependency graph for stm32f401xx_i2c_driver.c:
```



Macros

- #define I2C1_Index 0
- #define I2C2_Index 1

Functions

- void `MCAL_I2C_Init (I2C_TypeDef *I2Cx, S_I2C_Config_t *I2C_Init_Struct)`
Initializes the I2C peripheral.
- void `MCAL_I2C_Deinit (I2C_TypeDef *I2Cx)`
De-initializes the I2C peripheral.
- void `MCAL_I2C_Set_GPIO (I2C_TypeDef *I2Cx)`
Configures GPIO pins for I2C functionality.
- void `MCAL_I2C_Master_Tx (I2C_TypeDef *I2Cx, uint16_t SlaveAdd, uint8_t *dataout, uint32_t datalen, StopCondition Stop, Repeated_Start start)`
Transmits data from the I2C master to a slave device.
- void `MCAL_I2C_Master_Rx (I2C_TypeDef *I2Cx, uint16_t SlaveAdd, uint8_t *dataout, uint32_t datalen, StopCondition Stop, Repeated_Start start)`
Receives data from an I2C slave device.
- void `I2C_Generate_Start (I2C_TypeDef *I2Cx, Functional_State state, Repeated_Start start)`
Generates a start condition or a repeated start condition.
- void `I2C_AcknowledgeConfig (I2C_TypeDef *I2Cx, Functional_State NewState)`
Configures the acknowledgment feature of the I2C peripheral.
- `FlagStatus I2C_Get_FlagStatus (I2C_TypeDef *I2Cx, Status flag)`
Checks the status of I2C flags.
- void `I2C_SendAddress (I2C_TypeDef *I2Cx, uint16_t SlaveAddress, I2C_Direction Direction)`
Sends a 7-bit slave address to the I2C peripheral.
- void `I2C_Generate_Stop (I2C_TypeDef *I2Cx, Functional_State NewState)`
Generates a stop condition on the I2C bus.
- void `MCAL_I2C_Slave_SendData (I2C_TypeDef *I2Cx, uint8_t data)`
Sends a single byte of data in slave mode.
- uint8_t `MCAL_I2C_SlaveReceiveData (I2C_TypeDef *I2Cx)`
Receives a single byte of data in slave mode.

Variables

- `S_I2C_Config_t Global_I2C_Config [2]`

7.26.1 Detailed Description

I2C Peripheral Driver Implementation for STM32F103C6.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file implements the driver functions for the I2C (Inter-Integrated Circuit) peripheral of the STM32F103C6 microcontroller. The driver includes functions for initialization, de-initialization, data transmission, and reception. Additionally, it provides utility functions to facilitate I2C operations.

Copyright

Copyright (c) 2023 Fam Ayman. All rights reserved.

7.26.2 Macro Definition Documentation

7.26.2.1 I2C1_Index

```
#define I2C1_Index 0
```

Index for I2C1 peripheral.

7.26.2.2 I2C2_Index

```
#define I2C2_Index 1
```

Index for I2C2 peripheral.

7.26.3 Variable Documentation

7.26.3.1 Global_I2C_Config

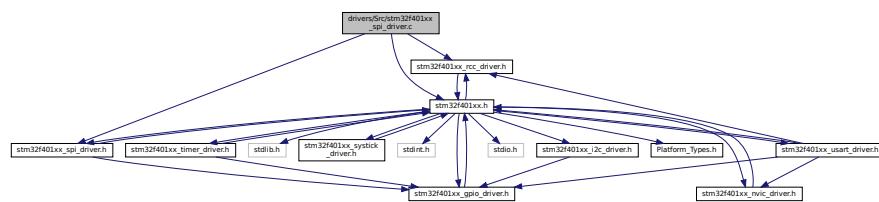
```
S_I2C_Config_t Global_I2C_Config[2]
```

Global array to hold I2C configurations.

7.27 drivers/Src/stm32f401xx_spi_driver.c File Reference

Contains the implementation of the SPI driver APIs for STM32F401.

```
#include "stm32f401xx_rcc_driver.h"
#include "stm32f401xx_spi_driver.h"
#include "stm32f401xx.h"
Include dependency graph for stm32f401xx_spi_driver.c:
```



Macros

- `#define SPI1_Index 0`
Macro definitions for SPI indices and status flags.
- `#define SPI2_Index 1`
- `#define SPI_SR_TXE (uint8_t)(1 << 1)`
- `#define SPI_SR_RXNE (uint8_t)(1 << 0)`

Functions

- `void MCAL_SPI_Init (SPI_TypeDef *SPIx, S_SPI_Config_t *Config)`
Initializes the SPI peripheral.
- `void MCAL_SPI_DeInit (SPI_TypeDef *SPIx)`
Deinitializes the SPI peripheral.
- `void MCAL_SPI_SendData (SPI_TypeDef *SPIx, uint16_t *pTXBuffer, enum SPI_PollingMechanism Polling_En)`
Sends data through the SPI peripheral.
- `void MCAL_SPI_ReceiveData (SPI_TypeDef *SPIx, uint16_t *pRXBuffer, enum SPI_PollingMechanism Polling_En)`
Receives data from the SPI peripheral.
- `void MCAL_SPI_GPIO_Set_Pins (SPI_TypeDef *SPIx)`
Configures GPIO pins for SPI communication based on SPI peripheral.
- `void MCAL_SPI_Tx_Rx (SPI_TypeDef *SPIx, uint16_t *TX_RX_pBuffer, enum SPI_PollingMechanism Polling_En)`
Transmit and receive data through SPI.
- `void SPI1_IRQHandler (void)`
SPI1 interrupt handler.
- `void SPI2_IRQHandler (void)`
SPI2 interrupt handler.

Variables

- `S_SPI_Config_t * Global_SPI_Config [2] = {NULL, NULL}`
Global array to hold SPI configurations.

7.27.1 Detailed Description

Contains the implementation of the SPI driver APIs for STM32F401.

Version

0.1

Date

2024-09-21

Copyright

Copyright (c) 2024

This file provides the implementation of the APIs for configuring and using the SPI (Serial Peripheral Interface) module of the STM32F401 microcontroller. It includes initialization, deinitialization, data transmission, and reception functions, as well as GPIO pin configuration and IRQ handling.

7.27.2 Macro Definition Documentation

7.27.2.1 SPI1_Index

```
#define SPI1_Index 0
```

Macro definitions for SPI indices and status flags.

Index for SPI1 in Global_SPI_Config array

7.27.2.2 SPI2_Index

```
#define SPI2_Index 1
```

Index for SPI2 in Global_SPI_Config array

7.27.2.3 SPI_SR_RXNE

```
#define SPI_SR_RXNE (uint8_t)(1 << 0)
```

SPI Status Register RXNE flag

7.27.2.4 SPI_SR_TXE

```
#define SPI_SR_TXE (uint8_t)(1 << 1)
```

SPI Status Register TXE flag

7.27.3 Function Documentation

7.27.3.1 SPI1_IRQHandler()

```
void SPI1_IRQHandler (
    void )
```

SPI1 interrupt handler.

This function handles the interrupt for SPI1. It retrieves the interrupt status (TXE, RXE, and ERRI) and calls the appropriate callback function.

Return values

None

7.27.3.2 SPI2_IRQHandler()

```
void SPI2_IRQHandler (
```

SPI2 interrupt handler.

This function handles the interrupt for SPI2. It retrieves the interrupt status (TXE, RXE, and ERRI) and calls the appropriate callback function.

Return values

None

7.27.4 Variable Documentation

7.27.4.1 Global_SPI_Config

```
S_SPI_Config_t* Global_SPI_Config[2] = {NULL, NULL}
```

Global array to hold SPI configurations.

This array holds pointers to SPI configurations for SPI1 and SPI2.

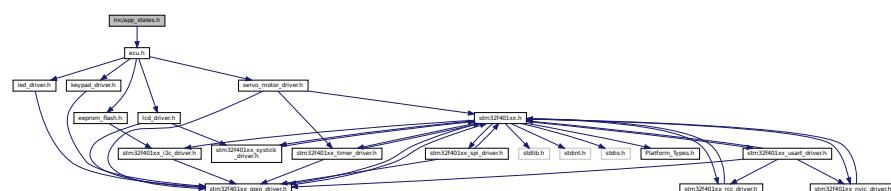
- Index 0: SPI1 configuration
 - Index 1: SPI2 configuration

7.28 Inc/app_states.h File Reference

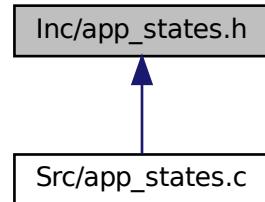
Header file containing definitions and macros for managing application states on the STM32F401xx MCU.

```
#include "ecu.h"
```

Include dependency graph for app_states.h:



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



Macros

- `#define STATE_API(_NAME) void ST_##_NAME(void)`
Macro to define a state handler function.
- `#define STATE_NAME(_NAME) ST_##_NAME`
Macro to create a state handler function name.

Enumerations

- `enum STATES {
 Init_STATE , Admin_STATE , Idle_STATE , Enter_Gate_STATE ,
 Exit_Gate_STATE , Full_STATE }`
Enum representing various application states.

Functions

- `STATE_API (Init_STATE)`
Handler function for the initialization state.
- `STATE_API (Admin_STATE)`
Handler function for the administration state.
- `STATE_API (Idle_STATE)`
Handler function for the idle state.
- `STATE_API (Enter_Gate_STATE)`
Handler function for the entering gate state.
- `STATE_API (Exit_Gate_STATE)`
Handler function for the exiting gate state.
- `STATE_API (Full_STATE)`
Handler function for the full state.

Variables

- `void(* fp_App_State_Handler)()`
Function pointer for the current application state handler.

7.28.1 Detailed Description

Header file containing definitions and macros for managing application states on the STM32F401xx MCU.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file provides definitions for the application states and macros used in managing these states for the STM32F401xx microcontroller. It includes state definitions and function pointer declarations for handling different application states.

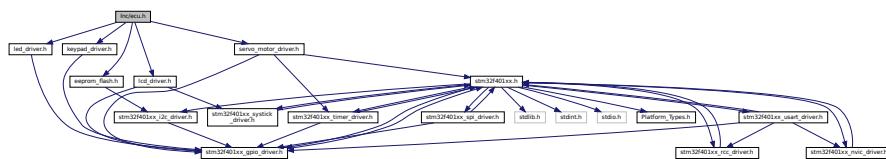
Copyright

Copyright (c) 2024

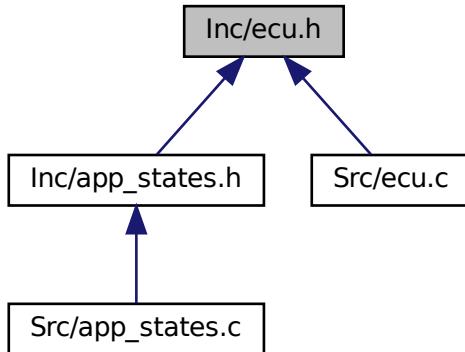
7.29 Inc/ecu.h File Reference

Header file containing all the necessary information about the STM32F401xx MCU.

```
#include "servo_motor_driver.h"
#include "lcd_driver.h"
#include "led_driver.h"
#include "keypad_driver.h"
#include "eeprom_flash.h"
Include dependency graph for ecu.h:
```



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



Macros

- `#define NO_OF_SLOTS 3`
Number of parking slots.
- `#define USERS_COUNT 3`
Total number of users.
- `#define USER1 0`
User 1 ID.
- `#define USER2 1`
User 2 ID.
- `#define USER3 2`
User 3 ID.
- `#define ENTER_USART_INSTANT USART1`
Enter gate USART instance.
- `#define EXIT_USART_INSTANT USART2`
Exit gate USART instance.
- `#define ENTER_PIR_PORT GPIOA`
Enter gate PIR sensor GPIO port.
- `#define ENTER_PIR_PIN GPIO_PIN_7`
Enter gate PIR sensor GPIO pin.
- `#define EXIT_PIR_PORT GPIOA`
Exit gate PIR sensor GPIO port.
- `#define EXIT_PIR_PIN GPIO_PIN_1`
Exit gate PIR sensor GPIO pin.

Enumerations

- enum `ID_Check_Result { ID_NOT_Found , ID_Found }`

Functions

- void `ECU_Init` (void)
Initializes all hardware components and peripherals.
- void `Admin_Init` (void)
Initializes the admin settings, including setting up user IDs.
- void `UserLCD_PrintFreeSlots` (void)
Prints the number of free parking slots on the LCD.
- `ID_Check_Result Check_ID` (`uint8 _ID`)
Checks if the given ID exists in the system.
- `uint8 Check_Flag` (`uint8 _ID`)
Returns the status flag associated with the given ID.
- void `Flag_SET_RESET` (`uint8 _ID`)
Toggles the flag associated with the given ID.
- `uint8 Check_Password` (`uint8 _ID`)
Checks if the entered password matches the stored password for the given ID.
- void `combineArrays` (`uint8_t *array1, int size1, uint8_t *array2, int size2, uint8_t *array3, int size3, uint8_t *array4, int size4, uint8_t *combinedArray`)
Combines four arrays into a single array.
- void `Enter_Gate_Open` (void)
Opens the entrance gate and displays the status on the LCD.
- void `Exit_Gate_Open` (void)
Opens the exit gate and displays the status on the LCD.
- void `Wrong_RFID` (void)
Triggers the alarm and prints "UNKNOWN ID!" on the LCD.
- void `Trigger_Alarm` (`USART_TypeDef *_USART`)
Echoes the entered ID via USART and flashes the red LED.

7.29.1 Detailed Description

Header file containing all the necessary information about the STM32F401xx MCU.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file contains the definitions and macros for the STM32F401xx microcontroller peripherals and memory maps. It also provides APIs for the initialization and management of key hardware components and system peripherals used in the application.

Copyright

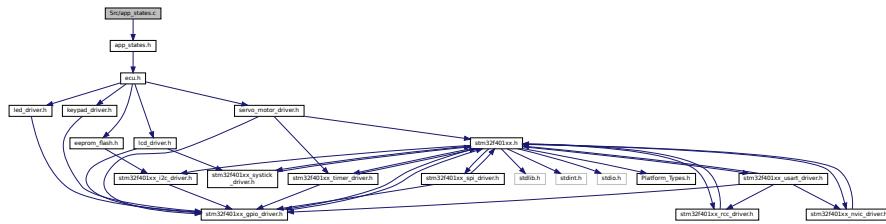
Copyright (c) 2024

7.30 Src/app_states.c File Reference

Application state machine implementation for managing entry/exit gates and parking slots using the STM32F401xx MCU.

```
#include "app_states.h"
```

Include dependency graph for app_states.c:



Functions

- **STATE_API (Init_STATE)**
Initializes the system and sets up the initial state.
- **STATE_API (Admin_STATE)**
Admin state to initialize the user IDs and prepare the system.
- **STATE_API (Idle_STATE)**
Idle state where the system waits for user input (entry/exit requests).
- **STATE_API (Enter_Gate_STATE)**
Handles the logic for opening the entry gate.
- **STATE_API (Exit_Gate_STATE)**
Handles the logic for opening the exit gate.
- **STATE_API (Full_STATE)**
Full state where the parking lot is full, and no entry is allowed.

Variables

- volatile uint8 Enter_Flag
Flags and variables used for managing system state.
- volatile uint8 Exit_Flag
Flags for entry and exit state triggers.
- uint8 Free_Slots
Available parking slots and LCD print flag.
- uint8 Print_Slots_LCD_Flag
Function pointer to manage current state transitions.
- STATES_APP_Current_State
Current application state.

7.30.1 Detailed Description

Application state machine implementation for managing entry/exit gates and parking slots using the STM32F401xx MCU.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file defines the application states and associated functions to manage the parking system, including initializing the system, handling user entry/exit, and updating the number of available parking slots.

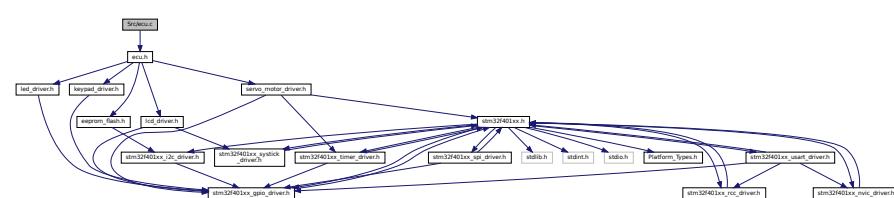
Copyright

Copyright (c) 2024

7.31 Src/ecu.c File Reference

Header file containing all necessary information and functionality for managing STM32F401xx MCU-based peripherals.

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



Macros

- #define **SIZE1** 3
Definitions for combining arrays.
- #define **SIZE2** 4
- #define **SIZE3** 4
- #define **SIZE4** 4
- #define **TOTAL_SIZE** (**SIZE1** + **SIZE2** + **SIZE3** + **SIZE4**)

Functions

- void [Enter_UART_CallBack \(void\)](#)
Callback function for handling entry UART events.
- void [Exit_UART_CallBack \(void\)](#)
Callback function for handling exit UART events.
- void [ECU_Init \(void\)](#)
Initializes all hardware components and peripherals.
- void [Admin_Init \(void\)](#)
Initializes user IDs and displays them on the LCD.
- void [UserLCD_PrintFreeSlots \(void\)](#)
Prints the number of free slots on the UserLCD.
- [ID_Check_Result Check_ID \(uint8 _ID\)](#)
Checks if the given ID is present in the saved IDs.
- [uint8 Check_Flag \(uint8 _ID\)](#)
Returns the status flag associated with the given ID.
- void [Flag_SET_RESET \(uint8 _ID\)](#)
Toggles the flag associated with the given ID.
- [uint8 Check_Password \(uint8 _ID\)](#)
Checks if the entered password matches the stored password for the given ID.
- void [combineArrays \(uint8_t *array1, int size1, uint8_t *array2, int size2, uint8_t *array3, int size3, uint8_t *array4, int size4, uint8_t *combinedArray\)](#)
Combines four arrays into a single array.
- void [Enter_Gate_Open \(void\)](#)
Opens the entry gate and displays a message on the LCD.
- void [Exit_Gate_Open \(void\)](#)
Opens the exit gate and displays a message on the LCD.
- void [Wrong_RFID \(void\)](#)
Triggers an alarm and displays an "UNKNOWN ID!" message on the LCD.
- void [Trigger_Alarm \(USART_TypeDef *_USART\)](#)
Triggers an alarm by echoing the received ID on UART and flashing the red LED.

Variables

- volatile uint8 [Enter_Flag](#)
Flags and variables used for managing system state.
- volatile uint8 [Exit_Flag](#)
- [uint8 Free_Slots = 3](#)
- [uint8 Print_Slots_LCD_Flag](#)

7.31.1 Detailed Description

Header file containing all necessary information and functionality for managing STM32F401xx MCU-based peripherals.

Contains the main program logic for managing STM32F401xx MCU-based peripherals.

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file contains the initialization and control logic for peripherals including LEDs, LCDs, UART, PIR sensors, and servo motors.

Copyright

Copyright (c) 2024

Author

Mohamed Ali Haoufa

Version

0.1

Date

2024-09-21

This file includes the initialization and control logic for peripherals such as LEDs, LCDs, UART, PIR sensors, and servo motors. It operates the system based on a state machine, managing various hardware components.

Copyright

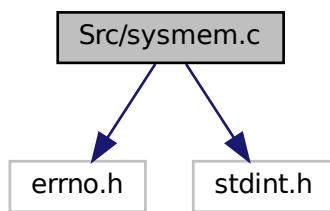
Copyright (c) 2024 Mohamed Ali Haoufa. All rights reserved.

7.32 Src/sysmem.c File Reference

STM32CubeIDE System Memory calls file.

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

Include dependency graph for sysmem.c:



Functions

- void * [_sbrk](#) (ptrdiff_t incr)

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

7.32.1 Detailed Description

STM32CubeIDE System Memory calls file.

Author

Auto-generated by STM32 VS Code Extension

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

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.

7.32.2 Function Documentation

7.32.2.1 [_sbrk\(\)](#)

```
void* _sbrk (
    ptrdiff_t incr )
```

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

```
* ##### .data # .bss #      newlib heap      #          MSP stack      #
* #      #      #                      # Reserved by _Min_Stack_Size #
* ##### #      ##### #      ##### #      ##### #      ##### #      ##### #
* ^-- RAM start      ^-- _end                  _estack, RAM end --^
*
```

This implementation starts allocating at the '_end' linker symbol. The '_Min_Stack_Size' linker symbol reserves a memory for the MSP stack. The implementation considers '_estack' linker symbol to be RAM end. NOTE: If the MSP stack, at any point during execution, grows larger than the reserved size, please increase the '_Min_Stack_Size'.

Parameters

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

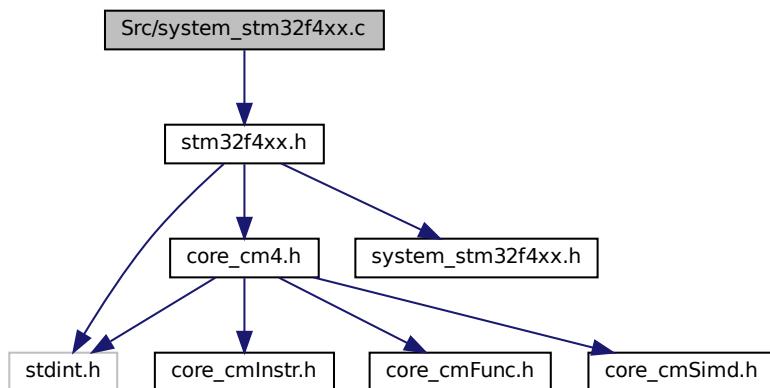
Returns

Pointer to allocated memory

7.33 Src/system_stm32f4xx.c File Reference

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File. This file contains the system clock configuration for STM32F4xx devices.

```
#include "stm32f4xx.h"
Include dependency graph for system_stm32f4xx.c:
```



Macros

- #define VECT_TAB_OFFSET 0x00
- #define PLL_Q 7

Functions

- void [SystemInit](#)(void)
Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void [SystemInit_Old](#)(void)
- void [SystemCoreClockUpdate](#)(void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Variables

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

7.33.1 Detailed Description

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File. This file contains the system clock configuration for STM32F4xx devices.

Author

MCD Application Team

Version

V1.8.0

Date

09-November-2016

1. This file provides two functions and one global variable to be called from user application:
 - `SystemInit()`: Setsups the system clock (System clock source, PLL Multiplier and Divider factors, AHB/APBx prescalers and Flash settings), depending on the configuration made in the clock xls tool. 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.
2. After each device reset the HSI (16 MHz) is used as system clock source. Then `SystemInit()` function is called, in "startup_stm32f4xx.s" file, to configure the system clock before to branch to main program.
3. If the system clock source selected by user fails to startup, the `SystemInit()` function will do nothing and HSI still used as system clock source. User can add some code to deal with this issue inside the `SetSysClock()` function.
4. The default value of HSE crystal is set to 25MHz, refer to "HSE_VALUE" define in "stm32f4xx.h" file. When HSE is used as system clock source, directly or through PLL, and you are using different crystal you have to adapt the HSE value to your own configuration.

7.33.2 5. This file configures the system clock as follows:

=====

7.33.2.1 Supported STM32F40xxx/41xxx devices

7.33.2.2 System Clock source | PLL (HSE)

7.33.2.3 SYSCLK(Hz) | 168000000

7.33.2.4 HCLK(Hz) | 168000000

7.33.2.5 AHB Prescaler | 1

7.33.2.6 APB1 Prescaler | 4

7.33.2.7 APB2 Prescaler | 2

7.33.2.8 HSE Frequency(Hz) | 25000000

7.33.2.9 PLL_M | 25

7.33.2.10 PLL_N | 336

7.33.2.11 PLL_P | 2

7.33.2.12 PLL_Q | 7

7.33.2.13 PLLI2S_N | NA

7.33.2.14 PLLI2S_R | NA

7.33.2.15 I2S input clock | NA

7.33.2.16 VDD(V) | 3.3

7.33.2.17 Main regulator output voltage | Scale1 mode

7.33.2.18 Flash Latency(WS) | 5

7.33.2.19 Prefetch Buffer | ON

7.33.2.20 Instruction cache | ON

7.33.2.21 Data cache | ON

Require 48MHz for USB OTG FS, | Disabled

7.33.2.22 SDIO and RNG clock |

7.33.3 =====

7.33.3.1 Supported STM32F42xxx/43xxx devices

7.33.3.2 System Clock source | PLL (HSE)

7.33.3.3 SYSCLK(Hz) | 180000000

7.33.3.4 HCLK(Hz) | 180000000

7.33.3.5 AHB Prescaler | 1

7.33.3.6 APB1 Prescaler | 4

7.33.3.7 APB2 Prescaler | 2

7.33.3.8 HSE Frequency(Hz) | 25000000

7.33.3.9 PLL_M | 25

7.33.3.10 PLL_N | 360

7.33.3.11 PLL_P | 2

7.33.3.12 PLL_Q | 7

7.33.3.13 PLLI2S_N | NA

7.33.3.14 PLLI2S_R | NA

7.33.3.15 I2S input clock | NA

7.33.3.16 VDD(V) | 3.3

7.33.3.17 Main regulator output voltage | Scale1 mode

7.33.3.18 Flash Latency(WS) | 5

7.33.3.19 Prefetch Buffer | ON

7.33.3.20 Instruction cache | ON

7.33.3.21 Data cache | ON

Require 48MHz for USB OTG FS, | Disabled

7.33.3.22 SDIO and RNG clock |

7.33.4 =====

7.33.4.1 Supported STM32F401xx devices

7.33.4.2 System Clock source | PLL (HSE)

7.33.4.3 SYSCLK(Hz) | 84000000

7.33.4.4 HCLK(Hz) | 84000000

7.33.4.5 AHB Prescaler | 1

7.33.4.6 APB1 Prescaler | 2

7.33.4.7 APB2 Prescaler | 1

7.33.4.8 HSE Frequency(Hz) | 25000000

7.33.4.9 PLL_M | 25

7.33.4.10 PLL_N | 336

7.33.4.11 PLL_P | 4

7.33.4.12 PLL_Q | 7

7.33.4.13 PLLI2S_N | NA

7.33.4.14 PLLI2S_R | NA

7.33.4.15 I2S input clock | NA

7.33.4.16 VDD(V) | 3.3

7.33.4.17 Main regulator output voltage | Scale1 mode

7.33.4.18 Flash Latency(WS) | 2

7.33.4.19 Prefetch Buffer | ON

7.33.4.20 Instruction cache | ON

7.33.4.21 Data cache | ON

Require 48MHz for USB OTG FS, | Disabled

7.33.4.22 SDIO and RNG clock |

7.33.5 =====

7.33.5.1 Supported STM32F411xx/STM32F410xx devices

7.33.5.2 System Clock source | PLL (HSI)

7.33.5.3 SYSCLK(Hz) | 100000000

7.33.5.4 HCLK(Hz) | 100000000

7.33.5.5 AHB Prescaler | 1

7.33.5.6 APB1 Prescaler | 2

7.33.5.7 APB2 Prescaler | 1

7.33.5.8 HSI Frequency(Hz) | 16000000

7.33.5.9 PLL_M | 16

7.33.5.10 PLL_N | 400

7.33.5.11 PLL_P | 4

7.33.5.12 PLL_Q | 7

7.33.5.13 PLLI2S_N | NA

7.33.5.14 PLLI2S_R | NA

7.33.5.15 I2S input clock | NA

7.33.5.16 VDD(V) | 3.3

7.33.5.17 Main regulator output voltage | Scale1 mode

7.33.5.18 Flash Latency(WS) | 3

7.33.5.19 Prefetch Buffer | ON

7.33.5.20 Instruction cache | ON

7.33.5.21 Data cache | ON

Require 48MHz for USB OTG FS, | Disabled

7.33.5.22 SDIO and RNG clock |

7.33.6 =====

7.33.6.1 Supported STM32F446xx devices

7.33.6.2 System Clock source | PLL (HSE)

7.33.6.3 SYSCLK(Hz) | 180000000

7.33.6.4 HCLK(Hz) | 180000000

7.33.6.5 AHB Prescaler | 1

7.33.6.6 APB1 Prescaler | 4

7.33.6.7 APB2 Prescaler | 2

7.33.6.8 HSE Frequency(Hz) | 8000000

7.33.6.9 PLL_M | 8

7.33.6.10 PLL_N | 360

7.33.6.11 PLL_P | 2

7.33.6.12 PLL_Q | 7

7.33.6.13 PLL_R | NA

7.33.6.14 PLLI2S_M | NA

7.33.6.15 PLLI2S_N | NA

7.33.6.16 PLLI2S_P | NA

7.33.6.17 PLLI2S_Q | NA

7.33.6.18 PLLI2S_R | NA

7.33.6.19 I2S input clock | NA

7.33.6.20 VDD(V) | 3.3

7.33.6.21 Main regulator output voltage | Scale1 mode

7.33.6.22 Flash Latency(WS) | 5

7.33.6.23 Prefetch Buffer | ON

7.33.6.24 Instruction cache | ON

7.33.6.25 Data cache | ON

Require 48MHz for USB OTG FS, | Disabled

7.33.6.26 SDIO and RNG clock |

=====

Attention

© COPYRIGHT 2016 STMicroelectronics

Licensed under MCD-ST Liberty SW License Agreement V2, (the "License"); You may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.st.com/software_license_agreement_liberty_v2

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Index

__CM4_CMSIS_VERSION
 CMSIS, [398](#)
__CM4_CMSIS_VERSION_MAIN
 CMSIS, [398](#)
__CM4_CMSIS_VERSION_SUB
 CMSIS, [398](#)
__CM4_REV
 Configuration_section_for\CMSIS, [523](#)
__CORE_CM4_H_DEPENDANT
 CMSIS, [398](#)
__CORTEX_M
 CMSIS, [399](#)
__FPU_PRESENT
 Configuration_section_for\CMSIS, [523](#)
_I
 CMSIS, [399](#)
_IO
 CMSIS, [399](#)
_MPU_PRESENT
 Configuration_section_for\CMSIS, [523](#)
_NVIC_PRIO_BITS
 Configuration_section_for\CMSIS, [524](#)
_O
 CMSIS, [399](#)
_STM32F4XX_STDPERIPH_VERSION
 Library_configuration_section, [520](#)
_STM32F4XX_STDPERIPH_VERSION_MAIN
 Library_configuration_section, [521](#)
_STM32F4XX_STDPERIPH_VERSION_RC
 Library_configuration_section, [521](#)
_STM32F4XX_STDPERIPH_VERSION_SUB1
 Library_configuration_section, [521](#)
_STM32F4XX_STDPERIPH_VERSION_SUB2
 Library_configuration_section, [521](#)
_Vendor_SysTickConfig
 Configuration_section_for\CMSIS, [524](#)
_reserved0
 ITM Functions, [124, 125](#)
_reserved1
 ITM Functions, [125, 126](#)
_sbrk
 sysmem.c, [1232](#)

ACPR
 ITM Functions, [126](#)
ACR
 CMSIS, [461](#)
ACTLR
 ITM Functions, [126](#)
ADC_CCR_ADCPRE
 Peripheral_Registers_Bits_Definition, [623](#)
ADC_CCR_ADCPRE_0
 Peripheral_Registers_Bits_Definition, [623](#)
ADC_CCR_ADCPRE_1
 Peripheral_Registers_Bits_Definition, [623](#)
ADC_CCR_DDS
 Peripheral_Registers_Bits_Definition, [624](#)
ADC_CCR_DELAY
 Peripheral_Registers_Bits_Definition, [624](#)
ADC_CCR_DELAY_0
 Peripheral_Registers_Bits_Definition, [624](#)
ADC_CCR_DELAY_1
 Peripheral_Registers_Bits_Definition, [624](#)
ADC_CCR_DELAY_2
 Peripheral_Registers_Bits_Definition, [624](#)
ADC_CCR_DELAY_3
 Peripheral_Registers_Bits_Definition, [624](#)
ADC_CCR_DMA
 Peripheral_Registers_Bits_Definition, [624](#)
ADC_CCR_DMA_0
 Peripheral_Registers_Bits_Definition, [624](#)
ADC_CCR_DMA_1
 Peripheral_Registers_Bits_Definition, [625](#)
ADC_CCR_MULTI
 Peripheral_Registers_Bits_Definition, [625](#)
ADC_CCR_MULTI_0
 Peripheral_Registers_Bits_Definition, [625](#)
ADC_CCR_MULTI_1
 Peripheral_Registers_Bits_Definition, [625](#)
ADC_CCR_MULTI_2
 Peripheral_Registers_Bits_Definition, [625](#)
ADC_CCR_MULTI_3
 Peripheral_Registers_Bits_Definition, [625](#)
ADC_CCR_MULTI_4
 Peripheral_Registers_Bits_Definition, [625](#)
ADC_CCR_TSVREFE
 Peripheral_Registers_Bits_Definition, [625](#)
ADC_CCR_VBATE
 Peripheral_Registers_Bits_Definition, [626](#)
ADC_CDR_DATA1
 Peripheral_Registers_Bits_Definition, [626](#)
ADC_CDR_DATA2
 Peripheral_Registers_Bits_Definition, [626](#)
ADC_Common_TypeDef
 [999](#)
ADC_CR1_AWDCH
 Peripheral_Registers_Bits_Definition, [626](#)
ADC_CR1_AWDCH_0
 Peripheral_Registers_Bits_Definition, [626](#)
ADC_CR1_AWDCH_1
 Peripheral_Registers_Bits_Definition, [626](#)

Peripheral_Registers_Bits_Definition, 626
ADC_CR1_AWDCH_2
 Peripheral_Registers_Bits_Definition, 626
ADC_CR1_AWDCH_3
 Peripheral_Registers_Bits_Definition, 626
ADC_CR1_AWDCH_4
 Peripheral_Registers_Bits_Definition, 627
ADC_CR1_AWDEN
 Peripheral_Registers_Bits_Definition, 627
ADC_CR1_AWDIE
 Peripheral_Registers_Bits_Definition, 627
ADC_CR1_AWDSGL
 Peripheral_Registers_Bits_Definition, 627
ADC_CR1_DISCEN
 Peripheral_Registers_Bits_Definition, 627
ADC_CR1_DISCNUM
 Peripheral_Registers_Bits_Definition, 627
ADC_CR1_DISCNUM_0
 Peripheral_Registers_Bits_Definition, 627
ADC_CR1_DISCNUM_1
 Peripheral_Registers_Bits_Definition, 628
ADC_CR1_DISCNUM_2
 Peripheral_Registers_Bits_Definition, 628
ADC_CR1_EOCIE
 Peripheral_Registers_Bits_Definition, 628
ADC_CR1_JAUTO
 Peripheral_Registers_Bits_Definition, 628
ADC_CR1_JAWDEN
 Peripheral_Registers_Bits_Definition, 628
ADC_CR1_JDISCEN
 Peripheral_Registers_Bits_Definition, 628
ADC_CR1_JEOCIE
 Peripheral_Registers_Bits_Definition, 628
ADC_CR1_OVRIE
 Peripheral_Registers_Bits_Definition, 629
ADC_CR1_RES
 Peripheral_Registers_Bits_Definition, 629
ADC_CR1_RES_0
 Peripheral_Registers_Bits_Definition, 629
ADC_CR1_RES_1
 Peripheral_Registers_Bits_Definition, 629
ADC_CR1_SCAN
 Peripheral_Registers_Bits_Definition, 629
ADC_CR2_ADON
 Peripheral_Registers_Bits_Definition, 629
ADC_CR2_ALIGN
 Peripheral_Registers_Bits_Definition, 629
ADC_CR2_CONT
 Peripheral_Registers_Bits_Definition, 630
ADC_CR2_DDS
 Peripheral_Registers_Bits_Definition, 630
ADC_CR2_DMA
 Peripheral_Registers_Bits_Definition, 630
ADC_CR2_EOCS
 Peripheral_Registers_Bits_Definition, 630
ADC_CR2_EXTEN
 Peripheral_Registers_Bits_Definition, 630
ADC_CR2_EXTEN_0
 Peripheral_Registers_Bits_Definition, 630
ADC_CR2_EXTEN_1
 Peripheral_Registers_Bits_Definition, 630
ADC_CR2_EXTSEL
 Peripheral_Registers_Bits_Definition, 631
ADC_CR2_EXTSEL_0
 Peripheral_Registers_Bits_Definition, 631
ADC_CR2_EXTSEL_1
 Peripheral_Registers_Bits_Definition, 631
ADC_CR2_EXTSEL_2
 Peripheral_Registers_Bits_Definition, 631
ADC_CR2_EXTSEL_3
 Peripheral_Registers_Bits_Definition, 631
ADC_CR2_JEXTEN
 Peripheral_Registers_Bits_Definition, 631
ADC_CR2_JEXTEN_0
 Peripheral_Registers_Bits_Definition, 631
ADC_CR2_JEXTEN_1
 Peripheral_Registers_Bits_Definition, 631
ADC_CR2_JEXTSEL
 Peripheral_Registers_Bits_Definition, 632
ADC_CR2_JEXTSEL_0
 Peripheral_Registers_Bits_Definition, 632
ADC_CR2_JEXTSEL_1
 Peripheral_Registers_Bits_Definition, 632
ADC_CR2_JEXTSEL_2
 Peripheral_Registers_Bits_Definition, 632
ADC_CR2_JEXTSEL_3
 Peripheral_Registers_Bits_Definition, 632
ADC_CR2_JSWSTART
 Peripheral_Registers_Bits_Definition, 632
ADC_CR2_SWSTART
 Peripheral_Registers_Bits_Definition, 632
ADC_CSR_AWD1
 Peripheral_Registers_Bits_Definition, 632
ADC_CSR_AWD2
 Peripheral_Registers_Bits_Definition, 633
ADC_CSR_AWD3
 Peripheral_Registers_Bits_Definition, 633
ADC_CSR_EOC1
 Peripheral_Registers_Bits_Definition, 633
ADC_CSR_EOC2
 Peripheral_Registers_Bits_Definition, 633
ADC_CSR_EOC3
 Peripheral_Registers_Bits_Definition, 633
ADC_CSR_JEOC1
 Peripheral_Registers_Bits_Definition, 633
ADC_CSR_JEOC2
 Peripheral_Registers_Bits_Definition, 633
ADC_CSR_JEOC3
 Peripheral_Registers_Bits_Definition, 633
ADC_CSR_JSTRT1
 Peripheral_Registers_Bits_Definition, 634
ADC_CSR_JSTRT2
 Peripheral_Registers_Bits_Definition, 634
ADC_CSR_JSTRT3
 Peripheral_Registers_Bits_Definition, 634
ADC_CSR_OVR1

Peripheral_Registers_Bits_Definition, 634
ADC_CSR_OVR2
 Peripheral_Registers_Bits_Definition, 634
ADC_CSR_OVR3
 Peripheral_Registers_Bits_Definition, 634
ADC_CSR_STRT1
 Peripheral_Registers_Bits_Definition, 634
ADC_CSR_STRT2
 Peripheral_Registers_Bits_Definition, 634
ADC_CSR_STRT3
 Peripheral_Registers_Bits_Definition, 635
ADC_DR_ADC2DATA
 Peripheral_Registers_Bits_Definition, 635
ADC_DR_DATA
 Peripheral_Registers_Bits_Definition, 635
ADC_HTR_HT
 Peripheral_Registers_Bits_Definition, 635
ADC IRQn
 Configuration_section_for_CMSIS, 525
ADC_JDR1_JDATA
 Peripheral_Registers_Bits_Definition, 635
ADC_JDR2_JDATA
 Peripheral_Registers_Bits_Definition, 635
ADC_JDR3_JDATA
 Peripheral_Registers_Bits_Definition, 635
ADC_JDR4_JDATA
 Peripheral_Registers_Bits_Definition, 635
ADC_JOFR1_JOFFSET1
 Peripheral_Registers_Bits_Definition, 636
ADC_JOFR2_JOFFSET2
 Peripheral_Registers_Bits_Definition, 636
ADC_JOFR3_JOFFSET3
 Peripheral_Registers_Bits_Definition, 636
ADC_JOFR4_JOFFSET4
 Peripheral_Registers_Bits_Definition, 636
ADC_JSQR_JL
 Peripheral_Registers_Bits_Definition, 636
ADC_JSQR_JL_0
 Peripheral_Registers_Bits_Definition, 636
ADC_JSQR_JL_1
 Peripheral_Registers_Bits_Definition, 636
ADC_JSQR_JSQ1
 Peripheral_Registers_Bits_Definition, 636
ADC_JSQR_JSQ1_0
 Peripheral_Registers_Bits_Definition, 637
ADC_JSQR_JSQ1_1
 Peripheral_Registers_Bits_Definition, 637
ADC_JSQR_JSQ1_2
 Peripheral_Registers_Bits_Definition, 637
ADC_JSQR_JSQ1_3
 Peripheral_Registers_Bits_Definition, 637
ADC_JSQR_JSQ1_4
 Peripheral_Registers_Bits_Definition, 637
ADC_JSQR_JSQ2
 Peripheral_Registers_Bits_Definition, 637
ADC_JSQR_JSQ2_0
 Peripheral_Registers_Bits_Definition, 637
ADC_JSQR_JSQ2_1
 Peripheral_Registers_Bits_Definition, 637
ADC_JSQR_JSQ2_2
 Peripheral_Registers_Bits_Definition, 638
ADC_JSQR_JSQ2_3
 Peripheral_Registers_Bits_Definition, 638
ADC_JSQR_JSQ2_4
 Peripheral_Registers_Bits_Definition, 638
ADC_JSQR_JSQ3
 Peripheral_Registers_Bits_Definition, 638
ADC_JSQR_JSQ3_0
 Peripheral_Registers_Bits_Definition, 638
ADC_JSQR_JSQ3_1
 Peripheral_Registers_Bits_Definition, 638
ADC_JSQR_JSQ3_2
 Peripheral_Registers_Bits_Definition, 638
ADC_JSQR_JSQ3_3
 Peripheral_Registers_Bits_Definition, 638
ADC_JSQR_JSQ3_4
 Peripheral_Registers_Bits_Definition, 639
ADC_JSQR_JSQ4
 Peripheral_Registers_Bits_Definition, 639
ADC_JSQR_JSQ4_0
 Peripheral_Registers_Bits_Definition, 639
ADC_JSQR_JSQ4_1
 Peripheral_Registers_Bits_Definition, 639
ADC_JSQR_JSQ4_2
 Peripheral_Registers_Bits_Definition, 639
ADC_JSQR_JSQ4_3
 Peripheral_Registers_Bits_Definition, 639
ADC_JSQR_JSQ4_4
 Peripheral_Registers_Bits_Definition, 639
ADC_LTR_LT
 Peripheral_Registers_Bits_Definition, 639
ADC_SMPR1_SMP10
 Peripheral_Registers_Bits_Definition, 640
ADC_SMPR1_SMP10_0
 Peripheral_Registers_Bits_Definition, 640
ADC_SMPR1_SMP10_1
 Peripheral_Registers_Bits_Definition, 640
ADC_SMPR1_SMP10_2
 Peripheral_Registers_Bits_Definition, 640
ADC_SMPR1_SMP11
 Peripheral_Registers_Bits_Definition, 640
ADC_SMPR1_SMP11_0
 Peripheral_Registers_Bits_Definition, 640
ADC_SMPR1_SMP11_1
 Peripheral_Registers_Bits_Definition, 640
ADC_SMPR1_SMP11_2
 Peripheral_Registers_Bits_Definition, 640
ADC_SMPR1_SMP12
 Peripheral_Registers_Bits_Definition, 641
ADC_SMPR1_SMP12_0
 Peripheral_Registers_Bits_Definition, 641
ADC_SMPR1_SMP12_1
 Peripheral_Registers_Bits_Definition, 641
ADC_SMPR1_SMP12_2
 Peripheral_Registers_Bits_Definition, 641
ADC_SMPR1_SMP13

Peripheral_Registers_Bits_Definition, 656
ADC_SQR2_SQ8
 Peripheral_Registers_Bits_Definition, 656
ADC_SQR2_SQ8_0
 Peripheral_Registers_Bits_Definition, 656
ADC_SQR2_SQ8_1
 Peripheral_Registers_Bits_Definition, 656
ADC_SQR2_SQ8_2
 Peripheral_Registers_Bits_Definition, 656
ADC_SQR2_SQ8_3
 Peripheral_Registers_Bits_Definition, 656
ADC_SQR2_SQ8_4
 Peripheral_Registers_Bits_Definition, 656
ADC_SQR2_SQ9
 Peripheral_Registers_Bits_Definition, 656
ADC_SQR2_SQ9_0
 Peripheral_Registers_Bits_Definition, 657
ADC_SQR2_SQ9_1
 Peripheral_Registers_Bits_Definition, 657
ADC_SQR2_SQ9_2
 Peripheral_Registers_Bits_Definition, 657
ADC_SQR2_SQ9_3
 Peripheral_Registers_Bits_Definition, 657
ADC_SQR2_SQ9_4
 Peripheral_Registers_Bits_Definition, 657
ADC_SQR3_SQ1
 Peripheral_Registers_Bits_Definition, 657
ADC_SQR3_SQ1_0
 Peripheral_Registers_Bits_Definition, 657
ADC_SQR3_SQ1_1
 Peripheral_Registers_Bits_Definition, 657
ADC_SQR3_SQ1_2
 Peripheral_Registers_Bits_Definition, 658
ADC_SQR3_SQ1_3
 Peripheral_Registers_Bits_Definition, 658
ADC_SQR3_SQ1_4
 Peripheral_Registers_Bits_Definition, 658
ADC_SQR3_SQ2
 Peripheral_Registers_Bits_Definition, 658
ADC_SQR3_SQ2_0
 Peripheral_Registers_Bits_Definition, 658
ADC_SQR3_SQ2_1
 Peripheral_Registers_Bits_Definition, 658
ADC_SQR3_SQ2_2
 Peripheral_Registers_Bits_Definition, 658
ADC_SQR3_SQ2_3
 Peripheral_Registers_Bits_Definition, 658
ADC_SQR3_SQ2_4
 Peripheral_Registers_Bits_Definition, 659
ADC_SQR3_SQ3
 Peripheral_Registers_Bits_Definition, 659
ADC_SQR3_SQ3_0
 Peripheral_Registers_Bits_Definition, 659
ADC_SQR3_SQ3_1
 Peripheral_Registers_Bits_Definition, 659
ADC_SQR3_SQ3_2
 Peripheral_Registers_Bits_Definition, 659
ADC_SQR3_SQ3_3
 Peripheral_Registers_Bits_Definition, 659
 Peripheral_Registers_Bits_Definition, 659
ADC_SQR3_SQ3_4
 Peripheral_Registers_Bits_Definition, 659
ADC_SQR3_SQ4
 Peripheral_Registers_Bits_Definition, 659
ADC_SQR3_SQ4_0
 Peripheral_Registers_Bits_Definition, 660
ADC_SQR3_SQ4_1
 Peripheral_Registers_Bits_Definition, 660
ADC_SQR3_SQ4_2
 Peripheral_Registers_Bits_Definition, 660
ADC_SQR3_SQ4_3
 Peripheral_Registers_Bits_Definition, 660
ADC_SQR3_SQ4_4
 Peripheral_Registers_Bits_Definition, 660
ADC_SQR3_SQ5
 Peripheral_Registers_Bits_Definition, 660
ADC_SQR3_SQ5_0
 Peripheral_Registers_Bits_Definition, 660
ADC_SQR3_SQ5_1
 Peripheral_Registers_Bits_Definition, 660
ADC_SQR3_SQ5_2
 Peripheral_Registers_Bits_Definition, 661
ADC_SQR3_SQ5_3
 Peripheral_Registers_Bits_Definition, 661
ADC_SQR3_SQ5_4
 Peripheral_Registers_Bits_Definition, 661
ADC_SQR3_SQ6
 Peripheral_Registers_Bits_Definition, 661
ADC_SQR3_SQ6_0
 Peripheral_Registers_Bits_Definition, 661
ADC_SQR3_SQ6_1
 Peripheral_Registers_Bits_Definition, 661
ADC_SQR3_SQ6_2
 Peripheral_Registers_Bits_Definition, 661
ADC_SQR3_SQ6_3
 Peripheral_Registers_Bits_Definition, 661
ADC_SQR3_SQ6_4
 Peripheral_Registers_Bits_Definition, 662
ADC_SR_AWD
 Peripheral_Registers_Bits_Definition, 662
ADC_SR_EOC
 Peripheral_Registers_Bits_Definition, 662
ADC_SR_JEOC
 Peripheral_Registers_Bits_Definition, 662
ADC_SR_JSTRT
 Peripheral_Registers_Bits_Definition, 662
ADC_SR_OVR
 Peripheral_Registers_Bits_Definition, 662
ADC_SR_STRT
 Peripheral_Registers_Bits_Definition, 662
ADC_TypeDef, 999
Admin_Init
 API_Definitions, 372
 ECU APIs, 362
Admin_STATE
 Application States, 352
ADR

ITM Functions, 126
AFR
 CMSIS, 461
 GPIO_TypeDef, 1017
AFSR
 ITM Functions, 126
AHB Peripherals Base Addresses, 158
AHB1ENR
 CMSIS, 461
 RCC_TypeDef, 1033
AHB1LPENR
 CMSIS, 461
 RCC_TypeDef, 1033
AHB1RSTR
 CMSIS, 461
 RCC_TypeDef, 1034
AHB2ENR
 CMSIS, 461
 RCC_TypeDef, 1034
AHB2LPENR
 CMSIS, 461
 RCC_TypeDef, 1034
AHB2PERIPH_BASE
 Peripheral_memory_map, 534
AHB2RSTR
 CMSIS, 462
 RCC_TypeDef, 1034
AHB3ENR
 CMSIS, 462
AHB3LPENR
 CMSIS, 462
AHB3RSTR
 CMSIS, 462
AIRCR
 ITM Functions, 127
 SCB_TypeDef, 1047
ALRMAR
 CMSIS, 462
ALRMASSR
 CMSIS, 462
ALRMBR
 CMSIS, 462
ALRMBSSR
 CMSIS, 462
AMTCR
 CMSIS, 463
APB1 Peripherals Base Addresses, 159
APB1ENR
 CMSIS, 463
 RCC_TypeDef, 1034
APB1FZ
 CMSIS, 463
APB1LPENR
 CMSIS, 463
 RCC_TypeDef, 1034
APB1RSTR
 CMSIS, 463
 RCC_TypeDef, 1034
APB2 Peripherals Base Addresses, 158
APB2ENR
 CMSIS, 463
 RCC_TypeDef, 1034
APB2FZ
 CMSIS, 463
APB2LPENR
 CMSIS, 463
 RCC_TypeDef, 1035
APB2RSTR
 CMSIS, 464
 RCC_TypeDef, 1035
API Function Prototypes, 16, 20, 25, 33, 38, 259, 277, 291, 310, 326, 334, 341, 346
dms, 342
dus, 342
EEPROM_Init, 16
EEPROM_Read_byte, 17
EEPROM_Write_NBytes, 18
I2C_ACKnowledgeConfig, 278
I2C_Generate_Start, 278
I2C_Generate_Stop, 279
I2C_Get_FlagStatus, 279
I2C_SendAddress, 280
keypad_Get_Pressed_Key, 20
keypad_init, 21
LCD_Init, 25
LCD_Send_Char, 26
LCD_Send_Char_Pos, 27
LCD_Send_Command, 27
LCD_Send_Enable_Signal, 28
LCD_Send_String, 29
LCD_Send_String_Pos, 30
LCD_Set_Cursor, 30
LED_Init, 34
LED_Toggle, 35
LED_TurnOff, 35
LED_TurnOn, 36
MCAL_GPIO_DeInit, 259
MCAL_GPIO_Init, 260
MCAL_GPIO_LockPin, 261
MCAL_GPIO_ReadPin, 262
MCAL_GPIO_ReadPort, 263
MCAL_GPIO_TogglePin, 264
MCAL_GPIO_TogglePort, 264
MCAL_GPIO_WritePin, 265
MCAL_GPIO_WritePort, 266
MCAL_I2C_Deinit, 280
MCAL_I2C_Init, 281
MCAL_I2C_Master_Rx, 282
MCAL_I2C_Master_Tx, 282
MCAL_I2C_Set_GPIO, 283
MCAL_I2C_Slave_SendData, 284
MCAL_I2C_SlaveReceiveData, 284
MCAL_NVIC_ClearPendingIRQ, 292
MCAL_NVIC_DisableIRQ, 293
MCAL_NVIC_EnableIRQ, 293
MCAL_NVIC_GetActive, 294

MCAL_NVIC_GetPendingIRQ, 295
 MCAL_NVIC_GetPriority, 295
 MCAL_NVIC_GetPriorityGrouping, 296
 MCAL_NVIC_SetPendingIRQ, 296
 MCAL_NVIC_SetPriority, 298
 MCAL_NVIC_SetPriorityGrouping, 299
 MCAL_NVIC_SystemReset, 299
 MCAL_RCC_Enable_Peripheral, 311
 MCAL_RCC_GetHCLKFreq, 311
 MCAL_RCC_GetPCLK1Freq, 312
 MCAL_RCC_GetPCLK2Freq, 312
 MCAL_RCC_GetSYS_CLKFreq, 313
 MCAL_RCC_Reset_Peripheral, 314
 MCAL_RCC_Select_Clock, 314
 MCAL_SPI_DeInit, 326
 MCAL_SPI_GPIO_Set_Pins, 327
 MCAL_SPI_Init, 328
 MCAL_SPI_ReceiveData, 328
 MCAL_SPI_SendData, 329
 MCAL_SPI_Tx_Rx, 330
 MCAL_STK_Config, 335
 MCAL_STK_Delay, 336
 MCAL_STK_Delay1ms, 336
 MCAL_STK_SetCallback, 337
 MCAL_STK_SetReload, 338
 MCAL_STK_StartTimer, 338
 MCAL_STK_StopTimer, 339
 MCAL_USART_DeInit, 347
 MCAL_USART_GPIO_Set_Pins, 347
 MCAL_USART_Init, 348
 MCAL_USART_ReceiveBuffer, 348
 MCAL_USART_ReceiveData, 349
 MCAL_USART_SendData, 349
 MCAL_USART_SendString, 350
 MCAL_USART_Wait_TC, 350
 RCC_GetPLLOutputClock, 314
 Servo1_Entry_Gate, 39
 Servo1_Entry_Gate_Init, 39
 Servo2_Exit_Gate, 40
 Servo2_Exit_Gate_Init, 41
 Timer2_init, 343
 API_Definitions, 371
 Admin_Init, 372
 Check_Flag, 373
 Check_ID, 373
 Check_Password, 373
 combineArrays, 374
 ECU_Init, 375
 Enter_Gate_Open, 375
 Enter_UART_CallBack, 375
 Exit_Gate_Open, 376
 Exit_UART_CallBack, 376
 Flag_SET_RESET, 376
 Trigger_Alarm, 377
 UserLCD_PrintFreeSlots, 377
 Wrong_RFID, 378
 Application States, 350
 Admin_STATE, 352
 Enter_Gate_STATE, 352
 Exit_Gate_STATE, 352
 fp_App_State_Handler, 354
 Free_Slots, 354
 Full_STATE, 352
 Idle_STATE, 352
 Init_STATE, 352
 STATE_API, 352, 353
 STATES, 352
 APSR_C_Msk
 CMSIS, 399
 Status and Control Registers, 44
 APSR_C_Pos
 CMSIS, 399
 Status and Control Registers, 44
 APSR_GE_Msk
 CMSIS, 399
 Status and Control Registers, 44
 APSR_GE_Pos
 CMSIS, 400
 Status and Control Registers, 44
 APSR_N_Msk
 CMSIS, 400
 Status and Control Registers, 45
 APSR_N_Pos
 CMSIS, 400
 Status and Control Registers, 45
 APSR_Q_Msk
 CMSIS, 400
 Status and Control Registers, 45
 APSR_Q_Pos
 CMSIS, 400
 Status and Control Registers, 45
 APSR_Type, 1000
 APSR_V_Msk
 CMSIS, 400
 Status and Control Registers, 45
 APSR_V_Pos
 CMSIS, 400
 Status and Control Registers, 45
 APSR_Z_Msk
 CMSIS, 400
 Status and Control Registers, 45
 APSR_Z_Pos
 CMSIS, 401
 Status and Control Registers, 45
 ARG
 CMSIS, 464
 ARR
 CMSIS, 464
 TIM1_TypeDef, 1057
 AT24C1024B EEPROM Driver, 15
 AWCR
 CMSIS, 464
 b
 ITM Functions, 127
 BCCR
 CMSIS, 464

BDCR CMSIS, 467
 RCC_TypeDef, 1035

BDTR CMSIS, 464
 TIM1_TypeDef, 1057

BFAR ITM Functions, 127
 SCB_TypeDef, 1047

BFCR CMSIS, 464

BGCLUT CMSIS, 465

BGCMAR CMSIS, 465

BGCOLR CMSIS, 465

BGMAR CMSIS, 465

BGOR CMSIS, 465

BGPFCCR CMSIS, 465

BKP0R CMSIS, 465

BKP10R CMSIS, 465

BKP11R CMSIS, 466

BKP12R CMSIS, 466

BKP13R CMSIS, 466

BKP14R CMSIS, 466

BKP15R CMSIS, 466

BKP16R CMSIS, 466

BKP17R CMSIS, 466

BKP18R CMSIS, 466

BKP19R CMSIS, 467

BKP1R CMSIS, 467

BKP2R CMSIS, 467

BKP3R CMSIS, 467

BKP4R CMSIS, 467

BKP5R CMSIS, 467

BKP6R CMSIS, 467

BKP7R CMSIS, 467

BKP8R CMSIS, 468

BKP9R CMSIS, 468

BKPSRAM_BASE Peripheral_memory_map, 534

BKPSRAM_BB_BASE Peripheral_memory_map, 534

boolean Platform Standard Types, 152

BPCR CMSIS, 468

BRR CMSIS, 468
 USART_TypeDef, 1062

bsp/Inc/eeprom_flash.h, 1065

bsp/Inc/keypad_driver.h, 1067

bsp/Inc/lcd_driver.h, 1069

bsp/Inc/led_driver.h, 1072

bsp/Src/eeprom_flash.c, 1074

bsp/Src/keypad_driver.c, 1075

bsp/Src/lcd_driver.c, 1076

bsp/Src/led_driver.c, 1078

BSRR GPIO_TypeDef, 1017

BSRRH CMSIS, 468

BSRRL CMSIS, 468

BTR CMSIS, 468

BusFault_IRQn Configuration_section_for\CMSIS, 524

C ITM Functions, 127, 128

CACR CMSIS, 469

CALIB ITM Functions, 128
 STK_TypeDef, 1053

CALIBR CMSIS, 469

Callback_Function STK_config_t, 1052

CALR CMSIS, 469

CAN_BTR_BRP Peripheral_Registers_Bits_Definition, 663

CAN_BTR_LBKM Peripheral_Registers_Bits_Definition, 663

CAN_BTR_SILM Peripheral_Registers_Bits_Definition, 663

CAN_BTR_SJW Peripheral_Registers_Bits_Definition, 663

CAN_BTR_TS1 Peripheral_Registers_Bits_Definition, 663

CAN_BTR_TS2 Peripheral_Registers_Bits_Definition, 663

Peripheral_Registers_Bits_Definition, 779
CAN_FFA1R_FFA3
 Peripheral_Registers_Bits_Definition, 779
CAN_FFA1R_FFA4
 Peripheral_Registers_Bits_Definition, 780
CAN_FFA1R_FFA5
 Peripheral_Registers_Bits_Definition, 780
CAN_FFA1R_FFA6
 Peripheral_Registers_Bits_Definition, 780
CAN_FFA1R_FFA7
 Peripheral_Registers_Bits_Definition, 780
CAN_FFA1R_FFA8
 Peripheral_Registers_Bits_Definition, 780
CAN_FFA1R_FFA9
 Peripheral_Registers_Bits_Definition, 780
CAN_FIFOMailBox_TypeDef, 1000
CAN_FilterRegister_TypeDef, 1001
CAN_FM1R_FBM
 Peripheral_Registers_Bits_Definition, 780
CAN_FM1R_FBM0
 Peripheral_Registers_Bits_Definition, 780
CAN_FM1R_FBM1
 Peripheral_Registers_Bits_Definition, 781
CAN_FM1R_FBM10
 Peripheral_Registers_Bits_Definition, 781
CAN_FM1R_FBM11
 Peripheral_Registers_Bits_Definition, 781
CAN_FM1R_FBM12
 Peripheral_Registers_Bits_Definition, 781
CAN_FM1R_FBM13
 Peripheral_Registers_Bits_Definition, 781
CAN_FM1R_FBM2
 Peripheral_Registers_Bits_Definition, 781
CAN_FM1R_FBM3
 Peripheral_Registers_Bits_Definition, 781
CAN_FM1R_FBM4
 Peripheral_Registers_Bits_Definition, 781
CAN_FM1R_FBM5
 Peripheral_Registers_Bits_Definition, 782
CAN_FM1R_FBM6
 Peripheral_Registers_Bits_Definition, 782
CAN_FM1R_FBM7
 Peripheral_Registers_Bits_Definition, 782
CAN_FM1R_FBM8
 Peripheral_Registers_Bits_Definition, 782
CAN_FM1R_FBM9
 Peripheral_Registers_Bits_Definition, 782
CAN_FMR_INIT
 Peripheral_Registers_Bits_Definition, 782
CAN_FS1R_FSC
 Peripheral_Registers_Bits_Definition, 782
CAN_FS1R_FSC0
 Peripheral_Registers_Bits_Definition, 782
CAN_FS1R_FSC1
 Peripheral_Registers_Bits_Definition, 783
CAN_FS1R_FSC10
 Peripheral_Registers_Bits_Definition, 783
CAN_FS1R_FSC11
 Peripheral_Registers_Bits_Definition, 783
Peripheral_Registers_Bits_Definition, 783
CAN_FS1R_FSC12
 Peripheral_Registers_Bits_Definition, 783
CAN_FS1R_FSC13
 Peripheral_Registers_Bits_Definition, 783
CAN_FS1R_FSC2
 Peripheral_Registers_Bits_Definition, 783
CAN_FS1R_FSC3
 Peripheral_Registers_Bits_Definition, 783
CAN_FS1R_FSC4
 Peripheral_Registers_Bits_Definition, 783
CAN_FS1R_FSC5
 Peripheral_Registers_Bits_Definition, 784
CAN_FS1R_FSC6
 Peripheral_Registers_Bits_Definition, 784
CAN_FS1R_FSC7
 Peripheral_Registers_Bits_Definition, 784
CAN_FS1R_FSC8
 Peripheral_Registers_Bits_Definition, 784
CAN_FS1R_FSC9
 Peripheral_Registers_Bits_Definition, 784
CAN_IER_BOFIE
 Peripheral_Registers_Bits_Definition, 784
CAN_IER_EPVIE
 Peripheral_Registers_Bits_Definition, 784
CAN_IER_ERRIE
 Peripheral_Registers_Bits_Definition, 784
CAN_IER_EWGIE
 Peripheral_Registers_Bits_Definition, 785
CAN_IER_FFIE0
 Peripheral_Registers_Bits_Definition, 785
CAN_IER_FFIE1
 Peripheral_Registers_Bits_Definition, 785
CAN_IER_FMPIE0
 Peripheral_Registers_Bits_Definition, 785
CAN_IER_FMPIE1
 Peripheral_Registers_Bits_Definition, 785
CAN_IER_FOVIE0
 Peripheral_Registers_Bits_Definition, 785
CAN_IER_FOVIE1
 Peripheral_Registers_Bits_Definition, 785
CAN_IER_LCIE
 Peripheral_Registers_Bits_Definition, 785
CAN_IER_SLKIE
 Peripheral_Registers_Bits_Definition, 786
CAN_IER_TMEIE
 Peripheral_Registers_Bits_Definition, 786
CAN_IER_WKUIE
 Peripheral_Registers_Bits_Definition, 786
CAN_MCR_ABOM
 Peripheral_Registers_Bits_Definition, 786
CAN_MCR_AWUM
 Peripheral_Registers_Bits_Definition, 786
CAN_MCR_INRQ
 Peripheral_Registers_Bits_Definition, 786
CAN_MCR_NART
 Peripheral_Registers_Bits_Definition, 786
CAN_MCR_RESET

- Peripheral_Registers_Bits_Definition, 786
CAN_MCR_RFLM
 Peripheral_Registers_Bits_Definition, 787
CAN_MCR_SLEEP
 Peripheral_Registers_Bits_Definition, 787
CAN_MCR_TTCM
 Peripheral_Registers_Bits_Definition, 787
CAN_MCR_TXFP

Peripheral_Registers_Bits_Definition, 794
CAN_TD1R_DATA7
 Peripheral_Registers_Bits_Definition, 794
CAN_TD2R_DATA4
 Peripheral_Registers_Bits_Definition, 794
CAN_TD2R_DATA5
 Peripheral_Registers_Bits_Definition, 794
CAN_TD2R_DATA6
 Peripheral_Registers_Bits_Definition, 794
CAN_TD2R_DATA7
 Peripheral_Registers_Bits_Definition, 794
CAN_TDL0R_DATA0
 Peripheral_Registers_Bits_Definition, 794
CAN_TDL0R_DATA1
 Peripheral_Registers_Bits_Definition, 795
CAN_TDL0R_DATA2
 Peripheral_Registers_Bits_Definition, 795
CAN_TDL0R_DATA3
 Peripheral_Registers_Bits_Definition, 795
CAN_TDL1R_DATA0
 Peripheral_Registers_Bits_Definition, 795
CAN_TDL1R_DATA1
 Peripheral_Registers_Bits_Definition, 795
CAN_TDL1R_DATA2
 Peripheral_Registers_Bits_Definition, 795
CAN_TDL1R_DATA3
 Peripheral_Registers_Bits_Definition, 795
CAN_TDL2R_DATA0
 Peripheral_Registers_Bits_Definition, 795
CAN_TDL2R_DATA1
 Peripheral_Registers_Bits_Definition, 796
CAN_TDL2R_DATA2
 Peripheral_Registers_Bits_Definition, 796
CAN_TDL2R_DATA3
 Peripheral_Registers_Bits_Definition, 796
CAN_TDT0R_DLC
 Peripheral_Registers_Bits_Definition, 796
CAN_TDT0R_TGT
 Peripheral_Registers_Bits_Definition, 796
CAN_TDT0R_TIME
 Peripheral_Registers_Bits_Definition, 796
CAN_TDT1R_DLC
 Peripheral_Registers_Bits_Definition, 796
CAN_TDT1R_TGT
 Peripheral_Registers_Bits_Definition, 796
CAN_TDT1R_TIME
 Peripheral_Registers_Bits_Definition, 797
CAN_TDT2R_DLC
 Peripheral_Registers_Bits_Definition, 797
CAN_TDT2R_TGT
 Peripheral_Registers_Bits_Definition, 797
CAN_TDT2R_TIME
 Peripheral_Registers_Bits_Definition, 797
CAN_TI0R_EXID
 Peripheral_Registers_Bits_Definition, 797
CAN_TI0R_IDE
 Peripheral_Registers_Bits_Definition, 797
CAN_TI0R_RTR
 Peripheral_Registers_Bits_Definition, 797
Peripheral_Registers_Bits_Definition, 797
CAN_TI0R_STID
 Peripheral_Registers_Bits_Definition, 797
CAN_TI0R_TXRQ
 Peripheral_Registers_Bits_Definition, 798
CAN_TI1R_EXID
 Peripheral_Registers_Bits_Definition, 798
CAN_TI1R_IDE
 Peripheral_Registers_Bits_Definition, 798
CAN_TI1R_RTR
 Peripheral_Registers_Bits_Definition, 798
CAN_TI1R_STID
 Peripheral_Registers_Bits_Definition, 798
CAN_TI1R_TXRQ
 Peripheral_Registers_Bits_Definition, 798
CAN_TI2R_EXID
 Peripheral_Registers_Bits_Definition, 798
CAN_TI2R_IDE
 Peripheral_Registers_Bits_Definition, 798
CAN_TI2R_RTR
 Peripheral_Registers_Bits_Definition, 799
CAN_TI2R_STID
 Peripheral_Registers_Bits_Definition, 799
CAN_TI2R_TXRQ
 Peripheral_Registers_Bits_Definition, 799
CAN_TSR_ABRQ0
 Peripheral_Registers_Bits_Definition, 799
CAN_TSR_ABRQ1
 Peripheral_Registers_Bits_Definition, 799
CAN_TSR_ABRQ2
 Peripheral_Registers_Bits_Definition, 799
CAN_TSR_ALST0
 Peripheral_Registers_Bits_Definition, 799
CAN_TSR_ALST1
 Peripheral_Registers_Bits_Definition, 799
CAN_TSR_ALST2
 Peripheral_Registers_Bits_Definition, 800
CAN_TSR_CODE
 Peripheral_Registers_Bits_Definition, 800
CAN_TSR_LOW
 Peripheral_Registers_Bits_Definition, 800
CAN_TSR_LOW0
 Peripheral_Registers_Bits_Definition, 800
CAN_TSR_LOW1
 Peripheral_Registers_Bits_Definition, 800
CAN_TSR_LOW2
 Peripheral_Registers_Bits_Definition, 800
CAN_TSR_RQCP0
 Peripheral_Registers_Bits_Definition, 800
CAN_TSR_RQCP1
 Peripheral_Registers_Bits_Definition, 800
CAN_TSR_RQCP2
 Peripheral_Registers_Bits_Definition, 801
CAN_TSR_TERR0
 Peripheral_Registers_Bits_Definition, 801
CAN_TSR_TERR1
 Peripheral_Registers_Bits_Definition, 801
CAN_TSR_TERR2

Peripheral_Registers_Bits_Definition, 801
CAN_TSR_TME
Peripheral_Registers_Bits_Definition, 801
CAN_TSR_TME0
Peripheral_Registers_Bits_Definition, 801
CAN_TSR_TME1
Peripheral_Registers_Bits_Definition, 801
CAN_TSR_TME2
Peripheral_Registers_Bits_Definition, 801
CAN_TSR_TXOK0
Peripheral_Registers_Bits_Definition, 802
CAN_TSR_TXOK1
Peripheral_Registers_Bits_Definition, 802
CAN_TSR_TXOK2
Peripheral_Registers_Bits_Definition, 802
CAN_TxMailBox_TypeDef, 1001
CAN_TypeDef, 1002
CCER
CMSIS, 469
TIM1_TypeDef, 1057
CCMDATARAM_BASE
Peripheral_memory_map, 534
CCMDATARAM_BB_BASE
Peripheral_memory_map, 535
CCMR1
CMSIS, 469
TIM1_TypeDef, 1057
CCMR2
CMSIS, 469
TIM1_TypeDef, 1057
CCR
CMSIS, 469, 470
I2C_TypeDef, 1020
ITM Functions, 128
SCB_TypeDef, 1047
CCR1
CMSIS, 470
TIM1_TypeDef, 1057
CCR2
CMSIS, 470
TIM1_TypeDef, 1057
CCR3
CMSIS, 470
TIM1_TypeDef, 1057
CCR4
CMSIS, 470
TIM1_TypeDef, 1058
CDR
CMSIS, 470
CDSR
CMSIS, 470
CFBAR
CMSIS, 470
CFBLNR
CMSIS, 471
CFBLR
CMSIS, 471
CFGGR
CMSIS, 471
CMSIS, 471
RCC_TypeDef, 1035
SYSCFG_RegDef_t, 1054
CFR
CMSIS, 471
CFSR
ITM Functions, 128
SCB_TypeDef, 1047
Check_Flag
API_Definitions, 373
ECU APIs, 363
Check_ID
API_Definitions, 373
ECU APIs, 363
Check_Password
API_Definitions, 373
ECU APIs, 364
CID0
ITM Functions, 128
CID1
ITM Functions, 129
CID2
ITM Functions, 129
CID3
ITM Functions, 129
CIR
CMSIS, 471
RCC_TypeDef, 1035
CKCR
CMSIS, 471
CKGATENR
CMSIS, 471
CLAIMCLR
ITM Functions, 129
CLAIMSET
ITM Functions, 129
CLKCR
CMSIS, 471
Clock Source Selection, 306
HSE_CLK, 306
HSE_VALUE, 306
HSI_RC_CLK, 307
HSI_VALUE, 307
RCC_PLLCFGR_PLLSRC_HSE, 307
RCC_SELECT_HSE, 307
RCC_SELECT_HSI, 307
RCC_SELECT_PLL, 307
clock_config
STK_config_t, 1052
Clock_Phase
S_SPI_Config_t, 1043
Clock_Polarity
S_SPI_Config_t, 1043
CLRFR
CMSIS, 472
CLUTWR
CMSIS, 472
CMD

CMSIS, 472
CMPCR
 CMSIS, 472
 SYSCFG_RegDef_t, 1054
CMSIS, 379
 __CM4_CMSIS_VERSION, 398
 __CM4_CMSIS_VERSION_MAIN, 398
 __CM4_CMSIS_VERSION_SUB, 398
 __CORE_CM4_H_DEPENDANT, 398
 __CORTEX_M, 399
 __I, 399
 __IO, 399
 __O, 399
 ACR, 461
 AFR, 461
 AHB1ENR, 461
 AHB1LPENR, 461
 AHB1RSTR, 461
 AHB2ENR, 461
 AHB2LPENR, 461
 AHB2RSTR, 462
 AHB3ENR, 462
 AHB3LPENR, 462
 AHB3RSTR, 462
 ALRMAR, 462
 ALRMASSR, 462
 ALRMBMR, 462
 ALRMBSSR, 462
 AMTCR, 463
 APB1ENR, 463
 APB1FZ, 463
 APB1LPENR, 463
 APB1RSTR, 463
 APB2ENR, 463
 APB2FZ, 463
 APB2LPENR, 463
 APB2RSTR, 464
 APSR_C_Msk, 399
 APSR_C_Pos, 399
 APSR_GE_Msk, 399
 APSR_GE_Pos, 400
 APSR_N_Msk, 400
 APSR_N_Pos, 400
 APSR_Q_Msk, 400
 APSR_Q_Pos, 400
 APSR_V_Msk, 400
 APSR_V_Pos, 400
 APSR_Z_Msk, 400
 APSR_Z_Pos, 401
 ARG, 464
 ARR, 464
 AWCR, 464
 BCCR, 464
 BDCR, 464
 BDTR, 464
 BFCR, 464
 BGCLUT, 465
 BGCMAR, 465
 BGCOLR, 465
 BGMAR, 465
 BGOR, 465
 BGPFCCR, 465
 BKP0R, 465
 BKP10R, 465
 BKP11R, 466
 BKP12R, 466
 BKP13R, 466
 BKP14R, 466
 BKP15R, 466
 BKP16R, 466
 BKP17R, 466
 BKP18R, 466
 BKP19R, 467
 BKP1R, 467
 BKP2R, 467
 BKP3R, 467
 BKP4R, 467
 BKP5R, 467
 BKP6R, 467
 BKP7R, 467
 BKP8R, 468
 BKP9R, 468
 BPCR, 468
 BRR, 468
 BSRRH, 468
 BSRRL, 468
 BTR, 468
 CACR, 469
 CALIBR, 469
 CALR, 469
 CCER, 469
 CCMR1, 469
 CCMR2, 469
 CCR, 469, 470
 CCR1, 470
 CCR2, 470
 CCR3, 470
 CCR4, 470
 CDR, 470
 CDSR, 470
 CFBAR, 470
 CFBLNR, 471
 CFBLR, 471
 CFGGR, 471
 CFR, 471
 CIR, 471
 CKCR, 471
 CKGATENR, 471
 CLKCR, 471
 CLRFR, 472
 CLUTWR, 472
 CMD, 472
 CMPCR, 472
 CNT, 472
 CONTROL_FPCA_Msk, 401
 CONTROL_FPCA_Pos, 401

CONTROL_nPRIV_Msk, 401
CONTROL_nPRIV_Pos, 401
CONTROL_SPSEL_Msk, 401
CONTROL_SPSEL_Pos, 401
CoreDebug, 401
CoreDebug_BASE, 402
CoreDebug_DCRSR_REGSEL_Msk, 402
CoreDebug_DCRSR_REGSEL_Pos, 402
CoreDebug_DCRSR_REGWnR_Msk, 402
CoreDebug_DCRSR_REGWnR_Pos, 402
CoreDebug_DEMCR_MON_EN_Msk, 402
CoreDebug_DEMCR_MON_EN_Pos, 402
CoreDebug_DEMCR_MON_PEND_Msk, 403
CoreDebug_DEMCR_MON_PEND_Pos, 403
CoreDebug_DEMCR_MON_REQ_Msk, 403
CoreDebug_DEMCR_MON_REQ_Pos, 403
CoreDebug_DEMCR_MON_STEP_Msk, 403
CoreDebug_DEMCR_MON_STEP_Pos, 403
CoreDebug_DEMCR_TRCENA_Msk, 403
CoreDebug_DEMCR_TRCENA_Pos, 403
CoreDebug DEMCR_VC_BUSERR_Msk, 404
CoreDebug DEMCR_VC_BUSERR_Pos, 404
CoreDebug DEMCR_VC_CHKERR_Msk, 404
CoreDebug DEMCR_VC_CHKERR_Pos, 404
CoreDebug DEMCR_VC_CORERESET_Msk, 404
CoreDebug DEMCR_VC_CORERESET_Pos, 404
CoreDebug DEMCR_VC_HARDERR_Msk, 404
CoreDebug DEMCR_VC_HARDERR_Pos, 404
CoreDebug DEMCR_VC_INTERR_Msk, 405
CoreDebug DEMCR_VC_INTERR_Pos, 405
CoreDebug DEMCR_VC_MMERR_Msk, 405
CoreDebug DEMCR_VC_MMERR_Pos, 405
CoreDebug DEMCR_VC_NOCPERR_Msk, 405
CoreDebug DEMCR_VC_NOCPERR_Pos, 405
CoreDebug DEMCR_VC_STATERR_Msk, 405
CoreDebug DEMCR_VC_STATERR_Pos, 405
CoreDebug_DHCSR_C_DEBUGEN_Msk, 406
CoreDebug_DHCSR_C_DEBUGEN_Pos, 406
CoreDebug_DHCSR_C_HALT_Msk, 406
CoreDebug_DHCSR_C_HALT_Pos, 406
CoreDebug_DHCSR_C_MASKINTS_Msk, 406
CoreDebug_DHCSR_C_MASKINTS_Pos, 406
CoreDebug_DHCSR_C_SNAPSTALL_Msk, 406
CoreDebug_DHCSR_C_SNAPSTALL_Pos, 406
CoreDebug_DHCSR_C_STEP_Msk, 407
CoreDebug_DHCSR_C_STEP_Pos, 407
CoreDebug_DHCSR_DBGKEY_Msk, 407
CoreDebug_DHCSR_DBGKEY_Pos, 407
CoreDebug_DHCSR_S_HALT_Msk, 407
CoreDebug_DHCSR_S_HALT_Pos, 407
CoreDebug_DHCSR_S_LOCKUP_Msk, 407
CoreDebug_DHCSR_S_LOCKUP_Pos, 407
CoreDebug_DHCSR_S_REGRDY_Msk, 408
CoreDebug_DHCSR_S_REGRDY_Pos, 408
CoreDebug_DHCSR_S_RESET_ST_Msk, 408
CoreDebug_DHCSR_S_RESET_ST_Pos, 408
CoreDebug_DHCSR_S_RETIRE_ST_Msk, 408
CoreDebug_DHCSR_S_RETIRE_ST_Pos, 408
CoreDebug_DHCSR_S_SLEEP_Msk, 408
CoreDebug_DHCSR_S_SLEEP_Pos, 408
CPSR, 472
CR, 472–474
CR1, 475
CR2, 475, 476
CR3, 476
CRCPR, 476
CSGCM0R, 476
CSGCM1R, 476
CSGCM2R, 477
CSGCM3R, 477
CSGCM4R, 477
CSGCM5R, 477
CSGCM6R, 477
CSGCM7R, 477
CSGCMCCM0R, 477
CSGCMCCM1R, 477
CSGCMCCM2R, 478
CSGCMCCM3R, 478
CSGCMCCM4R, 478
CSGCMCCM5R, 478
CSGCMCCM6R, 478
CSGCMCCM7R, 478
CSR, 478, 479
CWSIZER, 479
CWSTRTR, 479
DCCR, 479
DCKCFG, 479
DCKCFG2, 479
DCOUNT, 479
DCR, 480
DCTRL, 480
DHR12L1, 480
DHR12L2, 480
DHR12LD, 480
DHR12R1, 480
DHR12R2, 480
DHR12RD, 480
DHR8R1, 481
DHR8R2, 481
DHR8RD, 481
DIER, 481
DIN, 481
DLEN, 481
DMACR, 481
DMAR, 482
DOR1, 482
DOR2, 482
DOUT, 482
DR, 482, 483
DTIMER, 483
DWT, 409
DWT_BASE, 409
DWT_CPICNT_CPICNT_Msk, 409
DWT_CPICNT_CPICNT_Pos, 409
DWT_CTRL_CPIEVTEA_Msk, 409

DWT_CTRL_CPIEVTEA_Pos, 409
 DWT_CTRL_CYCCNTENA_Msk, 409
 DWT_CTRL_CYCCNTENA_Pos, 410
 DWT_CTRL_CYCEVTENA_Msk, 410
 DWT_CTRL_CYCEVTENA_Pos, 410
 DWT_CTRL_CYCTAP_Msk, 410
 DWT_CTRL_CYCTAP_Pos, 410
 DWT_CTRL_EXCEVTENA_Msk, 410
 DWT_CTRL_EXCEVTENA_Pos, 410
 DWT_CTRL_EXCTRCENA_Msk, 410
 DWT_CTRL_EXCTRCENA_Pos, 411
 DWT_CTRL_FOLDEVTENA_Msk, 411
 DWT_CTRL_FOLDEVTENA_Pos, 411
 DWT_CTRL_LSUEVTENA_Msk, 411
 DWT_CTRL_LSUEVTENA_Pos, 411
 DWT_CTRL_NOCYCCNT_Msk, 411
 DWT_CTRL_NOCYCCNT_Pos, 411
 DWT_CTRL_NOEXTTRIG_Msk, 411
 DWT_CTRL_NOEXTTRIG_Pos, 412
 DWT_CTRL_NOPRFCNT_Msk, 412
 DWT_CTRL_NOPRFCNT_Pos, 412
 DWT_CTRL_NOTRCPKT_Msk, 412
 DWT_CTRL_NOTRCPKT_Pos, 412
 DWT_CTRL_NUMCOMP_Msk, 412
 DWT_CTRL_NUMCOMP_Pos, 412
 DWT_CTRL_PCSAMPLENA_Msk, 412
 DWT_CTRL_PCSAMPLENA_Pos, 413
 DWT_CTRL_POSTINIT_Msk, 413
 DWT_CTRL_POSTINIT_Pos, 413
 DWT_CTRL_POSTPRESET_Msk, 413
 DWT_CTRL_POSTPRESET_Pos, 413
 DWT_CTRL_SLEEPEVTENA_Msk, 413
 DWT_CTRL_SLEEPEVTENA_Pos, 413
 DWT_CTRL_SYNCTAP_Msk, 413
 DWT_CTRL_SYNCTAP_Pos, 414
 DWT_EXCCNT_EXCCNT_Msk, 414
 DWT_EXCCNT_EXCCNT_Pos, 414
 DWT_FOLDCNT_FOLDCNT_Msk, 414
 DWT_FOLDCNT_FOLDCNT_Pos, 414
 DWT_FUNCTION_CYCMATCH_Msk, 414
 DWT_FUNCTION_CYCMATCH_Pos, 414
 DWT_FUNCTION_DATAVADDR0_Msk, 414
 DWT_FUNCTION_DATAVADDR0_Pos, 415
 DWT_FUNCTION_DATAVADDR1_Msk, 415
 DWT_FUNCTION_DATAVADDR1_Pos, 415
 DWT_FUNCTION_DATAVMATCH_Msk, 415
 DWT_FUNCTION_DATAVMATCH_Pos, 415
 DWT_FUNCTION_DATAVSIZE_Msk, 415
 DWT_FUNCTION_DATAVSIZE_Pos, 415
 DWT_FUNCTION_EMITRANGE_Msk, 415
 DWT_FUNCTION_EMITRANGE_Pos, 416
 DWT_FUNCTION_FUNCTION_Msk, 416
 DWT_FUNCTION_FUNCTION_Pos, 416
 DWT_FUNCTION_LNK1ENA_Msk, 416
 DWT_FUNCTION_LNK1ENA_Pos, 416
 DWT_FUNCTION_MATCHED_Msk, 416
 DWT_FUNCTION_MATCHED_Pos, 416
 DWT_LSUCNT_LSUCNT_Msk, 416
 DWT_LSUCNT_LSUCNT_Pos, 417
 DWT_MASK_MASK_Msk, 417
 DWT_MASK_MASK_Pos, 417
 DWT_SLEEPcnt_SLEEPcnt_Msk, 417
 DWT_SLEEPcnt_SLEEPcnt_Pos, 417
 EGR, 483
 EMR, 484
 ESCR, 484
 ESR, 484
 ESUR, 484
 EXTICR, 484
 FA1R, 484
 FCR, 484
 FFA1R, 485
 FGCLUT, 485
 FGCMAR, 485
 FGCOLR, 485
 FGMAR, 485
 FGOR, 485
 FGPFCCR, 485
 FIFO, 486
 FIFOcnt, 486
 FLTR, 486
 FM1R, 486
 FMR, 486
 FR1, 486
 FR2, 486
 FRCR, 487
 FS1R, 487
 FTSR, 487
 GCR, 487
 GTPR, 487
 HIFCR, 487
 HISR, 488
 HR, 488
 HTR, 488
 I2SCFGR, 488
 I2SPR, 488
 ICR, 488, 489
 IDCODE, 489
 IDR, 489
 IER, 489, 490
 IFCR, 490
 IMR, 490
 IMSCR, 490
 IPSR_ISR_Msk, 417
 IPSR_ISR_Pos, 417
 ISR, 490, 491
 ITM, 417
 ITM_BASE, 418
 ITM_CheckChar, 453
 ITM_IMCR_INTEGRATION_Msk, 418
 ITM_IMCR_INTEGRATION_Pos, 418
 ITM_IRR_ATREADYM_Msk, 418
 ITM_IRR_ATREADYM_Pos, 418
 ITM_IWR_ATVALIDM_Msk, 418
 ITM_IWR_ATVALIDM_Pos, 418
 ITM_LSR_Access_Msk, 419

ITM_LSR_Access_Pos, 419
ITM_LSR_ByteAcc_Msk, 419
ITM_LSR_ByteAcc_Pos, 419
ITM_LSR_Present_Msk, 419
ITM_LSR_Present_Pos, 419
ITM_ReceiveChar, 453
ITM_RxBuffer, 491
ITM_RXBUFFER_EMPTY, 419
ITM_SendChar, 453
ITM_TCR_BUSY_Msk, 419
ITM_TCR_BUSY_Pos, 420
ITM_TCR_DWTENA_Msk, 420
ITM_TCR_DWTENA_Pos, 420
ITM_TCR_GTSFREQ_Msk, 420
ITM_TCR_GTSFREQ_Pos, 420
ITM_TCR_ITMENA_Msk, 420
ITM_TCR_ITMENA_Pos, 420
ITM_TCR_SWOENA_Msk, 420
ITM_TCR_SWOENA_Pos, 421
ITM_TCR_SYNCENA_Msk, 421
ITM_TCR_SYNCENA_Pos, 421
ITM_TCR_TraceBusID_Msk, 421
ITM_TCR_TraceBusID_Pos, 421
ITM_TCR_TSENA_Msk, 421
ITM_TCR_TSENA_Pos, 421
ITM_TCR_TSPrescale_Msk, 421
ITM_TCR_TSPrescale_Pos, 422
ITM_TPR_PRIVMASK_Msk, 422
ITM_TPR_PRIVMASK_Pos, 422
IV0LR, 491
IV0RR, 491
IV1LR, 491
IV1RR, 491
JDR1, 492
JDR2, 492
JDR3, 492
JDR4, 492
JOFR1, 492
JOFR2, 492
JOFR3, 492
JOFR4, 492
JSQR, 493
K0LR, 493
K0RR, 493
K1LR, 493
K1RR, 493
K2LR, 493
K2RR, 493
K3LR, 493
K3RR, 494
KEYR, 494
KR, 494
LCKR, 494
LIFCR, 494
LIPCR, 494
LISR, 494
LTR, 495
LWR, 495
M0AR, 495
M1AR, 495
MASK, 495
MCR, 495
MEMRMP, 495
MISR, 496
MODER, 496
MSR, 496
NDTR, 496
NLR, 496
NVIC, 422
NVIC_BASE, 422
NVIC_ClearPendingIRQ, 454
NVIC_DecodePriority, 454
NVIC_DisableIRQ, 455
NVIC_EnableIRQ, 455
NVIC_EncodePriority, 455
NVIC_GetActive, 457
NVIC_GetPendingIRQ, 457
NVIC_GetPriority, 458
NVIC_GetPriorityGrouping, 458
NVIC_SetPendingIRQ, 458
NVIC_SetPriority, 459
NVIC_SetPriorityGrouping, 459
NVIC_STIR_INTID_Msk, 422
NVIC_STIR_INTID_Pos, 422
NVIC_SystemReset, 459
OAR1, 496
OAR2, 497
OCOLR, 497
ODR, 497
OMAR, 497
OOR, 497
OPFCCR, 497
OPTCR, 497
OPTCR1, 498
OPTKEYR, 498
OR, 498
OSPEEDR, 498
OTYPER, 498
PAR, 498
PFCR, 498
PLLCFGR, 499
PLL2SCFGR, 499
PLLSAICFGR, 499
PMC, 499
POWER, 499
PR, 499
PRER, 500
PSC, 500
PUPDR, 500
RCR, 500
RDHR, 500
RDLR, 500
RDTR, 500
RESERVED, 501
RESERVED0, 501, 502
RESERVED1, 502–504

RESERVED10, 504
 RESERVED11, 504
 RESERVED12, 504
 RESERVED13, 504
 RESERVED14, 504
 RESERVED2, 505
 RESERVED3, 506
 RESERVED4, 507
 RESERVED5, 507, 508
 RESERVED6, 508, 509
 RESERVED7, 509
 RESERVED8, 510
 RESERVED9, 510
 RESP1, 510
 RESP2, 510
 RESP3, 511
 RESP4, 511
 RESPCMD, 511
 RF0R, 511
 RF1R, 511
 RIR, 511
 RISR, 511, 512
 RLR, 512
 RTSR, 512
 RXCRCR, 512
 SCB, 423
 SCB_AIRCR_ENDIANESS_Msk, 423
 SCB_AIRCR_ENDIANESS_Pos, 423
 SCB_AIRCR_PRIGROUP_Msk, 423
 SCB_AIRCR_PRIGROUP_Pos, 423
 SCB_AIRCR_SYSRESETREQ_Msk, 423
 SCB_AIRCR_SYSRESETREQ_Pos, 423
 SCB_AIRCR_VECTCLRACTIVE_Msk, 424
 SCB_AIRCR_VECTCLRACTIVE_Pos, 424
 SCB_AIRCR_VECTKEY_Msk, 424
 SCB_AIRCR_VECTKEY_Pos, 424
 SCB_AIRCR_VECTKEYSTAT_Msk, 424
 SCB_AIRCR_VECTKEYSTAT_Pos, 424
 SCB_AIRCR_VECTRESET_Msk, 424
 SCB_AIRCR_VECTRESET_Pos, 424
 SCB_BASE, 425
 SCB_CCR_BFHFMIGN_Msk, 425
 SCB_CCR_BFHFMIGN_Pos, 425
 SCB_CCR_DIV_0_TRP_Msk, 425
 SCB_CCR_DIV_0_TRP_Pos, 425
 SCB_CCR_NONBASETHRDENA_Msk, 425
 SCB_CCR_NONBASETHRDENA_Pos, 425
 SCB_CCR_STKALIGN_Msk, 426
 SCB_CCR_STKALIGN_Pos, 426
 SCB_CCR_UNALIGN_TRP_Msk, 426
 SCB_CCR_UNALIGN_TRP_Pos, 426
 SCB_CCR_USERSETMPEND_Msk, 426
 SCB_CCR_USERSETMPEND_Pos, 426
 SCB_CFSR_BUSFAULTSR_Msk, 426
 SCB_CFSR_BUSFAULTSR_Pos, 426
 SCB_CFSR_MEMFAULTSR_Msk, 427
 SCB_CFSR_MEMFAULTSR_Pos, 427
 SCB_CFSR_USGFAULTSR_Msk, 427
 SCB_CFSR_USGFAULTSR_Pos, 427
 SCB_CPUID_ARCHITECTURE_Msk, 427
 SCB_CPUID_ARCHITECTURE_Pos, 427
 SCB_CPUID_IMPLEMENTER_Msk, 427
 SCB_CPUID_IMPLEMENTER_Pos, 427
 SCB_CPUID_PARTNO_Msk, 428
 SCB_CPUID_PARTNO_Pos, 428
 SCB_CPUID_REVISION_Msk, 428
 SCB_CPUID_REVISION_Pos, 428
 SCB_CPUID_VARIANT_Msk, 428
 SCB_CPUID_VARIANT_Pos, 428
 SCB_DFSR_BKPT_Msk, 428
 SCB_DFSR_BKPT_Pos, 428
 SCB_DFSR_DWTTRAP_Msk, 429
 SCB_DFSR_DWTTRAP_Pos, 429
 SCB_DFSR_EXTERNAL_Msk, 429
 SCB_DFSR_EXTERNAL_Pos, 429
 SCB_DFSR_HALTED_Msk, 429
 SCB_DFSR_HALTED_Pos, 429
 SCB_DFSR_VCATCH_Msk, 429
 SCB_DFSR_VCATCH_Pos, 429
 SCB_HFSR_DEBUGEVT_Msk, 430
 SCB_HFSR_DEBUGEVT_Pos, 430
 SCB_HFSR_FORCED_Msk, 430
 SCB_HFSR_FORCED_Pos, 430
 SCB_HFSR_VECTTBL_Msk, 430
 SCB_HFSR_VECTTBL_Pos, 430
 SCB_ICSR_ISRPENDING_Msk, 430
 SCB_ICSR_ISRPENDING_Pos, 430
 SCB_ICSR_ISRPREEMPT_Msk, 431
 SCB_ICSR_ISRPREEMPT_Pos, 431
 SCB_ICSR_NMIPENDSET_Msk, 431
 SCB_ICSR_NMIPENDSET_Pos, 431
 SCB_ICSR_PENDSTCLR_Msk, 431
 SCB_ICSR_PENDSTCLR_Pos, 431
 SCB_ICSR_PENDSTSET_Msk, 431
 SCB_ICSR_PENDSTSET_Pos, 431
 SCB_ICSR_PENDSVCLR_Msk, 432
 SCB_ICSR_PENDSVCLR_Pos, 432
 SCB_ICSR_PENDSVSET_Msk, 432
 SCB_ICSR_PENDSVSET_Pos, 432
 SCB_ICSR_RETTOBASE_Msk, 432
 SCB_ICSR_RETTOBASE_Pos, 432
 SCB_ICSR_VECTACTIVE_Msk, 432
 SCB_ICSR_VECTACTIVE_Pos, 432
 SCB_ICSR_VECTPENDING_Msk, 433
 SCB_ICSR_VECTPENDING_Pos, 433
 SCB_SCR_SEVONPEND_Msk, 433
 SCB_SCR_SEVONPEND_Pos, 433
 SCB_SCR_SLEEPDEEP_Msk, 433
 SCB_SCR_SLEEPDEEP_Pos, 433
 SCB_SCR_SLEEPONEEXIT_Msk, 433
 SCB_SCR_SLEEPONEEXIT_Pos, 433
 SCB_SHCSR_BUSFAULTACT_Msk, 434
 SCB_SHCSR_BUSFAULTACT_Pos, 434
 SCB_SHCSR_BUSFAULTENA_Msk, 434
 SCB_SHCSR_BUSFAULTENA_Pos, 434
 SCB_SHCSR_BUSFAULTPENDED_Msk, 434

SCB_SHCSR_BUSFAULTPENDED_Pos, 434
SCB_SHCSR_MEMFAULTACT_Msk, 434
SCB_SHCSR_MEMFAULTACT_Pos, 434
SCB_SHCSR_MEMFAULTENA_Msk, 435
SCB_SHCSR_MEMFAULTENA_Pos, 435
SCB_SHCSR_MEMFAULTPENDED_Msk, 435
SCB_SHCSR_MEMFAULTPENDED_Pos, 435
SCB_SHCSR_MONITORACT_Msk, 435
SCB_SHCSR_MONITORACT_Pos, 435
SCB_SHCSR_PENDSVACT_Msk, 435
SCB_SHCSR_PENDSVACT_Pos, 435
SCB_SHCSR_SVCALLACT_Msk, 436
SCB_SHCSR_SVCALLACT_Pos, 436
SCB_SHCSR_SVCALLPENDED_Msk, 436
SCB_SHCSR_SVCALLPENDED_Pos, 436
SCB_SHCSR_SYSTICKACT_Msk, 436
SCB_SHCSR_SYSTICKACT_Pos, 436
SCB_SHCSR_USGFAULTACT_Msk, 436
SCB_SHCSR_USGFAULTACT_Pos, 436
SCB_SHCSR_USGFAULTENA_Msk, 437
SCB_SHCSR_USGFAULTENA_Pos, 437
SCB_SHCSR_USGFAULTPENDED_Msk, 437
SCB_SHCSR_USGFAULTPENDED_Pos, 437
SCB_VTOR_TBLOFF_Msk, 437
SCB_VTOR_TBLOFF_Pos, 437
SCnSCB, 437
SCnSCB_ACTLR_DISDEFWBUF_Msk, 437
SCnSCB_ACTLR_DISDEFWBUF_Pos, 438
SCnSCB_ACTLR_DISFOLD_Msk, 438
SCnSCB_ACTLR_DISFOLD_Pos, 438
SCnSCB_ACTLR_DISFPCA_Msk, 438
SCnSCB_ACTLR_DISFPCA_Pos, 438
SCnSCB_ACTLR_DISMCYCINT_Msk, 438
SCnSCB_ACTLR_DISMCYCINT_Pos, 438
SCnSCB_ACTLR_DISOOFP_Msk, 438
SCnSCB_ACTLR_DISOOFP_Pos, 439
SCnSCB_ICTR_INTLINESNUM_Msk, 439
SCnSCB_ICTR_INTLINESNUM_Pos, 439
SCS_BASE, 439
sFIFOMailBox, 512
sFilterRegister, 512
SHIFTR, 512
SLOTR, 513
SMCR, 513
SMPR1, 513
SMPR2, 513
SQR1, 513
SQR2, 513
SQR3, 513
SR, 513–515
SR1, 515
SR2, 515
SRCR, 515
SSCGR, 516
SSCR, 516
SSR, 516
STA, 516
STR, 516
sTxMailBox, 516
SWIER, 516
SWTRIGR, 517
SysTick, 439
SysTick_BASE, 439
SysTick_CALIB_NOREF_Msk, 439
SysTick_CALIB_NOREF_Pos, 440
SysTick_CALIB_SKEW_Msk, 440
SysTick_CALIB_SKEW_Pos, 440
SysTick_CALIB_TENMS_Msk, 440
SysTick_CALIB_TENMS_Pos, 440
SysTick_Config, 460
SysTick_CTRL_CLKSOURCE_Msk, 440
SysTick_CTRL_CLKSOURCE_Pos, 440
SysTick_CTRL_COUNTFLAG_Msk, 440
SysTick_CTRL_COUNTFLAG_Pos, 441
SysTick_CTRL_ENABLE_Msk, 441
SysTick_CTRL_ENABLE_Pos, 441
SysTick_CTRL_TICKINT_Msk, 441
SysTick_CTRL_TICKINT_Pos, 441
SysTick_Handler, 460
SysTick_LOAD_RELOAD_Msk, 441
SysTick_LOAD_RELOAD_Pos, 441
SysTick_VAL_CURRENT_Msk, 441
SysTick_VAL_CURRENT_Pos, 442
TAFCR, 517
TDHR, 517
TDLR, 517
TDTR, 517
TIR, 517
TPI, 442
TPI_ACPR_PRESCALER_Msk, 442
TPI_ACPR_PRESCALER_Pos, 442
TPI_BASE, 442
TPI_DEVID_AsynClkIn_Msk, 442
TPI_DEVID_AsynClkIn_Pos, 442
TPI_DEVID_MANCVALID_Msk, 443
TPI_DEVID_MANCVALID_Pos, 443
TPI_DEVID_MinBufSz_Msk, 443
TPI_DEVID_MinBufSz_Pos, 443
TPI_DEVID_NrTraceInput_Msk, 443
TPI_DEVID_NrTraceInput_Pos, 443
TPI_DEVID_NRZVALID_Msk, 443
TPI_DEVID_NRZVALID_Pos, 443
TPI_DEVID_PTINVALID_Msk, 444
TPI_DEVID_PTINVALID_Pos, 444
TPI_DEVTYPE_MajorType_Msk, 444
TPI_DEVTYPE_MajorType_Pos, 444
TPI_DEVTYPE_SubType_Msk, 444
TPI_DEVTYPE_SubType_Pos, 444
TPI_FFCR_EnFCont_Msk, 444
TPI_FFCR_EnFCont_Pos, 444
TPI_FFCR_TrigIn_Msk, 445
TPI_FFCR_TrigIn_Pos, 445
TPI_FFSR_FInProg_Msk, 445
TPI_FFSR_FInProg_Pos, 445
TPI_FFSR_FtNonStop_Msk, 445
TPI_FFSR_FtNonStop_Pos, 445

TPI_FFSR_FtStopped_Msk, 445
TPI_FFSR_FtStopped_Pos, 445
TPI_FFSR_TCPresent_Msk, 446
TPI_FFSR_TCPresent_Pos, 446
TPI_FIFO0_ETM0_Msk, 446
TPI_FIFO0_ETM0_Pos, 446
TPI_FIFO0_ETM1_Msk, 446
TPI_FIFO0_ETM1_Pos, 446
TPI_FIFO0_ETM2_Msk, 446
TPI_FIFO0_ETM2_Pos, 446
TPI_FIFO0_ETM_ATVALID_Msk, 447
TPI_FIFO0_ETM_ATVALID_Pos, 447
TPI_FIFO0_ETM_bytectcount_Msk, 447
TPI_FIFO0_ETM_bytectcount_Pos, 447
TPI_FIFO0_ITM_ATVALID_Msk, 447
TPI_FIFO0_ITM_ATVALID_Pos, 447
TPI_FIFO0_ITM_bytectcount_Msk, 447
TPI_FIFO0_ITM_bytectcount_Pos, 447
TPI_FIFO1_ETM_ATVALID_Msk, 448
TPI_FIFO1_ETM_ATVALID_Pos, 448
TPI_FIFO1_ETM_bytectcount_Msk, 448
TPI_FIFO1_ETM_bytectcount_Pos, 448
TPI_FIFO1_ITM0_Msk, 448
TPI_FIFO1_ITM0_Pos, 448
TPI_FIFO1_ITM1_Msk, 448
TPI_FIFO1_ITM1_Pos, 448
TPI_FIFO1_ITM2_Msk, 449
TPI_FIFO1_ITM2_Pos, 449
TPI_FIFO1_ITM_ATVALID_Msk, 449
TPI_FIFO1_ITM_ATVALID_Pos, 449
TPI_FIFO1_ITM_bytectcount_Msk, 449
TPI_FIFO1_ITM_bytectcount_Pos, 449
TPI_ITATBCTR0_ATREADY_Msk, 449
TPI_ITATBCTR0_ATREADY_Pos, 449
TPI_ITATBCTR2_ATREADY_Msk, 450
TPI_ITATBCTR2_ATREADY_Pos, 450
TPI_ITCTRL_Mode_Msk, 450
TPI_ITCTRL_Mode_Pos, 450
TPI_SPPR_TXMODE_Msk, 450
TPI_SPPR_TXMODE_Pos, 450
TPI_TRIGGER_TRIGGER_Msk, 450
TPI_TRIGGER_TRIGGER_Pos, 450
TR, 517
TRISE, 517
TSDR, 518
TSR, 518
TSSSR, 518
TSTR, 518
TWCR, 518
TXCRCR, 518
WHPCR, 518
WPR, 519
WUTR, 519
WVPCR, 519
xPSR_C_Msk, 451
xPSR_C_Pos, 451
xPSR_GE_Msk, 451
xPSR_GE_Pos, 451
xPSR_ISR_Msk, 451
xPSR_ISR_Pos, 451
xPSR_IT_Msk, 451
xPSR_IT_Pos, 451
xPSR_N_Msk, 452
xPSR_N_Pos, 452
xPSR_Q_Msk, 452
xPSR_Q_Pos, 452
xPSR_T_Msk, 452
xPSR_T_Pos, 452
xPSR_V_Msk, 452
xPSR_V_Pos, 452
xPSR_Z_Msk, 453
xPSR_Z_Pos, 453
CMSIS Core Instruction Interface, 148
CMSIS Core Register Access Functions, 147
CMSIS Global Defines, 42
CMSIS SIMD Intrinsics, 148
CMSIS/Inc/core_cm4.h, 1079
CMSIS/Inc/core_cmFunc.h, 1089
CMSIS/Inc/core_cmlInstr.h, 1090
CMSIS/Inc/core_cmSimd.h, 1091
CMSIS/Inc/stm32f4xx.h, 1092
CMSIS/Inc/system_stm32f4xx.h, 1184
CNT
 CMSIS, 472
 TIM1_TypeDef, 1058
combineArrays
 API_Definitions, 374
 ECU APIs, 364
Communication_Mode
 S_SPI_Config_t, 1044
COMP0
 ITM Functions, 129
COMP1
 ITM Functions, 129
COMP2
 ITM Functions, 130
COMP3
 ITM Functions, 130
Configuration Macros, 360
Configuration_section_for\CMSIS, 522
 __CM4_REV, 523
 __FPU_PRESENT, 523
 __MPU_PRESENT, 523
 __NVIC_PRIO_BITS, 524
 Vendor_SysTickConfig, 524
ADC IRQn, 525
BusFault IRQn, 524
DebugMonitor IRQn, 524
DMA1_Stream0 IRQn, 525
DMA1_Stream1 IRQn, 525
DMA1_Stream2 IRQn, 525
DMA1_Stream3 IRQn, 525
DMA1_Stream4 IRQn, 525
DMA1_Stream5 IRQn, 525
DMA1_Stream6 IRQn, 525
DMA1_Stream7 IRQn, 526

DMA2_Stream0_IRQHandler, 526
DMA2_Stream1_IRQHandler, 526
DMA2_Stream2_IRQHandler, 526
DMA2_Stream3_IRQHandler, 526
DMA2_Stream4_IRQHandler, 526
DMA2_Stream5_IRQHandler, 526
DMA2_Stream6_IRQHandler, 526
DMA2_Stream7_IRQHandler, 526
EXTI0_IRQHandler, 525
EXTI15_10_IRQHandler, 526
EXTI1_IRQHandler, 525
EXTI2_IRQHandler, 525
EXTI3_IRQHandler, 525
EXTI4_IRQHandler, 525
EXTI9_5_IRQHandler, 525
FLASH_IRQHandler, 525
FPU_IRQHandler, 526
I2C1_ER_IRQHandler, 525
I2C1_EV_IRQHandler, 525
I2C2_ER_IRQHandler, 526
I2C2_EV_IRQHandler, 526
I2C3_ER_IRQHandler, 526
I2C3_EV_IRQHandler, 526
IRQn, 524
MemoryManagement_IRQHandler, 524
NonMaskableInt_IRQHandler, 524
OTG_FS_IRQHandler, 526
OTG_FS_WKUP_IRQHandler, 526
PendSV_IRQHandler, 524
PVD_IRQHandler, 525
RCC_IRQHandler, 525
RTC_Alarm_IRQHandler, 526
RTC_WKUP_IRQHandler, 525
SDIO_IRQHandler, 526
SPI1_IRQHandler, 526
SPI2_IRQHandler, 526
SPI3_IRQHandler, 526
SPI4_IRQHandler, 526
SVCall_IRQHandler, 524
SysTick_IRQHandler, 524
TAMP_STAMP_IRQHandler, 525
TIM1_BRK_TIM9_IRQHandler, 525
TIM1_CC_IRQHandler, 525
TIM1_TRG_COM_TIM11_IRQHandler, 525
TIM1_UP_TIM10_IRQHandler, 525
TIM2_IRQHandler, 525
TIM3_IRQHandler, 525
TIM4_IRQHandler, 525
TIM5_IRQHandler, 526
UsageFault_IRQHandler, 524
USART1_IRQHandler, 526
USART2_IRQHandler, 526
USART6_IRQHandler, 526
WWDG_IRQHandler, 524
ConstVoidPtr
 Platform Standard Types, 152
CONTROL_FPCA_Msk
 CMSIS, 401
Status and Control Registers, 46
CONTROL_FPCA_Pos
 CMSIS, 401
 Status and Control Registers, 46
CONTROL_nPRIV_Msk
 CMSIS, 401
 Status and Control Registers, 46
CONTROL_nPRIV_Pos
 CMSIS, 401
 Status and Control Registers, 46
CONTROL_SPSEL_Msk
 CMSIS, 401
 Status and Control Registers, 46
CONTROL_SPSEL_Pos
 CMSIS, 401
 Status and Control Registers, 46
CONTROL_Type, 1003
Core Debug Registers (CoreDebug), 99
 CoreDebug_DCRSR_REGSEL_Msk, 101
 CoreDebug_DCRSR_REGSEL_Pos, 101
 CoreDebug_DCRSR_REGWnR_Msk, 101
 CoreDebug_DCRSR_REGWnR_Pos, 101
 CoreDebug_DEMCR_MON_EN_Msk, 101
 CoreDebug_DEMCR_MON_EN_Pos, 102
 CoreDebug_DEMCR_MON_PEND_Msk, 102
 CoreDebug_DEMCR_MON_PEND_Pos, 102
 CoreDebug_DEMCR_MON_REQ_Msk, 102
 CoreDebug_DEMCR_MON_REQ_Pos, 102
 CoreDebug_DEMCR_MON_STEP_Msk, 102
 CoreDebug_DEMCR_MON_STEP_Pos, 102
 CoreDebug_DEMCR_TRCENA_Msk, 102
 CoreDebug_DEMCR_TRCENA_Pos, 103
 CoreDebug_DEMCR_VC_BUSERR_Msk, 103
 CoreDebug_DEMCR_VC_BUSERR_Pos, 103
 CoreDebug_DEMCR_VC_CHKERR_Msk, 103
 CoreDebug_DEMCR_VC_CHKERR_Pos, 103
 CoreDebug_DEMCR_VC_CORERESET_Msk, 103
 CoreDebug_DEMCR_VC_CORERESET_Pos, 103
 CoreDebug_DEMCR_VC_HARDERR_Msk, 103
 CoreDebug_DEMCR_VC_HARDERR_Pos, 104
 CoreDebug_DEMCR_VC_INTERR_Msk, 104
 CoreDebug_DEMCR_VC_INTERR_Pos, 104
 CoreDebug_DEMCR_VC_MMERR_Msk, 104
 CoreDebug_DEMCR_VC_MMERR_Pos, 104
 CoreDebug_DEMCR_VC_NOCPERR_Msk, 104
 CoreDebug_DEMCR_VC_NOCPERR_Pos, 104
 CoreDebug_DEMCR_VC_STATERR_Msk, 104
 CoreDebug_DEMCR_VC_STATERR_Pos, 105
 CoreDebug_DHCSR_C_DEBUGEN_Msk, 105
 CoreDebug_DHCSR_C_DEBUGEN_Pos, 105
 CoreDebug_DHCSR_C_HALT_Msk, 105
 CoreDebug_DHCSR_C_HALT_Pos, 105
 CoreDebug_DHCSR_C_MASKINTS_Msk, 105
 CoreDebug_DHCSR_C_MASKINTS_Pos, 105
 CoreDebug_DHCSR_C_SNAPSTALL_Msk, 105
 CoreDebug_DHCSR_C_SNAPSTALL_Pos, 106
 CoreDebug_DHCSR_C_STEP_Msk, 106

CoreDebug_DHCSR_C_STEP_Pos, 106
CoreDebug_DHCSR_DBGKEY_Msk, 106
CoreDebug_DHCSR_DBGKEY_Pos, 106
CoreDebug_DHCSR_S_HALT_Msk, 106
CoreDebug_DHCSR_S_HALT_Pos, 106
CoreDebug_DHCSR_S_LOCKUP_Msk, 106
CoreDebug_DHCSR_S_LOCKUP_Pos, 107
CoreDebug_DHCSR_S_REGRDY_Msk, 107
CoreDebug_DHCSR_S_REGRDY_Pos, 107
CoreDebug_DHCSR_S_RESET_ST_Msk, 107
CoreDebug_DHCSR_S_RESET_ST_Pos, 107
CoreDebug_DHCSR_S_RETIRE_ST_Msk, 107
CoreDebug_DHCSR_S_RETIRE_ST_Pos, 107
CoreDebug_DHCSR_S_SLEEP_Msk, 107
CoreDebug_DHCSR_S_SLEEP_Pos, 108
Core Definitions, 108
CoreDebug, 109
CoreDebug_BASE, 109
DWT, 109
DWT_BASE, 109
ITM, 109
ITM_BASE, 109
NVIC, 110
NVIC_BASE, 110
SCB, 110
SCB_BASE, 110
SCnSCB, 110
SCS_BASE, 110
SysTick, 110
SysTick_BASE, 111
TPI, 111
TPI_BASE, 111
CoreDebug
 CMSIS, 401
 Core Definitions, 109
CoreDebug_BASE
 CMSIS, 402
 Core Definitions, 109
CoreDebug_DCRSR_REGSEL_Msk
 CMSIS, 402
 Core Debug Registers (CoreDebug), 101
CoreDebug_DCRSR_REGSEL_Pos
 CMSIS, 402
 Core Debug Registers (CoreDebug), 101
CoreDebug_DCRSR_REGWnR_Msk
 CMSIS, 402
 Core Debug Registers (CoreDebug), 101
CoreDebug_DCRSR_REGWnR_Pos
 CMSIS, 402
 Core Debug Registers (CoreDebug), 101
CoreDebug_DEMCR_MON_EN_Msk
 CMSIS, 402
 Core Debug Registers (CoreDebug), 101
CoreDebug_DEMCR_MON_EN_Pos
 CMSIS, 402
 Core Debug Registers (CoreDebug), 102
CoreDebug_DEMCR_MON_PEND_Msk
 CMSIS, 403
Core Debug Registers (CoreDebug), 102
CoreDebug_DEMCR_MON_PEND_Pos
 CMSIS, 403
 Core Debug Registers (CoreDebug), 102
CoreDebug_DEMCR_MON_REQ_Msk
 CMSIS, 403
 Core Debug Registers (CoreDebug), 102
CoreDebug_DEMCR_MON_REQ_Pos
 CMSIS, 403
 Core Debug Registers (CoreDebug), 102
CoreDebug_DEMCR_MON_STEP_Msk
 CMSIS, 403
 Core Debug Registers (CoreDebug), 102
CoreDebug_DEMCR_MON_STEP_Pos
 CMSIS, 403
 Core Debug Registers (CoreDebug), 102
CoreDebug_DEMCR_TRCENA_Msk
 CMSIS, 403
 Core Debug Registers (CoreDebug), 102
CoreDebug_DEMCR_TRCENA_Pos
 CMSIS, 403
 Core Debug Registers (CoreDebug), 103
CoreDebug_DEMCR_VC_BUSERR_Msk
 CMSIS, 404
 Core Debug Registers (CoreDebug), 103
CoreDebug_DEMCR_VC_BUSERR_Pos
 CMSIS, 404
 Core Debug Registers (CoreDebug), 103
CoreDebug_DEMCR_VC_CHKERR_Msk
 CMSIS, 404
 Core Debug Registers (CoreDebug), 103
CoreDebug_DEMCR_VC_CHKERR_Pos
 CMSIS, 404
 Core Debug Registers (CoreDebug), 103
CoreDebug_DEMCR_VC_CORERESET_Msk
 CMSIS, 404
 Core Debug Registers (CoreDebug), 103
CoreDebug_DEMCR_VC_CORERESET_Pos
 CMSIS, 404
 Core Debug Registers (CoreDebug), 103
CoreDebug_DEMCR_VC_HARDERR_Msk
 CMSIS, 404
 Core Debug Registers (CoreDebug), 103
CoreDebug_DEMCR_VC_HARDERR_Pos
 CMSIS, 404
 Core Debug Registers (CoreDebug), 104
CoreDebug_DEMCR_VC_INTERR_Msk
 CMSIS, 405
 Core Debug Registers (CoreDebug), 104
CoreDebug_DEMCR_VC_INTERR_Pos
 CMSIS, 405
 Core Debug Registers (CoreDebug), 104
CoreDebug_DEMCR_VC_MMERR_Msk
 CMSIS, 405
 Core Debug Registers (CoreDebug), 104
CoreDebug_DEMCR_VC_MMERR_Pos
 CMSIS, 405
 Core Debug Registers (CoreDebug), 104

CoreDebug_DEMCR_VC_NOCPERR_Msk
 CMSIS, 405
 Core Debug Registers (CoreDebug), 104

CoreDebug_DEMCR_VC_NOCPERR_Pos
 CMSIS, 405
 Core Debug Registers (CoreDebug), 104

CoreDebug_DEMCR_VC_STATERR_Msk
 CMSIS, 405
 Core Debug Registers (CoreDebug), 104

CoreDebug_DEMCR_VC_STATERR_Pos
 CMSIS, 405
 Core Debug Registers (CoreDebug), 105

CoreDebug_DHCSR_C_DEBUGEN_Msk
 CMSIS, 406
 Core Debug Registers (CoreDebug), 105

CoreDebug_DHCSR_C_DEBUGEN_Pos
 CMSIS, 406
 Core Debug Registers (CoreDebug), 105

CoreDebug_DHCSR_C_HALT_Msk
 CMSIS, 406
 Core Debug Registers (CoreDebug), 105

CoreDebug_DHCSR_C_HALT_Pos
 CMSIS, 406
 Core Debug Registers (CoreDebug), 105

CoreDebug_DHCSR_C_MASKINTS_Msk
 CMSIS, 406
 Core Debug Registers (CoreDebug), 105

CoreDebug_DHCSR_C_MASKINTS_Pos
 CMSIS, 406
 Core Debug Registers (CoreDebug), 105

CoreDebug_DHCSR_C_SNAPSTALL_Msk
 CMSIS, 406
 Core Debug Registers (CoreDebug), 105

CoreDebug_DHCSR_C_SNAPSTALL_Pos
 CMSIS, 406
 Core Debug Registers (CoreDebug), 106

CoreDebug_DHCSR_C_STEP_Msk
 CMSIS, 407
 Core Debug Registers (CoreDebug), 106

CoreDebug_DHCSR_C_STEP_Pos
 CMSIS, 407
 Core Debug Registers (CoreDebug), 106

CoreDebug_DHCSR_DBGKEY_Msk
 CMSIS, 407
 Core Debug Registers (CoreDebug), 106

CoreDebug_DHCSR_DBGKEY_Pos
 CMSIS, 407
 Core Debug Registers (CoreDebug), 106

CoreDebug_DHCSR_S_HALT_Msk
 CMSIS, 407
 Core Debug Registers (CoreDebug), 106

CoreDebug_DHCSR_S_HALT_Pos
 CMSIS, 407
 Core Debug Registers (CoreDebug), 106

CoreDebug_DHCSR_S_LOCKUP_Msk
 CMSIS, 407
 Core Debug Registers (CoreDebug), 106

CoreDebug_DHCSR_S_LOCKUP_Pos
 CMSIS, 407
 Core Debug Registers (CoreDebug), 106

CMSIS, 407
Core Debug Registers (CoreDebug), 107

CoreDebug_DHCSR_S_REGRDY_Msk
 CMSIS, 408
 Core Debug Registers (CoreDebug), 107

CoreDebug_DHCSR_S_REGRDY_Pos
 CMSIS, 408
 Core Debug Registers (CoreDebug), 107

CoreDebug_DHCSR_S_RESET_ST_Msk
 CMSIS, 408
 Core Debug Registers (CoreDebug), 107

CoreDebug_DHCSR_S_RESET_ST_Pos
 CMSIS, 408
 Core Debug Registers (CoreDebug), 107

CoreDebug_DHCSR_S_RETIRE_ST_Msk
 CMSIS, 408
 Core Debug Registers (CoreDebug), 107

CoreDebug_DHCSR_S_RETIRE_ST_Pos
 CMSIS, 408
 Core Debug Registers (CoreDebug), 107

CoreDebug_DHCSR_S_SLEEP_Msk
 CMSIS, 408
 Core Debug Registers (CoreDebug), 107

CoreDebug_DHCSR_S_SLEEP_Pos
 CMSIS, 408
 Core Debug Registers (CoreDebug), 108

CoreDebug_Type, 1003

Cortex-M4 Peripherals Base Addresses, 157

CPACR
 ITM Functions, 130

CPICNT
 ITM Functions, 130

CPSR
 CMSIS, 472

CPU_BIT_ORDER
 Platform Standard Types, 150

CPU_BYTE_ORDER
 Platform Standard Types, 150

CPU_TYPE
 Platform Standard Types, 150

CPU_TYPE_16
 Platform Standard Types, 150

CPU_TYPE_32
 Platform Standard Types, 150

CPU_TYPE_64
 Platform Standard Types, 150

CPU_TYPE_8
 Platform Standard Types, 150

CPUID
 ITM Functions, 130
 SCB_TypeDef, 1047

CR
 CMSIS, 472–474
 CRC_TypeDef, 1004
 RCC_TypeDef, 1035

CR1
 CMSIS, 475
 I2C_TypeDef, 1020

SPI_TypeDef, 1050
 TIM1_TypeDef, 1058
 USART_TypeDef, 1063
CR2
 CMSIS, 475, 476
 I2C_TypeDef, 1020
 SPI_TypeDef, 1050
 TIM1_TypeDef, 1058
 USART_TypeDef, 1063
CR3
 CMSIS, 476
 USART_TypeDef, 1063
CRC
 CRC Instance, 178
CRC Instance, 178
 CRC, 178
CRC Register Map, 166
CRC_CR_RESET
 Peripheral_Registers_Bits_Definition, 802
CRC_DR_DR
 Peripheral_Registers_Bits_Definition, 802
CRC_IDR_IDR
 Peripheral_Registers_Bits_Definition, 802
CRC_TypeDef, 1004
 CR, 1004
 DR, 1004
 IDR, 1005
 INIT, 1005
 POL, 1005
 RESERVED, 1005
 SR, 1005
CRCPR
 CMSIS, 476
 SPI_TypeDef, 1051
CRYP_TypeDef, 1005
CSGCM0R
 CMSIS, 476
CSGCM1R
 CMSIS, 476
CSGCM2R
 CMSIS, 477
CSGCM3R
 CMSIS, 477
CSGCM4R
 CMSIS, 477
CSGCM5R
 CMSIS, 477
CSGCM6R
 CMSIS, 477
CSGCM7R
 CMSIS, 477
CSGCMCCM0R
 CMSIS, 477
CSGCMCCM1R
 CMSIS, 477
CSGCMCCM2R
 CMSIS, 478
CSGCMCCM3R
 CMSIS, 478
 CMSIS, 478
 CSGCMCCM4R
 CMSIS, 478
 CSGCMCCM5R
 CMSIS, 478
 CSGCMCCM6R
 CMSIS, 478
 CSGCMCCM7R
 CMSIS, 478
CSPSR
 ITM Functions, 130
CSR
 CMSIS, 478, 479
 RCC_TypeDef, 1035
CTRL
 ITM Functions, 130, 131
 STK_TypeDef, 1053
CWSIZER
 CMSIS, 479
CWSTRTR
 CMSIS, 479
CYCCNT
 ITM Functions, 131
D0_PIN
 LCD_t, 1024
D1_PIN
 LCD_t, 1024
D2_PIN
 LCD_t, 1024
D3_PIN
 LCD_t, 1024
D4_PIN
 LCD_t, 1025
D5_PIN
 LCD_t, 1025
D6_PIN
 LCD_t, 1025
D7_PIN
 LCD_t, 1025
DAC_CR_BOFF1
 Peripheral_Registers_Bits_Definition, 802
DAC_CR_BOFF2
 Peripheral_Registers_Bits_Definition, 802
DAC_CR_DMAEN1
 Peripheral_Registers_Bits_Definition, 803
DAC_CR_DMAEN2
 Peripheral_Registers_Bits_Definition, 803
DAC_CR_DMAUDRIE1
 Peripheral_Registers_Bits_Definition, 803
DAC_CR_DMAUDRIE2
 Peripheral_Registers_Bits_Definition, 803
DAC_CR_EN1
 Peripheral_Registers_Bits_Definition, 803
DAC_CR_EN2
 Peripheral_Registers_Bits_Definition, 803
DAC_CR_MAMP1
 Peripheral_Registers_Bits_Definition, 803
DAC_CR_MAMP1_0

- Peripheral_Registers_Bits_Definition, 803
DAC_CR_MAMP1_1
 Peripheral_Registers_Bits_Definition, 804
DAC_CR_MAMP1_2
 Peripheral_Registers_Bits_Definition, 804
DAC_CR_MAMP1_3
 Peripheral_Registers_Bits_Definition, 804
DAC_CR_MAMP2

DWT_CTRL_PCSAMPLENA_Pos, 84
DWT_CTRL_POSTINIT_Msk, 84
DWT_CTRL_POSTINIT_Pos, 84
DWT_CTRL_POSTPRESET_Msk, 84
DWT_CTRL_POSTPRESET_Pos, 85
DWT_CTRL_SLEEPEVTENA_Msk, 85
DWT_CTRL_SLEEPEVTENA_Pos, 85
DWT_CTRL_SYNCTAP_Msk, 85
DWT_CTRL_SYNCTAP_Pos, 85
DWT_EXCCNT_EXCCNT_Msk, 85
DWT_EXCCNT_EXCCNT_Pos, 85
DWT_FOLDCNT_FOLDCNT_Msk, 85
DWT_FOLDCNT_FOLDCNT_Pos, 86
DWT_FUNCTION_CYCMATCH_Msk, 86
DWT_FUNCTION_CYCMATCH_Pos, 86
DWT_FUNCTION_DATAVADDR0_Msk, 86
DWT_FUNCTION_DATAVADDR0_Pos, 86
DWT_FUNCTION_DATAVADDR1_Msk, 86
DWT_FUNCTION_DATAVADDR1_Pos, 86
DWT_FUNCTION_DATAVMATCH_Msk, 86
DWT_FUNCTION_DATAVMATCH_Pos, 87
DWT_FUNCTION_DATAVSIZE_Msk, 87
DWT_FUNCTION_DATAVSIZE_Pos, 87
DWT_FUNCTION_EMITRANGE_Msk, 87
DWT_FUNCTION_EMITRANGE_Pos, 87
DWT_FUNCTION_FUNCTION_Msk, 87
DWT_FUNCTION_FUNCTION_Pos, 87
DWT_FUNCTION_LNK1ENA_Msk, 87
DWT_FUNCTION_LNK1ENA_Pos, 88
DWT_FUNCTION_MATCHED_Msk, 88
DWT_FUNCTION_MATCHED_Pos, 88
DWT_LSUCNT_LSUCNT_Msk, 88
DWT_LSUCNT_LSUCNT_Pos, 88
DWT_MASK_MASK_Msk, 88
DWT_MASK_MASK_Pos, 88
DWT_SLEEPCNT_SLEEPCNT_Msk, 88
DWT_SLEEPCNT_SLEEPCNT_Pos, 89
DBGMCU_CR_TRACE_MODE_0
Peripheral_Registers_Bits_Definition, 809
DBGMCU_CR_TRACE_MODE_1
Peripheral_Registers_Bits_Definition, 809
DBGMCU_TypeDef, 1007
DCCR
CMSIS, 479
DCKCFGR
CMSIS, 479
RCC_TypeDef, 1035
DCKCFGR2
CMSIS, 479
DCMI_TypeDef, 1008
DCOUNT
CMSIS, 479
DCR
CMSIS, 480
TIM1_TypeDef, 1058
DCRDR
ITM Functions, 131
DCRSR

ITM Functions, 131
DCTRL
CMSIS, 480
DebugMonitor_IRQHandler
Configuration_section_for\CMSIS, 524
Defines and Type Definitions, 42
DEMCR
ITM Functions, 131
Device_Mode
S_SPI_Config_t, 1044
DEVID
ITM Functions, 131
DEVTYPE
ITM Functions, 131
DFR
ITM Functions, 132
DFSDM_CHAWSCDR_AWFORD
Peripheral_Registers_Bits_Definition, 809
DFSDM_CHAWSCDR_AWFORD_0
Peripheral_Registers_Bits_Definition, 809
DFSDM_CHAWSCDR_AWFORD_1
Peripheral_Registers_Bits_Definition, 809
DFSDM_CHAWSCDR_AWFOSR
Peripheral_Registers_Bits_Definition, 809
DFSDM_CHAWSCDR_BKSCD
Peripheral_Registers_Bits_Definition, 810
DFSDM_CHAWSCDR_SCDT
Peripheral_Registers_Bits_Definition, 810
DFSDM_CHCFGRI_CHEN
Peripheral_Registers_Bits_Definition, 810
DFSDM_CHCFGRI_CHINSEL
Peripheral_Registers_Bits_Definition, 810
DFSDM_CHCFGRI_CKABEN
Peripheral_Registers_Bits_Definition, 810
DFSDM_CHCFGRI_CKOUTDIV
Peripheral_Registers_Bits_Definition, 810
DFSDM_CHCFGRI_CKOUTSRC
Peripheral_Registers_Bits_Definition, 810
DFSDM_CHCFGRI_DATMPX
Peripheral_Registers_Bits_Definition, 810
DFSDM_CHCFGRI_DATMPX_0
Peripheral_Registers_Bits_Definition, 811
DFSDM_CHCFGRI_DATMPX_1
Peripheral_Registers_Bits_Definition, 811
DFSDM_CHCFGRI_DATPACK
Peripheral_Registers_Bits_Definition, 811
DFSDM_CHCFGRI_DATPACK_0
Peripheral_Registers_Bits_Definition, 811
DFSDM_CHCFGRI_DATPACK_1
Peripheral_Registers_Bits_Definition, 811
DFSDM_CHCFGRI_DFSDMEN
Peripheral_Registers_Bits_Definition, 811
DFSDM_CHCFGRI_SCDEN
Peripheral_Registers_Bits_Definition, 811
DFSDM_CHCFGRI_SITP
Peripheral_Registers_Bits_Definition, 811
DFSDM_CHCFGRI_SITP_0
Peripheral_Registers_Bits_Definition, 812

DFSDM_CHCFG1_SITP_1
 Peripheral_Registers_Bits_Definition, 812
DFSDM_CHCFG1_SPICKSEL
 Peripheral_Registers_Bits_Definition, 812
DFSDM_CHCFG1_SPICKSEL_0
 Peripheral_Registers_Bits_Definition, 812
DFSDM_CHCFG1_SPICKSEL_1
 Peripheral_Registers_Bits_Definition, 812
DFSDM_CHCFG2_DTRBS
 Peripheral_Registers_Bits_Definition, 812
DFSDM_CHCFG2_OFFSET
 Peripheral_Registers_Bits_Definition, 812
DFSDM_CHDATINR_INDAT0
 Peripheral_Registers_Bits_Definition, 812
DFSDM_CHDATINR_INDAT1
 Peripheral_Registers_Bits_Definition, 813
DFSDM_CHWDATR_WDATA
 Peripheral_Registers_Bits_Definition, 813
DFSDM_FLTAWCFR_CLRAWHTF
 Peripheral_Registers_Bits_Definition, 813
DFSDM_FLTAWCFR_CLRAWLTF
 Peripheral_Registers_Bits_Definition, 813
DFSDM_FLTAWHTR_AWHT
 Peripheral_Registers_Bits_Definition, 813
DFSDM_FLTAWHTR_BKAWH
 Peripheral_Registers_Bits_Definition, 813
DFSDM_FLTAWLTR_AWLT
 Peripheral_Registers_Bits_Definition, 813
DFSDM_FLTAWLTR_BKAWL
 Peripheral_Registers_Bits_Definition, 813
DFSDM_FLTAWSR_AWHTF
 Peripheral_Registers_Bits_Definition, 814
DFSDM_FLTAWSR_AWLTF
 Peripheral_Registers_Bits_Definition, 814
DFSDM_FLTCR1_AWFSEL
 Peripheral_Registers_Bits_Definition, 814
DFSDM_FLTCR1_DFEN
 Peripheral_Registers_Bits_Definition, 814
DFSDM_FLTCR1_FAST
 Peripheral_Registers_Bits_Definition, 814
DFSDM_FLTCR1_JDMAEN
 Peripheral_Registers_Bits_Definition, 814
DFSDM_FLTCR1_JEXTEN
 Peripheral_Registers_Bits_Definition, 814
DFSDM_FLTCR1_JEXTEN_0
 Peripheral_Registers_Bits_Definition, 815
DFSDM_FLTCR1_JEXTEN_1
 Peripheral_Registers_Bits_Definition, 815
DFSDM_FLTCR1_JEXTSEL
 Peripheral_Registers_Bits_Definition, 815
DFSDM_FLTCR1_JEXTSEL_0
 Peripheral_Registers_Bits_Definition, 815
DFSDM_FLTCR1_JEXTSEL_1
 Peripheral_Registers_Bits_Definition, 815
DFSDM_FLTCR1_JEXTSEL_2
 Peripheral_Registers_Bits_Definition, 815
DFSDM_FLTCR1_JSCAN
 Peripheral_Registers_Bits_Definition, 815
DFSDM_FLTCR1_JSSTART
 Peripheral_Registers_Bits_Definition, 815
DFSDM_FLTCR1_JSYNC
 Peripheral_Registers_Bits_Definition, 816
DFSDM_FLTCR1_RCH
 Peripheral_Registers_Bits_Definition, 816
DFSDM_FLTCR1_RCONT
 Peripheral_Registers_Bits_Definition, 816
DFSDM_FLTCR1_RDMAEN
 Peripheral_Registers_Bits_Definition, 816
DFSDM_FLTCR1_RSWSTART
 Peripheral_Registers_Bits_Definition, 816
DFSDM_FLTCR1_RSYNC
 Peripheral_Registers_Bits_Definition, 816
DFSDM_FLTCR2_AWDCH
 Peripheral_Registers_Bits_Definition, 816
DFSDM_FLTCR2_AWDIE
 Peripheral_Registers_Bits_Definition, 817
DFSDM_FLTCR2_CKABIE
 Peripheral_Registers_Bits_Definition, 817
DFSDM_FLTCR2_EXCH
 Peripheral_Registers_Bits_Definition, 817
DFSDM_FLTCR2_JEOCIE
 Peripheral_Registers_Bits_Definition, 817
DFSDM_FLTCR2_JOVRIE
 Peripheral_Registers_Bits_Definition, 817
DFSDM_FLTCR2_REOCIE
 Peripheral_Registers_Bits_Definition, 817
DFSDM_FLTCR2_ROVRIE
 Peripheral_Registers_Bits_Definition, 817
DFSDM_FLTCR2_SCDIE
 Peripheral_Registers_Bits_Definition, 817
DFSDM_FLTEXMAX_EXMAX
 Peripheral_Registers_Bits_Definition, 818
DFSDM_FLTEXMAX_EXMAXCH
 Peripheral_Registers_Bits_Definition, 818
DFSDM_FLTEXMIN_EXMIN
 Peripheral_Registers_Bits_Definition, 818
DFSDM_FLTEXMIN_EXMINCH
 Peripheral_Registers_Bits_Definition, 818
DFSDM_FLTFCR_FORD
 Peripheral_Registers_Bits_Definition, 818
DFSDM_FLTFCR_FORD_0
 Peripheral_Registers_Bits_Definition, 818
DFSDM_FLTFCR_FORD_1
 Peripheral_Registers_Bits_Definition, 818
DFSDM_FLTFCR_FORD_2
 Peripheral_Registers_Bits_Definition, 818
DFSDM_FLTFCR_FOSR
 Peripheral_Registers_Bits_Definition, 819
DFSDM_FLTFCR_IOSR
 Peripheral_Registers_Bits_Definition, 819
DFSDM_FLTICR_CLRCKABF
 Peripheral_Registers_Bits_Definition, 819
DFSDM_FLTICR_CLRJOVRF
 Peripheral_Registers_Bits_Definition, 819

DFSDM_FLTICR_CLRROVRF
 Peripheral_Registers_Bits_Definition, 819

DFSDM_FLTICR_CLRSCDF
 Peripheral_Registers_Bits_Definition, 819

DFSDM_FLTISR_AWDF
 Peripheral_Registers_Bits_Definition, 819

DFSDM_FLTISR_CKABF
 Peripheral_Registers_Bits_Definition, 819

DFSDM_FLTISR_JCIP
 Peripheral_Registers_Bits_Definition, 820

DFSDM_FLTISR_JEOCF
 Peripheral_Registers_Bits_Definition, 820

DFSDM_FLTISR_JOVRF
 Peripheral_Registers_Bits_Definition, 820

DFSDM_FLTISR_RCIP
 Peripheral_Registers_Bits_Definition, 820

DFSDM_FLTISR_REOCF
 Peripheral_Registers_Bits_Definition, 820

DFSDM_FLTISR_ROVRF
 Peripheral_Registers_Bits_Definition, 820

DFSDM_FLTISR_SCDF
 Peripheral_Registers_Bits_Definition, 820

DFSDM_FLTJCHGR_JCHG
 Peripheral_Registers_Bits_Definition, 820

DFSDM_FLTJDATAR_JDATA
 Peripheral_Registers_Bits_Definition, 821

DFSDM_FLTJDATAR_JDATACH
 Peripheral_Registers_Bits_Definition, 821

DFSDM_FLTRDATAR_RDATA
 Peripheral_Registers_Bits_Definition, 821

DFSDM_FLTRDATAR_RDATACH
 Peripheral_Registers_Bits_Definition, 821

DFSDM_FLTRDATAR_RPEND
 Peripheral_Registers_Bits_Definition, 821

DFSR
 ITM Functions, 132

DHCSR
 ITM Functions, 132

DHR12L1
 CMSIS, 480

DHR12L2
 CMSIS, 480

DHR12LD
 CMSIS, 480

DHR12R1
 CMSIS, 480

DHR12R2
 CMSIS, 480

DHR12RD
 CMSIS, 480

DHR8R1
 CMSIS, 481

DHR8R2
 CMSIS, 481

DHR8RD
 CMSIS, 481

DIER
 CMSIS, 481

TIM1_TypeDef, 1058

DIN
 CMSIS, 481

DISABLE
 I2C Driver, 270

Disabled
 SPI Driver, 316

Display_Mode
 LCD_t, 1025

DLEN
 CMSIS, 481

DMA1_Stream0_IRQHandler
 Configuration_section_for\CMSIS, 525

DMA1_Stream1_IRQHandler
 Configuration_section_for\CMSIS, 525

DMA1_Stream2_IRQHandler
 Configuration_section_for\CMSIS, 525

DMA1_Stream3_IRQHandler
 Configuration_section_for\CMSIS, 525

DMA1_Stream4_IRQHandler
 Configuration_section_for\CMSIS, 525

DMA1_Stream5_IRQHandler
 Configuration_section_for\CMSIS, 525

DMA1_Stream6_IRQHandler
 Configuration_section_for\CMSIS, 525

DMA1_Stream7_IRQHandler
 Configuration_section_for\CMSIS, 526

DMA2_Stream0_IRQHandler
 Configuration_section_for\CMSIS, 526

DMA2_Stream1_IRQHandler
 Configuration_section_for\CMSIS, 526

DMA2_Stream2_IRQHandler
 Configuration_section_for\CMSIS, 526

DMA2_Stream3_IRQHandler
 Configuration_section_for\CMSIS, 526

DMA2_Stream4_IRQHandler
 Configuration_section_for\CMSIS, 526

DMA2_Stream5_IRQHandler
 Configuration_section_for\CMSIS, 526

DMA2_Stream6_IRQHandler
 Configuration_section_for\CMSIS, 526

DMA2_Stream7_IRQHandler
 Configuration_section_for\CMSIS, 526

DMA2D_AMTCR_DT
 Peripheral_Registers_Bits_Definition, 821

DMA2D_AMTCR_EN
 Peripheral_Registers_Bits_Definition, 821

DMA2D_BASE
 Peripheral_memory_map, 535

DMA2D_BGCMAR_MA
 Peripheral_Registers_Bits_Definition, 821

DMA2D_BGCOLR_BLUE
 Peripheral_Registers_Bits_Definition, 822

DMA2D_BGCOLR_GREEN
 Peripheral_Registers_Bits_Definition, 822

DMA2D_BGCOLR_RED
 Peripheral_Registers_Bits_Definition, 822

DMA2D_BGMAR_MA

- Peripheral_Registers_Bits_Definition, [822](#)
- DMA2D_BGOR_LO
 Peripheral_Registers_Bits_Definition, [822](#)
- DMA2D_BGPFCCR_ALPHA
 Peripheral_Registers_Bits_Definition, [822](#)
- DMA2D_BGPFCCR_AM
 Peripheral_Registers_Bits_Definition, [822](#)
- DMA2D_BGPFCCR_AM_0
 Peripheral_Registers_Bits_Definition, [822](#)
- DMA2D_BGPFCCR_AM_1
 Peripheral_Registers_Bits_Definition, [823](#)
- DMA2D_BGPFCCR_CCM
 Peripheral_Registers_Bits_Definition, [823](#)
- DMA2D_BGPFCCR_CM
 Peripheral_Registers_Bits_Definition, [823](#)
- DMA2D_BGPFCCR_CM_0
 Peripheral_Registers_Bits_Definition, [823](#)
- DMA2D_BGPFCCR_CM_1
 Peripheral_Registers_Bits_Definition, [823](#)
- DMA2D_BGPFCCR_CM_2
 Peripheral_Registers_Bits_Definition, [823](#)
- DMA2D_BGPFCCR_CS
 Peripheral_Registers_Bits_Definition, [823](#)
- DMA2D_BGPFCCR_START
 Peripheral_Registers_Bits_Definition, [823](#)
- DMA2D_CR_ABORT
 Peripheral_Registers_Bits_Definition, [824](#)
- DMA2D_CR_CAEIE
 Peripheral_Registers_Bits_Definition, [824](#)
- DMA2D_CR_CEIE
 Peripheral_Registers_Bits_Definition, [824](#)
- DMA2D_CR_CTCIE
 Peripheral_Registers_Bits_Definition, [824](#)
- DMA2D_CR_MODE
 Peripheral_Registers_Bits_Definition, [824](#)
- DMA2D_CR_START
 Peripheral_Registers_Bits_Definition, [824](#)
- DMA2D_CR_SUSP
 Peripheral_Registers_Bits_Definition, [824](#)
- DMA2D_CR_TCIE
 Peripheral_Registers_Bits_Definition, [824](#)
- DMA2D_CR_TEIE
 Peripheral_Registers_Bits_Definition, [825](#)
- DMA2D_CR_TWIE
 Peripheral_Registers_Bits_Definition, [825](#)
- DMA2D_FGCMAR_MA
 Peripheral_Registers_Bits_Definition, [825](#)
- DMA2D_FGCOLR_BLUE
 Peripheral_Registers_Bits_Definition, [825](#)
- DMA2D_FGCOLR_GREEN
 Peripheral_Registers_Bits_Definition, [825](#)
- DMA2D_FGCOLR_RED
 Peripheral_Registers_Bits_Definition, [825](#)
- DMA2D_FGMAR_MA
 Peripheral_Registers_Bits_Definition, [825](#)
- DMA2D_FGOR_LO
 Peripheral_Registers_Bits_Definition, [825](#)
- DMA2D_FGPFCCR_ALPHA
 Peripheral_Registers_Bits_Definition, [826](#)
- DMA2D_FGPFCCR_AM
 Peripheral_Registers_Bits_Definition, [826](#)
- DMA2D_FGPFCCR_AM_0
 Peripheral_Registers_Bits_Definition, [826](#)
- DMA2D_FGPFCCR_AM_1
 Peripheral_Registers_Bits_Definition, [826](#)
- DMA2D_FGPFCCR_CCM
 Peripheral_Registers_Bits_Definition, [826](#)
- DMA2D_FGPFCCR_CM
 Peripheral_Registers_Bits_Definition, [826](#)
- DMA2D_FGPFCCR_CM_0
 Peripheral_Registers_Bits_Definition, [826](#)
- DMA2D_FGPFCCR_CM_1
 Peripheral_Registers_Bits_Definition, [826](#)
- DMA2D_FGPFCCR_CM_2
 Peripheral_Registers_Bits_Definition, [827](#)
- DMA2D_FGPFCCR_CM_3
 Peripheral_Registers_Bits_Definition, [827](#)
- DMA2D_FGPFCCR_CS
 Peripheral_Registers_Bits_Definition, [827](#)
- DMA2D_FGPFCCR_START
 Peripheral_Registers_Bits_Definition, [827](#)
- DMA2D_IFCR_CAEIF
 Peripheral_Registers_Bits_Definition, [827](#)
- DMA2D_IFCR_CCEIF
 Peripheral_Registers_Bits_Definition, [827](#)
- DMA2D_IFCR_CCTCIF
 Peripheral_Registers_Bits_Definition, [828](#)
- DMA2D_IFCR_CTCIF
 Peripheral_Registers_Bits_Definition, [828](#)
- DMA2D_IFCR_CTEIF
 Peripheral_Registers_Bits_Definition, [828](#)
- DMA2D_IFCR_CTWIF
 Peripheral_Registers_Bits_Definition, [828](#)
- DMA2D_IFSR_CCAEIF
 Peripheral_Registers_Bits_Definition, [828](#)
- DMA2D_IFSR_CCEIF
 Peripheral_Registers_Bits_Definition, [828](#)
- DMA2D_IFSR_CCTCIF
 Peripheral_Registers_Bits_Definition, [828](#)
- DMA2D_IFSR_CTCIF
 Peripheral_Registers_Bits_Definition, [829](#)
- DMA2D_IFSR_CTEIF
 Peripheral_Registers_Bits_Definition, [829](#)
- DMA2D_IFSR_CTWIF
 Peripheral_Registers_Bits_Definition, [829](#)
- DMA2D_ISR_CAEIF
 Peripheral_Registers_Bits_Definition, [829](#)
- DMA2D_ISR_CEIF
 Peripheral_Registers_Bits_Definition, [829](#)
- DMA2D_ISR_CTCIF
 Peripheral_Registers_Bits_Definition, [829](#)
- DMA2D_ISR_TCIF
 Peripheral_Registers_Bits_Definition, [829](#)
- DMA2D_ISR_TEIF
 Peripheral_Registers_Bits_Definition, [830](#)
- DMA2D_ISR_TWIF

Peripheral_Registers_Bits_Definition, 830
DMA2D_LWR_LW
 Peripheral_Registers_Bits_Definition, 830
DMA2D_NLR_NL
 Peripheral_Registers_Bits_Definition, 830
DMA2D_NLR_PL
 Peripheral_Registers_Bits_Definition, 830
DMA2D_OCOLR_ALPHA_1
 Peripheral_Registers_Bits_Definition, 830
DMA2D_OCOLR_ALPHA_3
 Peripheral_Registers_Bits_Definition, 830
DMA2D_OCOLR_ALPHA_4
 Peripheral_Registers_Bits_Definition, 830
DMA2D_OCOLR_BLUE_1
 Peripheral_Registers_Bits_Definition, 831
DMA2D_OCOLR_BLUE_2
 Peripheral_Registers_Bits_Definition, 831
DMA2D_OCOLR_BLUE_3
 Peripheral_Registers_Bits_Definition, 831
DMA2D_OCOLR_BLUE_4
 Peripheral_Registers_Bits_Definition, 831
DMA2D_OCOLR_GREEN_1
 Peripheral_Registers_Bits_Definition, 831
DMA2D_OCOLR_GREEN_2
 Peripheral_Registers_Bits_Definition, 831
DMA2D_OCOLR_GREEN_3
 Peripheral_Registers_Bits_Definition, 831
DMA2D_OCOLR_GREEN_4
 Peripheral_Registers_Bits_Definition, 832
DMA2D_OCOLR_RED_1
 Peripheral_Registers_Bits_Definition, 832
DMA2D_OCOLR_RED_2
 Peripheral_Registers_Bits_Definition, 832
DMA2D_OCOLR_RED_3
 Peripheral_Registers_Bits_Definition, 832
DMA2D_OCOLR_RED_4
 Peripheral_Registers_Bits_Definition, 832
DMA2D_OMAR_MA
 Peripheral_Registers_Bits_Definition, 832
DMA2D_OOR_LO
 Peripheral_Registers_Bits_Definition, 832
DMA2D_OPFCCR_CM
 Peripheral_Registers_Bits_Definition, 833
DMA2D_OPFCCR_CM_0
 Peripheral_Registers_Bits_Definition, 833
DMA2D_OPFCCR_CM_1
 Peripheral_Registers_Bits_Definition, 833
DMA2D_OPFCCR_CM_2
 Peripheral_Registers_Bits_Definition, 833
DMA2D_TypeDef, 1008
DMA_Stream_TypeDef, 1009
DMA_TypeDef, 1010
DMACR
 CMSIS, 481
DMAR
 CMSIS, 482
 TIM1_TypeDef, 1058
dms

API Function Prototypes, 342
DO1
 CMSIS, 482
DOR2
 CMSIS, 482
DOUT
 CMSIS, 482
DR
 CMSIS, 482, 483
 CRC_TypeDef, 1004
 I2C_TypeDef, 1020
 SPI_TypeDef, 1051
 USART_TypeDef, 1063
drivers/Inc/Platform_Types.h, 1185
drivers/Inc/stm32f401xx.h, 1187
drivers/Inc/stm32f401xx_gpio_driver.h, 1197
drivers/Inc/stm32f401xx_i2c_driver.h, 1200
drivers/Inc/stm32f401xx_nvic_driver.h, 1202
drivers/Inc/stm32f401xx_rcc_driver.h, 1207
drivers/Inc/stm32f401xx_spi_driver.h, 1209
drivers/Inc/stm32f401xx_systick_driver.h, 1211
drivers/Inc/stm32f401xx_timer_driver.h, 1213
drivers/Inc/stm32f401xx_usart_driver.h, 1215
drivers/Src/stm32f401xx_gpio_driver.c, 1217
drivers/Src/stm32f401xx_i2c_driver.c, 1218
drivers/Src/stm32f401xx_spi_driver.c, 1220
DTIMER
 CMSIS, 483
dus
 API Function Prototypes, 342
DWT
 CMSIS, 409
 Core Definitions, 109
DWT_BASE
 CMSIS, 409
 Core Definitions, 109
DWT_CPICNT_CPICNT_Msk
 CMSIS, 409
 Data Watchpoint and Trace (DWT), 80
DWT_CPICNT_CPICNT_Pos
 CMSIS, 409
 Data Watchpoint and Trace (DWT), 81
DWT_CTRL_CPIEVTEA_Msk
 CMSIS, 409
 Data Watchpoint and Trace (DWT), 81
DWT_CTRL_CPIEVTEA_Pos
 CMSIS, 409
 Data Watchpoint and Trace (DWT), 81
DWT_CTRL_CYCCNTENA_Msk
 CMSIS, 409
 Data Watchpoint and Trace (DWT), 81
DWT_CTRL_CYCCNTENA_Pos
 CMSIS, 410
 Data Watchpoint and Trace (DWT), 81
DWT_CTRL_CYCEVTENA_Msk
 CMSIS, 410
 Data Watchpoint and Trace (DWT), 81
DWT_CTRL_CYCEVTENA_Pos

CMSIS, 410
Data Watchpoint and Trace (DWT), 81
DWT_CTRL_CYCTAP_Msk
CMSIS, 410
Data Watchpoint and Trace (DWT), 81
DWT_CTRL_CYCTAP_Pos
CMSIS, 410
Data Watchpoint and Trace (DWT), 82
DWT_CTRL_EXCEVTENA_Msk
CMSIS, 410
Data Watchpoint and Trace (DWT), 82
DWT_CTRL_EXCEVTENA_Pos
CMSIS, 410
Data Watchpoint and Trace (DWT), 82
DWT_CTRL_EXCTRCENA_Msk
CMSIS, 410
Data Watchpoint and Trace (DWT), 82
DWT_CTRL_EXCTRCENA_Pos
CMSIS, 411
Data Watchpoint and Trace (DWT), 82
DWT_CTRL_FOLDEVTENA_Msk
CMSIS, 411
Data Watchpoint and Trace (DWT), 82
DWT_CTRL_FOLDEVTENA_Pos
CMSIS, 411
Data Watchpoint and Trace (DWT), 82
DWT_CTRL_LSUEVTENA_Msk
CMSIS, 411
Data Watchpoint and Trace (DWT), 82
DWT_CTRL_LSUEVTENA_Pos
CMSIS, 411
Data Watchpoint and Trace (DWT), 83
DWT_CTRL_NOCYCCNT_Msk
CMSIS, 411
Data Watchpoint and Trace (DWT), 83
DWT_CTRL_NOCYCCNT_Pos
CMSIS, 411
Data Watchpoint and Trace (DWT), 83
DWT_CTRL_NOEXTTRIG_Msk
CMSIS, 411
Data Watchpoint and Trace (DWT), 83
DWT_CTRL_NOEXTTRIG_Pos
CMSIS, 412
Data Watchpoint and Trace (DWT), 83
DWT_CTRL_NOPRFCNT_Msk
CMSIS, 412
Data Watchpoint and Trace (DWT), 83
DWT_CTRL_NOPRFCNT_Pos
CMSIS, 412
Data Watchpoint and Trace (DWT), 83
DWT_CTRL_NOTRCPKT_Msk
CMSIS, 412
Data Watchpoint and Trace (DWT), 83
DWT_CTRL_NOTRCPKT_Pos
CMSIS, 412
Data Watchpoint and Trace (DWT), 84
DWT_CTRL_NUMCOMP_Msk
CMSIS, 412
Data Watchpoint and Trace (DWT), 84
Data Watchpoint and Trace (DWT), 84
DWT_CTRL_NUMCOMP_Pos
CMSIS, 412
Data Watchpoint and Trace (DWT), 84
DWT_CTRL_PCSAMPLENA_Msk
CMSIS, 412
Data Watchpoint and Trace (DWT), 84
DWT_CTRL_PCSAMPLENA_Pos
CMSIS, 413
Data Watchpoint and Trace (DWT), 84
DWT_CTRL_POSTINIT_Msk
CMSIS, 413
Data Watchpoint and Trace (DWT), 84
DWT_CTRL_POSTINIT_Pos
CMSIS, 413
Data Watchpoint and Trace (DWT), 84
DWT_CTRL_POSTPRESET_Msk
CMSIS, 413
Data Watchpoint and Trace (DWT), 84
DWT_CTRL_POSTPRESET_Pos
CMSIS, 413
Data Watchpoint and Trace (DWT), 85
DWT_CTRL_SLEEPEVTENA_Msk
CMSIS, 413
Data Watchpoint and Trace (DWT), 85
DWT_CTRL_SLEEPEVTENA_Pos
CMSIS, 413
Data Watchpoint and Trace (DWT), 85
DWT_CTRL_SYNCTAP_Msk
CMSIS, 413
Data Watchpoint and Trace (DWT), 85
DWT_CTRL_SYNCTAP_Pos
CMSIS, 414
Data Watchpoint and Trace (DWT), 85
DWT_EXCCNT_EXCCNT_Msk
CMSIS, 414
Data Watchpoint and Trace (DWT), 85
DWT_EXCCNT_EXCCNT_Pos
CMSIS, 414
Data Watchpoint and Trace (DWT), 85
DWT_FOLDCNT_FOLDCNT_Msk
CMSIS, 414
Data Watchpoint and Trace (DWT), 85
DWT_FOLDCNT_FOLDCNT_Pos
CMSIS, 414
Data Watchpoint and Trace (DWT), 86
DWT_FUNCTION_CYCMATCH_Msk
CMSIS, 414
Data Watchpoint and Trace (DWT), 86
DWT_FUNCTION_CYCMATCH_Pos
CMSIS, 414
Data Watchpoint and Trace (DWT), 86
DWT_FUNCTION_DATAVADDR0_Msk
CMSIS, 414
Data Watchpoint and Trace (DWT), 86
DWT_FUNCTION_DATAVADDR0_Pos
CMSIS, 415
Data Watchpoint and Trace (DWT), 86

DWT_FUNCTION_DATAVADDR1_Msk
 CMSIS, 415
 Data Watchpoint and Trace (DWT), 86

DWT_FUNCTION_DATAVADDR1_Pos
 CMSIS, 415
 Data Watchpoint and Trace (DWT), 86

DWT_FUNCTION_DATAVMATCH_Msk
 CMSIS, 415
 Data Watchpoint and Trace (DWT), 86

DWT_FUNCTION_DATAVMATCH_Pos
 CMSIS, 415
 Data Watchpoint and Trace (DWT), 87

DWT_FUNCTION_DATAVSIZE_Msk
 CMSIS, 415
 Data Watchpoint and Trace (DWT), 87

DWT_FUNCTION_DATAVSIZE_Pos
 CMSIS, 415
 Data Watchpoint and Trace (DWT), 87

DWT_FUNCTION_EMITRANGE_Msk
 CMSIS, 415
 Data Watchpoint and Trace (DWT), 87

DWT_FUNCTION_EMITRANGE_Pos
 CMSIS, 416
 Data Watchpoint and Trace (DWT), 87

DWT_FUNCTION_FUNCTION_Msk
 CMSIS, 416
 Data Watchpoint and Trace (DWT), 87

DWT_FUNCTION_FUNCTION_Pos
 CMSIS, 416
 Data Watchpoint and Trace (DWT), 87

DWT_FUNCTION_LNK1ENA_Msk
 CMSIS, 416
 Data Watchpoint and Trace (DWT), 87

DWT_FUNCTION_LNK1ENA_Pos
 CMSIS, 416
 Data Watchpoint and Trace (DWT), 88

DWT_FUNCTION_MATCHED_Msk
 CMSIS, 416
 Data Watchpoint and Trace (DWT), 88

DWT_FUNCTION_MATCHED_Pos
 CMSIS, 416
 Data Watchpoint and Trace (DWT), 88

DWT_LSUCNT_LSUCNT_Msk
 CMSIS, 416
 Data Watchpoint and Trace (DWT), 88

DWT_LSUCNT_LSUCNT_Pos
 CMSIS, 417
 Data Watchpoint and Trace (DWT), 88

DWT_MASK_MASK_Msk
 CMSIS, 417
 Data Watchpoint and Trace (DWT), 88

DWT_MASK_MASK_Pos
 CMSIS, 417
 Data Watchpoint and Trace (DWT), 88

DWT_SLEEPcnt_SLEEPcnt_Msk
 CMSIS, 417
 Data Watchpoint and Trace (DWT), 88

DWT_SLEEPcnt_SLEEPcnt_Pos

CMSIS, 417
 Data Watchpoint and Trace (DWT), 88

CMSIS, 417
 Data Watchpoint and Trace (DWT), 89

DWT_Type, 1010

ECU APIs, 361
 Admin_Init, 362
 Check_Flag, 363
 Check_ID, 363
 Check_Password, 364
 combineArrays, 364
 ECU_Init, 365
 Enter_Gate_Open, 366
 Exit_Gate_Open, 366
 Flag_SET_RESET, 367
 Trigger_Alarm, 367
 UserLCD_PrintFreeSlots, 368
 Wrong_RFID, 369

ECU Definitions and APIs, 359

ECU_Init
 API_Definitions, 375
 ECU APIs, 365

EEPROM Constants, 15
 EEPROM_Slave_address, 16

EEPROM_Init
 API Function Prototypes, 16

EEPROM_Read_byte
 API Function Prototypes, 17

EEPROM_Slave_address
 EEPROM Constants, 16

EEPROM_Write_NBytes
 API Function Prototypes, 18

EGR
 CMSIS, 483
 TIM1_TypeDef, 1058

EMR
 CMSIS, 484
 EXTI_TypeDef, 1013

EN_PIN
 LCD_t, 1025

ENABLE
 I2C Driver, 270

Enable_Dual_Address
 S_I2C_Slave_address, 1041

Enabled
 SPI Driver, 316

Enter_Gate_Open
 API_Definitions, 375
 ECU APIs, 366

Enter_Gate_STATE
 Application States, 352

Enter_UART_CallBack
 API_Definitions, 375

Entry_Mode
 LCD_t, 1025

Enum for specifying the LED mode., 32
 LED_Active_High, 33
 LED_Active_Low, 33
 LED_Mode_max, 33
 LED_Mode_t, 33

Enum representing the result of ID check., [360](#)
 ID_Check_Result, [360](#)
 ID_Found, [360](#)
 ID_NOT_Found, [360](#)

Enumeration and Structures for USART, [344](#)

ERRI
 S_IRQ_SRC, [1042](#)

ESCR
 CMSIS, [484](#)

ESR
 CMSIS, [484](#)

ESUR
 CMSIS, [484](#)

ETH_MACCR_BL
 Peripheral_Registers_Bits_Definition, [833](#)

ETH_TypeDef, [1011](#)

EV5
 I2C Driver, [271](#)

EV6
 I2C Driver, [271](#)

EV7
 I2C Driver, [271](#)

EV8
 I2C Driver, [271](#)

EV8_1
 I2C Driver, [271](#)

EXCCNT
 ITM Functions, [132](#)

Exit_Flag
 Global_Variables_Definitions, [371](#)

Exit_Gate_Open
 API_Definitions, [376](#)
 ECU APIs, [366](#)

Exit_Gate_STATE
 Application States, [352](#)

Exit_UART_CallBack
 API_Definitions, [376](#)

Exported_constants, [539](#)

Exported_macro, [989](#)

Exported_types, [527](#)
 FlagStatus, [529](#)
 Reset, [530](#)
 s32, [528](#)
 sc16, [528](#)
 sc32, [528](#)
 sc8, [528](#)
 Set, [530](#)
 uc16, [528](#)
 uc32, [528](#)
 uc8, [528](#)
 vsc16, [529](#)
 vsc32, [529](#)
 vsc8, [529](#)
 vuc16, [529](#)
 vuc32, [529](#)
 vuc8, [529](#)

EXTI
 EXTI Instance, [173](#)

 EXTI Instance, [173](#)
 EXTI, [173](#)

EXTI Register Map, [164](#)

EXTI0_IRQHandler
 Configuration_section_for\CMSIS, [525](#)

EXTI15_10_IRQHandler
 Configuration_section_for\CMSIS, [526](#)

EXTI1_IRQHandler
 Configuration_section_for\CMSIS, [525](#)

EXTI2_IRQHandler
 Configuration_section_for\CMSIS, [525](#)

EXTI3_IRQHandler
 Configuration_section_for\CMSIS, [525](#)

EXTI4_IRQHandler
 Configuration_section_for\CMSIS, [525](#)

EXTI9_5_IRQHandler
 Configuration_section_for\CMSIS, [525](#)

EXTI_EMR_MR0
 Peripheral_Registers_Bits_Definition, [833](#)

EXTI_EMR_MR1
 Peripheral_Registers_Bits_Definition, [833](#)

EXTI_EMR_MR10
 Peripheral_Registers_Bits_Definition, [834](#)

EXTI_EMR_MR11
 Peripheral_Registers_Bits_Definition, [834](#)

EXTI_EMR_MR12
 Peripheral_Registers_Bits_Definition, [834](#)

EXTI_EMR_MR13
 Peripheral_Registers_Bits_Definition, [834](#)

EXTI_EMR_MR14
 Peripheral_Registers_Bits_Definition, [834](#)

EXTI_EMR_MR15
 Peripheral_Registers_Bits_Definition, [834](#)

EXTI_EMR_MR16
 Peripheral_Registers_Bits_Definition, [834](#)

EXTI_EMR_MR17
 Peripheral_Registers_Bits_Definition, [834](#)

EXTI_EMR_MR18
 Peripheral_Registers_Bits_Definition, [835](#)

EXTI_EMR_MR19
 Peripheral_Registers_Bits_Definition, [835](#)

EXTI_EMR_MR2
 Peripheral_Registers_Bits_Definition, [835](#)

EXTI_EMR_MR23
 Peripheral_Registers_Bits_Definition, [835](#)

EXTI_EMR_MR3
 Peripheral_Registers_Bits_Definition, [835](#)

EXTI_EMR_MR4
 Peripheral_Registers_Bits_Definition, [835](#)

EXTI_EMR_MR5
 Peripheral_Registers_Bits_Definition, [835](#)

EXTI_EMR_MR6
 Peripheral_Registers_Bits_Definition, [835](#)

EXTI_EMR_MR7
 Peripheral_Registers_Bits_Definition, [836](#)

EXTI_EMR_MR8
 Peripheral_Registers_Bits_Definition, [836](#)

EXTI_EMR_MR9

Peripheral_Registers_Bits_Definition, 843
EXTI_PR_PR5
 Peripheral_Registers_Bits_Definition, 843
EXTI_PR_PR6
 Peripheral_Registers_Bits_Definition, 843
EXTI_PR_PR7
 Peripheral_Registers_Bits_Definition, 843
EXTI_PR_PR8
 Peripheral_Registers_Bits_Definition, 844
EXTI_PR_PR9
 Peripheral_Registers_Bits_Definition, 844
EXTI_RTSR_TR0
 Peripheral_Registers_Bits_Definition, 844
EXTI_RTSR_TR1
 Peripheral_Registers_Bits_Definition, 844
EXTI_RTSR_TR10
 Peripheral_Registers_Bits_Definition, 844
EXTI_RTSR_TR11
 Peripheral_Registers_Bits_Definition, 844
EXTI_RTSR_TR12
 Peripheral_Registers_Bits_Definition, 844
EXTI_RTSR_TR13
 Peripheral_Registers_Bits_Definition, 844
EXTI_RTSR_TR14
 Peripheral_Registers_Bits_Definition, 845
EXTI_RTSR_TR15
 Peripheral_Registers_Bits_Definition, 845
EXTI_RTSR_TR16
 Peripheral_Registers_Bits_Definition, 845
EXTI_RTSR_TR17
 Peripheral_Registers_Bits_Definition, 845
EXTI_RTSR_TR18
 Peripheral_Registers_Bits_Definition, 845
EXTI_RTSR_TR19
 Peripheral_Registers_Bits_Definition, 845
EXTI_RTSR_TR2
 Peripheral_Registers_Bits_Definition, 845
EXTI_RTSR_TR23
 Peripheral_Registers_Bits_Definition, 845
EXTI_RTSR_TR3
 Peripheral_Registers_Bits_Definition, 846
EXTI_RTSR_TR4
 Peripheral_Registers_Bits_Definition, 846
EXTI_RTSR_TR5
 Peripheral_Registers_Bits_Definition, 846
EXTI_RTSR_TR6
 Peripheral_Registers_Bits_Definition, 846
EXTI_RTSR_TR7
 Peripheral_Registers_Bits_Definition, 846
EXTI_RTSR_TR8
 Peripheral_Registers_Bits_Definition, 846
EXTI_RTSR_TR9
 Peripheral_Registers_Bits_Definition, 846
EXTI_SWIER_SWIER0
 Peripheral_Registers_Bits_Definition, 846
EXTI_SWIER_SWIER1
 Peripheral_Registers_Bits_Definition, 847
EXTI_SWIER_SWIER10
 Peripheral_Registers_Bits_Definition, 847
Peripheral_Registers_Bits_Definition, 847
EXTI_SWIER_SWIER11
 Peripheral_Registers_Bits_Definition, 847
EXTI_SWIER_SWIER12
 Peripheral_Registers_Bits_Definition, 847
EXTI_SWIER_SWIER13
 Peripheral_Registers_Bits_Definition, 847
EXTI_SWIER_SWIER14
 Peripheral_Registers_Bits_Definition, 847
EXTI_SWIER_SWIER15
 Peripheral_Registers_Bits_Definition, 847
EXTI_SWIER_SWIER16
 Peripheral_Registers_Bits_Definition, 847
EXTI_SWIER_SWIER17
 Peripheral_Registers_Bits_Definition, 848
EXTI_SWIER_SWIER18
 Peripheral_Registers_Bits_Definition, 848
EXTI_SWIER_SWIER19
 Peripheral_Registers_Bits_Definition, 848
EXTI_SWIER_SWIER2
 Peripheral_Registers_Bits_Definition, 848
EXTI_SWIER_SWIER23
 Peripheral_Registers_Bits_Definition, 848
EXTI_SWIER_SWIER3
 Peripheral_Registers_Bits_Definition, 848
EXTI_SWIER_SWIER4
 Peripheral_Registers_Bits_Definition, 848
EXTI_SWIER_SWIER5
 Peripheral_Registers_Bits_Definition, 848
EXTI_SWIER_SWIER6
 Peripheral_Registers_Bits_Definition, 849
EXTI_SWIER_SWIER7
 Peripheral_Registers_Bits_Definition, 849
EXTI_SWIER_SWIER8
 Peripheral_Registers_Bits_Definition, 849
EXTI_SWIER_SWIER9
 Peripheral_Registers_Bits_Definition, 849
EXTI_TypeDef, 1012
 EMR, 1013
 FTSR, 1013
 IMR, 1013
 PR, 1013
 RTSR, 1013
 SWIER, 1014
EXTICR
 CMSIS, 484
 SYSCFG_RegDef_t, 1054
FA1R
 CMSIS, 484
FALSE
 Platform Standard Types, 151
FCR
 CMSIS, 484
FFA1R
 CMSIS, 485
FFCR
 ITM Functions, 132
FFSR

ITM Functions, 132
FGCLUT
 CMSIS, 485
FGCMAR
 CMSIS, 485
FGCOLR
 CMSIS, 485
FGMAR
 CMSIS, 485
FGOR
 CMSIS, 485
FGPFCCR
 CMSIS, 485
FIFO
 CMSIS, 486
FIFO0
 ITM Functions, 132
FIFO1
 ITM Functions, 133
FIFOCNT
 CMSIS, 486
FLAG_RESET
 Platform Standard Types, 151
FLAG_SET
 Platform Standard Types, 151
Flag_Set_Reset
 API_Definitions, 376
 ECU APIs, 367
FlagStatus
 Exported_types, 529
 I2C Driver, 269
FLASH_BASE
 Peripheral_memory_map, 535
FLASH_IRQn
 Configuration_section_for_CMSIS, 525
FLASH_TypeDef, 1014
float32
 Platform Standard Types, 152
float64
 Platform Standard Types, 153
FLTR
 CMSIS, 486
FM1R
 CMSIS, 486
FMR
 CMSIS, 486
FOLDCNT
 ITM Functions, 133
fp_App_State_Handler
 Application States, 354
 Prototypes, 356
FPCA
 ITM Functions, 133
FPU_IRQHandler
 Configuration_section_for_CMSIS, 526
FR1
 CMSIS, 486
FR2

CMSIS, 486
Frame_Format
 S_SPI_Config_t, 1044
FRCR
 CMSIS, 487
Free_Slots
 Application States, 354
 Global_Variables_Definitions, 371
FS1R
 CMSIS, 487
FSCR
 ITM Functions, 133
FTSR
 CMSIS, 487
 EXTI_TypeDef, 1013
Full_STATE
 Application States, 352
FUNCTION0
 ITM Functions, 133
FUNCTION1
 ITM Functions, 133
FUNCTION2
 ITM Functions, 134
FUNCTION3
 ITM Functions, 134
Functional_State
 I2C Driver, 269
Functions_and_Instructions_Reference, 111
Functions_Declarations, 369
GCR
 CMSIS, 487
GE
 ITM Functions, 134
General_Call_Address_Detection
 S_I2C_Config_t, 1039
Global_I2C_Config
 stm32f401xx_i2c_driver.c, 1220
Global_SPI_Config
 stm32f401xx_spi_driver.c, 1223
Global_Variables_Definitions, 370
 Exit_Flag, 371
 Free_Slots, 371
 Print_Slots_LCD_Flag, 371
 SIZE1, 370
GPIO_Base_Address_to_Code_Conversion_Macros, 246
 GPIO_BASEADDR_TO_CODE, 246
GPIO_Driver, 247
GPIO_Instances, 171
 GPIOA, 171
 GPIOB, 171
 GPIOC, 172
 GPIOD, 172
 GPIOE, 172
GPIO_Output_Speeds, 255
 GPIO_SPEED_FAST, 256
 GPIO_SPEED_HIGH, 256
 GPIO_SPEED_LOW, 256
 GPIO_SPEED_MEDIUM, 256

GPIO Output Types, 258
 GPIO_OP_TYPE_OD, 258
 GPIO_OP_TYPE_PP, 258
GPIO Pin Modes, 249, 254
 GPIO_MODE_AF_INPUT, 250
 GPIO_MODE_ALTFN, 254
 GPIO_MODE_ANALOG, 254
 GPIO_MODE_IN, 254
 GPIO_MODE_INPUT_FLO, 250
 GPIO_MODE_INPUT_PD, 250
 GPIO_MODE_INPUT_PU, 250
 GPIO_MODE_IT_FT, 255
 GPIO_MODE_IT_RFT, 255
 GPIO_MODE_IT_RT, 255
 GPIO_MODE_OUT, 255
 GPIO_MODE_OUTPUT_AF_OD, 250
 GPIO_MODE_OUTPUT_AF_PP, 250
 GPIO_MODE_OUTPUT_OD, 250
 GPIO_MODE_OUTPUT_PP, 251
GPIO Pin Numbers, 248
GPIO Pin Speeds, 251
 GPIO_SPEED_10M, 251
 GPIO_SPEED_2M, 251
 GPIO_SPEED_50M, 252
GPIO Pin States, 252
 GPIO_PIN_RESET, 252
 GPIO_PIN_SET, 252
GPIO Pull-up/Pull-down Configurations, 257
 GPIO_NO_PUPD, 257
 GPIO_PIN_PD, 257
 GPIO_PIN_PU, 257
GPIO Register Map, 163
GPIO Return Lock Status, 253
 GPIO_RETURN_LOCK_ERROR, 253
 GPIO_RETURN_LOCK_OK, 253
GPIO_BASEADDR_TO_CODE
 GPIO Base Address to Code Conversion Macros, 246
GPIO_MODE
 GPIO_PinConfig_t, 1015
GPIO_MODE_AF_INPUT
 GPIO Pin Modes, 250
GPIO_MODE_ALTFN
 GPIO Pin Modes, 254
GPIO_MODE_ANALOG
 GPIO Pin Modes, 254
GPIO_MODE_IN
 GPIO Pin Modes, 254
GPIO_MODE_INPUT_FLO
 GPIO Pin Modes, 250
GPIO_MODE_INPUT_PD
 GPIO Pin Modes, 250
GPIO_MODE_INPUT_PU
 GPIO Pin Modes, 250
GPIO_MODE_IT_FT
 GPIO Pin Modes, 255
GPIO_MODE_IT_RFT
 GPIO Pin Modes, 255
GPIO_MODE_IT_RT
 GPIO Pin Modes, 255
GPIO_MODE_OUT
 GPIO Pin Modes, 255
GPIO_MODE_OUTPUT_AF_OD
 GPIO Pin Modes, 250
GPIO_MODE_OUTPUT_AF_PP
 GPIO Pin Modes, 250
GPIO_MODE_OUTPUT_OD
 GPIO Pin Modes, 250
GPIO_MODE_OUTPUT_PP
 GPIO Pin Modes, 251
GPIO_NO_PUPD
 GPIO Pull-up/Pull-down Configurations, 257
GPIO_OP_TYPE_OD
 GPIO Output Types, 258
GPIO_OP_TYPE_PP
 GPIO Output Types, 258
GPIO_PIN_PD
 GPIO Pull-up/Pull-down Configurations, 257
GPIO_PIN_PU
 GPIO Pull-up/Pull-down Configurations, 257
GPIO_PIN_RESET
 GPIO Pin States, 252
GPIO_PIN_SET
 GPIO Pin States, 252
GPIO_PinAltFunMode
 GPIO_PinConfig_t, 1015
GPIO_PinConfig_t, 1014
 GPIO_MODE, 1015
 GPIO_PinAltFunMode, 1015
 GPIO_PinNumber, 1015
 GPIO_PinOPType, 1015
 GPIO_PinPuPdControl, 1015
 GPIO_PinSpeed, 1016
GPIO_PinNumber
 GPIO_PinConfig_t, 1015
GPIO_PinOPType
 GPIO_PinConfig_t, 1015
GPIO_PinPuPdControl
 GPIO_PinConfig_t, 1015
GPIO_PinSpeed
 GPIO_PinConfig_t, 1016
GPIO_PORT
 LCD_t, 1025
GPIO_RETURN_LOCK_ERROR
 GPIO Return Lock Status, 253
GPIO_RETURN_LOCK_OK
 GPIO Return Lock Status, 253
GPIO_SPEED_10M
 GPIO Pin Speeds, 251
GPIO_SPEED_2M
 GPIO Pin Speeds, 251
GPIO_SPEED_50M
 GPIO Pin Speeds, 252
GPIO_SPEED_FAST
 GPIO Output Speeds, 256
GPIO_SPEED_HIGH

GPIO Output Speeds, 256
GPIO_SPEED_LOW
 GPIO Output Speeds, 256
GPIO_SPEED_MEDIUM
 GPIO Output Speeds, 256
GPIO_TypeDef, 1016
 AFR, 1017
 BSRR, 1017
 IDR, 1017
 LCKR, 1017
 MODER, 1017
 ODR, 1017
 OSPEEDR, 1017
 OTYPER, 1017
 PUPDR, 1018
GPIOA
 GPIO Instances, 171
GPIOA_BASE
 Peripheral_memory_map, 535
GPIOB
 GPIO Instances, 171
GPIOC
 GPIO Instances, 172
GPIOD
 GPIO Instances, 172
GPIOE
 GPIO Instances, 172
GTPR
 CMSIS, 487
 USART_TypeDef, 1063
HASH_DIGEST_TypeDef, 1018
HASH_TypeDef, 1018
HFSR
 ITM Functions, 134
 SCB_TypeDef, 1047
HIFCR
 CMSIS, 487
HIGH_BYTE_FIRST
 Platform Standard Types, 151
HISR
 CMSIS, 488
HR
 CMSIS, 488
HSE_CLK
 Clock Source Selection, 306
HSE_STARTUP_TIMEOUT
 Library_configuration_section, 521
HSE_VALUE
 Clock Source Selection, 306
 Library_configuration_section, 521
HSI_RC_CLK
 Clock Source Selection, 307
HSI_VALUE
 Clock Source Selection, 307
 Library_configuration_section, 522
HTR
 CMSIS, 488

I2C Acknowledge Control Definitions, 275
 I2C_Ack_Control_Disable, 276
 I2C_Ack_Control_Enable, 276
I2C Addressing Modes, 274
 I2C_Addressing_Slave_10bits, 275
 I2C_Addressing_Slave_7bits, 275
I2C CCR Register Bit Definitions, 203
 I2C_CCR_CCR, 204
 I2C_CCR_CCR_Msk, 204
 I2C_CCR_CCR_Pos, 204
 I2C_CCR_DUTY, 204
 I2C_CCR_DUTY_Msk, 204
 I2C_CCR_DUTY_Pos, 204
I2C CR1 Register Bit Definitions, 180
 I2C_CR1_ACK, 181
 I2C_CR1_ACK_Msk, 181
 I2C_CR1_ACK_Pos, 181
 I2C_CR1_ALERT, 182
 I2C_CR1_ALERT_Msk, 182
 I2C_CR1_ALERT_Pos, 182
 I2C_CR1_ENARP, 182
 I2C_CR1_ENARP_Msk, 182
 I2C_CR1_ENARP_Pos, 182
 I2C_CR1_ENGC, 182
 I2C_CR1_ENGC_Msk, 183
 I2C_CR1_ENGC_Pos, 183
 I2C_CR1_ENPEC, 183
 I2C_CR1_ENPEC_Msk, 183
 I2C_CR1_ENPEC_Pos, 183
 I2C_CR1_NOSTRETCH, 183
 I2C_CR1_NOSTRETCH_Msk, 183
 I2C_CR1_NOSTRETCH_Pos, 184
 I2C_CR1_PE, 184
 I2C_CR1_PE_Msk, 184
 I2C_CR1_PE_Pos, 184
 I2C_CR1_PEC, 184
 I2C_CR1_PEC_Msk, 184
 I2C_CR1_PEC_Pos, 184
 I2C_CR1_POS, 185
 I2C_CR1_POS_Msk, 185
 I2C_CR1_POS_Pos, 185
 I2C_CR1_SMBTYPE, 185
 I2C_CR1_SMBTYPE_Msk, 185
 I2C_CR1_SMBTYPE_Pos, 185
 I2C_CR1_SMBUS, 185
 I2C_CR1_SMBUS_Msk, 186
 I2C_CR1_SMBUS_Pos, 186
 I2C_CR1_START, 186
 I2C_CR1_START_Msk, 186
 I2C_CR1_START_Pos, 186
 I2C_CR1_STOP, 186
 I2C_CR1_STOP_Msk, 186
 I2C_CR1_STOP_Pos, 187
 I2C_CR1_SWRST, 187
 I2C_CR1_SWRST_Msk, 187
 I2C_CR1_SWRST_Pos, 187
I2C CR2 Register Bit Definitions, 187
 I2C_CR2_DMAEN, 188

I2C_CR2_DMAEN_Msk, 188
I2C_CR2_DMAEN_Pos, 188
I2C_CR2_FREQ, 189
I2C_CR2_FREQ_Msk, 189
I2C_CR2_FREQ_Pos, 189
I2C_CR2_ITBUFEN, 189
I2C_CR2_ITBUFEN_Msk, 189
I2C_CR2_ITBUFEN_Pos, 189
I2C_CR2_ITERREN, 189
I2C_CR2_ITERREN_Msk, 190
I2C_CR2_ITERREN_Pos, 190
I2C_CR2_IteVTEN, 190
I2C_CR2_IteVTEN_Msk, 190
I2C_CR2_IteVTEN_Pos, 190
I2C_CR2_LAST, 190
I2C_CR2_LAST_Msk, 190
I2C_CR2_LAST_Pos, 191
I2C Driver, 267
DISABLE, 270
ENABLE, 270
EV5, 271
EV6, 271
EV7, 271
EV8, 271
EV8_1, 271
FlagStatus, 269
Functional_State, 269
I2C_Direction, 270
I2C_Direction_Receiver, 270
I2C_Direction_Transmitter, 270
I2C_Error_AF, 270
I2C_Ev_Address_Matched, 270
I2C_Ev_Data_RCV, 270
I2C_Ev_Data_Req, 270
I2C_EV_Stop, 270
I2C_Flag_Busy, 271
Master_Byte_Transmitting, 271
Repeated, 270
Repeated_Start, 270
Reset, 269
Set, 269
Slave_State, 270
Start, 270
Status, 271
StopCondition, 271
WithoutStop, 271
WithStop, 271
I2C General Call Enable Definitions, 276
I2C_ENGC_Disable, 276
I2C_ENGC_Enable, 277
I2C Instances, 177
I2C1, 178
I2C2, 178
I2C Mode Definitions, 273
I2C_Mode_I2C, 274
I2C_Mode_SMBus, 274
I2C OAR2 Register Bit Definitions, 191
I2C_OAR2_ADD2, 191
I2C_OAR2_ADD2_Msk, 192
I2C_OAR2_ADD2_Pos, 192
I2C_OAR2_ENDUAL, 192
I2C_OAR2_ENDUAL_Msk, 192
I2C_OAR2_ENDUAL_Pos, 192
I2C Register Map, 166
I2C Registers, 179
I2C Speed Definitions, 271
I2C_SCK_FM_200K, 272
I2C_SCK_FM_400K, 272
I2C_SCK_SM_100K, 272
I2C_SCK_SM_50K, 272
I2C SR1 Register Bit Definitions, 192
I2C_SR1_ADD10, 193
I2C_SR1_ADD10_Msk, 194
I2C_SR1_ADD10_Pos, 194
I2C_SR1_ADDR, 194
I2C_SR1_ADDR_Msk, 194
I2C_SR1_ADDR_Pos, 194
I2C_SR1_AF, 194
I2C_SR1_AF_Msk, 194
I2C_SR1_AF_Pos, 194
I2C_SR1_ARLO, 195
I2C_SR1_ARLO_Msk, 195
I2C_SR1_ARLO_Pos, 195
I2C_SR1_BERR, 195
I2C_SR1_BERR_Msk, 195
I2C_SR1_BERR_Pos, 195
I2C_SR1_BTF, 195
I2C_SR1_BTF_Msk, 195
I2C_SR1_BTF_Pos, 196
I2C_SR1_OVR, 196
I2C_SR1_OVR_Msk, 196
I2C_SR1_OVR_Pos, 196
I2C_SR1_PECERR, 196
I2C_SR1_PECERR_Msk, 196
I2C_SR1_PECERR_Pos, 196
I2C_SR1_RXNE, 196
I2C_SR1_RXNE_Msk, 197
I2C_SR1_RXNE_Pos, 197
I2C_SR1_SB, 197
I2C_SR1_SB_Msk, 197
I2C_SR1_SB_Pos, 197
I2C_SR1_SMBALERT, 197
I2C_SR1_SMBALERT_Msk, 197
I2C_SR1_SMBALERT_Pos, 198
I2C_SR1_STOPF, 198
I2C_SR1_STOPF_Msk, 198
I2C_SR1_STOPF_Pos, 198
I2C_SR1_TIMEOUT, 198
I2C_SR1_TIMEOUT_Msk, 198
I2C_SR1_TIMEOUT_Pos, 198
I2C_SR1_TXE, 198
I2C_SR1_TXE_Msk, 199
I2C_SR1_TXE_Pos, 199
I2C SR2 Register Bit Definitions, 199
I2C_SR2_BUSY, 200
I2C_SR2_BUSY_Msk, 200

I2C_SR2_BUSY_Pos, 200
I2C_SR2_DUALF, 200
I2C_SR2_DUALF_Msk, 200
I2C_SR2_DUALF_Pos, 200
I2C_SR2_GENCALL, 201
I2C_SR2_GENCALL_Msk, 201
I2C_SR2_GENCALL_Pos, 201
I2C_SR2_MSL, 201
I2C_SR2_MSL_Msk, 201
I2C_SR2_MSL_Pos, 201
I2C_SR2_PEC, 201
I2C_SR2_PEC_Msk, 202
I2C_SR2_PEC_Pos, 202
I2C_SR2_SMBDEFAULT, 202
I2C_SR2_SMBDEFAULT_Msk, 202
I2C_SR2_SMBDEFAULT_Pos, 202
I2C_SR2_SMBHOST, 202
I2C_SR2_SMBHOST_Msk, 202
I2C_SR2_SMBHOST_Pos, 202
I2C_SR2_TRA, 203
I2C_SR2_TRA_Msk, 203
I2C_SR2_TRA_Pos, 203
I2C Stretch Mode Definitions, 273
I2C_StretchMode_disabled, 273
I2C_StretchMode_enabled, 273
I2C TRISE Register Bit Definitions, 205
I2C_TRISE_TRISE, 205
I2C_TRISE_TRISE_Msk, 205
I2C_TRISE_TRISE_Pos, 205
I2C1
I2C Instances, 178
I2C1_ER_IRQn
Configuration_section_for\CMSIS, 525
I2C1_EV_IRQn
Configuration_section_for\CMSIS, 525
I2C1_Index
stm32f401xx_i2c_driver.c, 1220
I2C2
I2C Instances, 178
I2C2_ER_IRQn
Configuration_section_for\CMSIS, 526
I2C2_EV_IRQn
Configuration_section_for\CMSIS, 526
I2C2_Index
stm32f401xx_i2c_driver.c, 1220
I2C3_ER_IRQn
Configuration_section_for\CMSIS, 526
I2C3_EV_IRQn
Configuration_section_for\CMSIS, 526
I2C_Ack_Conrtol
S_I2C_Config_t, 1040
I2C_Ack_Control_Disable
I2C Acknowledge Control Definitions, 276
I2C_Ack_Control_Enable
I2C Acknowledge Control Definitions, 276
I2C_ACKnowledgeConfig
API Function Prototypes, 278
I2C_Addressing_Mode
S_I2C_Slave_address, 1041
I2C_Addressing_Slave_10bits
I2C Addressing Modes, 275
I2C_Addressing_Slave_7bits
I2C Addressing Modes, 275
I2C_CCR_CCR
I2C CCR Register Bit Definitions, 204
Peripheral_Registers_Bits_Definition, 849
I2C_CCR_CCR_Msk
I2C CCR Register Bit Definitions, 204
I2C_CCR_CCR_Pos
I2C CCR Register Bit Definitions, 204
I2C_CCR_DUTY
I2C CCR Register Bit Definitions, 204
Peripheral_Registers_Bits_Definition, 849
I2C_CCR_DUTY_Msk
I2C CCR Register Bit Definitions, 204
I2C_CCR_DUTY_Pos
I2C CCR Register Bit Definitions, 204
I2C_CCR_FS
Peripheral_Registers_Bits_Definition, 849
I2C_CR1_ACK
I2C CR1 Register Bit Definitions, 181
Peripheral_Registers_Bits_Definition, 850
I2C_CR1_ACK_Msk
I2C CR1 Register Bit Definitions, 181
I2C_CR1_ACK_Pos
I2C CR1 Register Bit Definitions, 181
I2C_CR1_ALERT
I2C CR1 Register Bit Definitions, 182
Peripheral_Registers_Bits_Definition, 850
I2C_CR1_ALERT_Msk
I2C CR1 Register Bit Definitions, 182
I2C_CR1_ALERT_Pos
I2C CR1 Register Bit Definitions, 182
I2C_CR1_ENARP
I2C CR1 Register Bit Definitions, 182
Peripheral_Registers_Bits_Definition, 850
I2C_CR1_ENARP_Msk
I2C CR1 Register Bit Definitions, 182
I2C_CR1_ENARP_Pos
I2C CR1 Register Bit Definitions, 182
I2C_CR1_ENGC
I2C CR1 Register Bit Definitions, 182
Peripheral_Registers_Bits_Definition, 850
I2C_CR1_ENGC_Msk
I2C CR1 Register Bit Definitions, 183
I2C_CR1_ENGC_Pos
I2C CR1 Register Bit Definitions, 183
I2C_CR1_ENPEC
I2C CR1 Register Bit Definitions, 183
Peripheral_Registers_Bits_Definition, 850
I2C_CR1_ENPEC_Msk
I2C CR1 Register Bit Definitions, 183
I2C_CR1_ENPEC_Pos
I2C CR1 Register Bit Definitions, 183
I2C_CR1_NOSTRETCH
I2C CR1 Register Bit Definitions, 183

Peripheral_Registers_Bits_Definition, 850
I2C_CR1_NOSTRETCH_Msk
 I2C CR1 Register Bit Definitions, 183
I2C_CR1_NOSTRETCH_Pos
 I2C CR1 Register Bit Definitions, 184
I2C_CR1_PE
 I2C CR1 Register Bit Definitions, 184
 Peripheral_Registers_Bits_Definition, 850
I2C_CR1_PE_Msk
 I2C CR1 Register Bit Definitions, 184
I2C_CR1_PE_Pos
 I2C CR1 Register Bit Definitions, 184
I2C_CR1_PEC
 I2C CR1 Register Bit Definitions, 184
 Peripheral_Registers_Bits_Definition, 851
I2C_CR1_PEC_Msk
 I2C CR1 Register Bit Definitions, 184
I2C_CR1_PEC_Pos
 I2C CR1 Register Bit Definitions, 184
I2C_CR1_POS
 I2C CR1 Register Bit Definitions, 185
 Peripheral_Registers_Bits_Definition, 851
I2C_CR1_POS_Msk
 I2C CR1 Register Bit Definitions, 185
I2C_CR1_POS_Pos
 I2C CR1 Register Bit Definitions, 185
I2C_CR1_SMBTYPE
 I2C CR1 Register Bit Definitions, 185
 Peripheral_Registers_Bits_Definition, 851
I2C_CR1_SMBTYPE_Msk
 I2C CR1 Register Bit Definitions, 185
I2C_CR1_SMBTYPE_Pos
 I2C CR1 Register Bit Definitions, 185
I2C_CR1_SMBUS
 I2C CR1 Register Bit Definitions, 185
 Peripheral_Registers_Bits_Definition, 851
I2C_CR1_SMBUS_Msk
 I2C CR1 Register Bit Definitions, 186
I2C_CR1_SMBUS_Pos
 I2C CR1 Register Bit Definitions, 186
I2C_CR1_START
 I2C CR1 Register Bit Definitions, 186
 Peripheral_Registers_Bits_Definition, 851
I2C_CR1_START_Msk
 I2C CR1 Register Bit Definitions, 186
I2C_CR1_START_Pos
 I2C CR1 Register Bit Definitions, 186
I2C_CR1_STOP
 I2C CR1 Register Bit Definitions, 186
 Peripheral_Registers_Bits_Definition, 851
I2C_CR1_STOP_Msk
 I2C CR1 Register Bit Definitions, 186
I2C_CR1_STOP_Pos
 I2C CR1 Register Bit Definitions, 187
I2C_CR1_SWRST
 I2C CR1 Register Bit Definitions, 187
 Peripheral_Registers_Bits_Definition, 851
I2C_CR1_SWRST_Msk
 I2C CR1 Register Bit Definitions, 187
 Peripheral_Registers_Bits_Definition, 852
I2C_CR1_SWRST_Pos
 I2C CR1 Register Bit Definitions, 187
I2C_CR2_DMAEN
 I2C CR2 Register Bit Definitions, 188
 Peripheral_Registers_Bits_Definition, 852
I2C_CR2_DMAEN_Msk
 I2C CR2 Register Bit Definitions, 188
I2C_CR2_DMAEN_Pos
 I2C CR2 Register Bit Definitions, 188
I2C_CR2_FREQ
 I2C CR2 Register Bit Definitions, 189
 Peripheral_Registers_Bits_Definition, 852
I2C_CR2_FREQ_0
 Peripheral_Registers_Bits_Definition, 852
I2C_CR2_FREQ_1
 Peripheral_Registers_Bits_Definition, 852
I2C_CR2_FREQ_2
 Peripheral_Registers_Bits_Definition, 852
I2C_CR2_FREQ_3
 Peripheral_Registers_Bits_Definition, 852
I2C_CR2_FREQ_4
 Peripheral_Registers_Bits_Definition, 852
I2C_CR2_FREQ_5
 Peripheral_Registers_Bits_Definition, 853
I2C_CR2_FREQ_Msk
 I2C CR2 Register Bit Definitions, 189
I2C_CR2_FREQ_Pos
 I2C CR2 Register Bit Definitions, 189
I2C_CR2_ITBUFEN
 I2C CR2 Register Bit Definitions, 189
 Peripheral_Registers_Bits_Definition, 853
I2C_CR2_ITBUFEN_Msk
 I2C CR2 Register Bit Definitions, 189
I2C_CR2_ITBUFEN_Pos
 I2C CR2 Register Bit Definitions, 189
I2C_CR2_ITERREN
 I2C CR2 Register Bit Definitions, 189
 Peripheral_Registers_Bits_Definition, 853
I2C_CR2_ITERREN_Msk
 I2C CR2 Register Bit Definitions, 190
I2C_CR2_ITERREN_Pos
 I2C CR2 Register Bit Definitions, 190
I2C_CR2_IteVTEN
 I2C CR2 Register Bit Definitions, 190
 Peripheral_Registers_Bits_Definition, 853
I2C_CR2_IteVTEN_Msk
 I2C CR2 Register Bit Definitions, 190
I2C_CR2_IteVTEN_Pos
 I2C CR2 Register Bit Definitions, 190
I2C_CR2_LAST
 I2C CR2 Register Bit Definitions, 190
 Peripheral_Registers_Bits_Definition, 853
I2C_CR2_LAST_Msk
 I2C CR2 Register Bit Definitions, 190
I2C_CR2_LAST_Pos
 I2C CR2 Register Bit Definitions, 191
I2C_Direction

I2C Driver, 270
I2C_Direction_Receiver
I2C Driver, 270
I2C_Direction_Transmitter
I2C Driver, 270
I2C_DR_DR
Peripheral_Registers_Bits_Definition, 853
I2C_ENGC_Disable
I2C General Call Enable Definitions, 276
I2C_ENGC_Enable
I2C General Call Enable Definitions, 277
I2C_Error_AF
I2C Driver, 270
I2C_Ev_Address_Matched
I2C Driver, 270
I2C_Ev_Data_RCV
I2C Driver, 270
I2C_Ev_Data_Req
I2C Driver, 270
I2C_EV_Stop
I2C Driver, 270
I2C_Flag_Busy
I2C Driver, 271
I2C_FLTR_ANOFF
Peripheral_Registers_Bits_Definition, 853
I2C_FLTR_DNF
Peripheral_Registers_Bits_Definition, 854
I2C_Generate_Start
API Function Prototypes, 278
I2C_Generate_Stop
API Function Prototypes, 279
I2C_Get_FlagStatus
API Function Prototypes, 279
I2C_Mode
S_I2C_Config_t, 1040
I2C_Mode_I2C
I2C Mode Definitions, 274
I2C_Mode_SMBus
I2C Mode Definitions, 274
I2C_OAR1_ADD0
Peripheral_Registers_Bits_Definition, 854
I2C_OAR1_ADD1
Peripheral_Registers_Bits_Definition, 854
I2C_OAR1_ADD1_7
Peripheral_Registers_Bits_Definition, 854
I2C_OAR1_ADD2
Peripheral_Registers_Bits_Definition, 854
I2C_OAR1_ADD3
Peripheral_Registers_Bits_Definition, 854
I2C_OAR1_ADD4
Peripheral_Registers_Bits_Definition, 854
I2C_OAR1_ADD5
Peripheral_Registers_Bits_Definition, 854
I2C_OAR1_ADD6
Peripheral_Registers_Bits_Definition, 855
I2C_OAR1_ADD7
Peripheral_Registers_Bits_Definition, 855
I2C_OAR1_ADD8
Peripheral_Registers_Bits_Definition, 855
I2C_OAR1_ADD8_9
Peripheral_Registers_Bits_Definition, 855
I2C_OAR1_ADD9
Peripheral_Registers_Bits_Definition, 855
I2C_OAR1_ADDMODE
Peripheral_Registers_Bits_Definition, 855
I2C_OAR2_ADD2
I2C OAR2 Register Bit Definitions, 191
Peripheral_Registers_Bits_Definition, 855
I2C_OAR2_ADD2_Msk
I2C OAR2 Register Bit Definitions, 192
I2C_OAR2_ADD2_Pos
I2C OAR2 Register Bit Definitions, 192
I2C_OAR2_ENDUAL
I2C OAR2 Register Bit Definitions, 192
Peripheral_Registers_Bits_Definition, 855
I2C_OAR2_ENDUAL_Msk
I2C OAR2 Register Bit Definitions, 192
I2C_OAR2_ENDUAL_Pos
I2C OAR2 Register Bit Definitions, 192
I2C_SCK_FM_200K
I2C Speed Definitions, 272
I2C_SCK_FM_400K
I2C Speed Definitions, 272
I2C_SCK_SM_100K
I2C Speed Definitions, 272
I2C_SCK_SM_50K
I2C Speed Definitions, 272
I2C_SendAddress
API Function Prototypes, 280
I2C_Slave_address
S_I2C_Config_t, 1040
I2C_Speed
S_I2C_Config_t, 1040
I2C_SR1_ADD10
I2C SR1 Register Bit Definitions, 193
Peripheral_Registers_Bits_Definition, 856
I2C_SR1_ADD10_Msk
I2C SR1 Register Bit Definitions, 194
I2C_SR1_ADD10_Pos
I2C SR1 Register Bit Definitions, 194
I2C_SR1_ADDR
I2C SR1 Register Bit Definitions, 194
Peripheral_Registers_Bits_Definition, 856
I2C_SR1_ADDR_Msk
I2C SR1 Register Bit Definitions, 194
I2C_SR1_ADDR_Pos
I2C SR1 Register Bit Definitions, 194
I2C_SR1_AF
I2C SR1 Register Bit Definitions, 194
Peripheral_Registers_Bits_Definition, 856
I2C_SR1_AF_Msk
I2C SR1 Register Bit Definitions, 194
I2C_SR1_AF_Pos
I2C SR1 Register Bit Definitions, 194
I2C_SR1_ARLO
I2C SR1 Register Bit Definitions, 195

Peripheral_Registers_Bits_Definition, 856
I2C_SR1_ARLO_Msk
 I2C SR1 Register Bit Definitions, 195
I2C_SR1_ARLO_Pos
 I2C SR1 Register Bit Definitions, 195
I2C_SR1_BERR
 I2C SR1 Register Bit Definitions, 195
 Peripheral_Registers_Bits_Definition, 856
I2C_SR1_BERR_Msk
 I2C SR1 Register Bit Definitions, 195
I2C_SR1_BERR_Pos
 I2C SR1 Register Bit Definitions, 195
I2C_SR1_BTF
 I2C SR1 Register Bit Definitions, 195
 Peripheral_Registers_Bits_Definition, 856
I2C_SR1_BTF_Msk
 I2C SR1 Register Bit Definitions, 195
I2C_SR1_BTF_Pos
 I2C SR1 Register Bit Definitions, 196
I2C_SR1_OVR
 I2C SR1 Register Bit Definitions, 196
 Peripheral_Registers_Bits_Definition, 856
I2C_SR1_OVR_Msk
 I2C SR1 Register Bit Definitions, 196
I2C_SR1_OVR_Pos
 I2C SR1 Register Bit Definitions, 196
I2C_SR1_PECERR
 I2C SR1 Register Bit Definitions, 196
 Peripheral_Registers_Bits_Definition, 857
I2C_SR1_PECERR_Msk
 I2C SR1 Register Bit Definitions, 196
I2C_SR1_PECERR_Pos
 I2C SR1 Register Bit Definitions, 196
I2C_SR1_RXNE
 I2C SR1 Register Bit Definitions, 196
 Peripheral_Registers_Bits_Definition, 857
I2C_SR1_RXNE_Msk
 I2C SR1 Register Bit Definitions, 197
I2C_SR1_RXNE_Pos
 I2C SR1 Register Bit Definitions, 197
I2C_SR1_SB
 I2C SR1 Register Bit Definitions, 197
 Peripheral_Registers_Bits_Definition, 857
I2C_SR1_SB_Msk
 I2C SR1 Register Bit Definitions, 197
I2C_SR1_SB_Pos
 I2C SR1 Register Bit Definitions, 197
I2C_SR1_SMBALERT
 I2C SR1 Register Bit Definitions, 197
 Peripheral_Registers_Bits_Definition, 857
I2C_SR1_SMBALERT_Msk
 I2C SR1 Register Bit Definitions, 197
I2C_SR1_SMBALERT_Pos
 I2C SR1 Register Bit Definitions, 198
I2C_SR1_STOPF
 I2C SR1 Register Bit Definitions, 198
 Peripheral_Registers_Bits_Definition, 857
I2C_SR1_STOPF_Msk
 I2C SR1 Register Bit Definitions, 198
I2C_SR1_STOPF_Pos
 I2C SR1 Register Bit Definitions, 198
I2C_SR1_TIMEOUT
 I2C SR1 Register Bit Definitions, 198
 Peripheral_Registers_Bits_Definition, 857
I2C_SR1_TIMEOUT_Msk
 I2C SR1 Register Bit Definitions, 198
I2C_SR1_TIMEOUT_Pos
 I2C SR1 Register Bit Definitions, 198
I2C_SR1_TXE
 I2C SR1 Register Bit Definitions, 198
 Peripheral_Registers_Bits_Definition, 857
I2C_SR1_TXE_Msk
 I2C SR1 Register Bit Definitions, 199
I2C_SR1_TXE_Pos
 I2C SR1 Register Bit Definitions, 199
I2C_SR2_BUSY
 I2C SR2 Register Bit Definitions, 200
 Peripheral_Registers_Bits_Definition, 858
I2C_SR2_BUSY_Msk
 I2C SR2 Register Bit Definitions, 200
I2C_SR2_BUSY_Pos
 I2C SR2 Register Bit Definitions, 200
I2C_SR2_DUALF
 I2C SR2 Register Bit Definitions, 200
 Peripheral_Registers_Bits_Definition, 858
I2C_SR2_DUALF_Msk
 I2C SR2 Register Bit Definitions, 200
I2C_SR2_DUALF_Pos
 I2C SR2 Register Bit Definitions, 200
I2C_SR2_GENCALL
 I2C SR2 Register Bit Definitions, 201
 Peripheral_Registers_Bits_Definition, 858
I2C_SR2_GENCALL_Msk
 I2C SR2 Register Bit Definitions, 201
I2C_SR2_GENCALL_Pos
 I2C SR2 Register Bit Definitions, 201
I2C_SR2_MSL
 I2C SR2 Register Bit Definitions, 201
 Peripheral_Registers_Bits_Definition, 858
I2C_SR2_MSL_Msk
 I2C SR2 Register Bit Definitions, 201
I2C_SR2_MSL_Pos
 I2C SR2 Register Bit Definitions, 201
I2C_SR2_PEC
 I2C SR2 Register Bit Definitions, 201
 Peripheral_Registers_Bits_Definition, 858
I2C_SR2_PEC_Msk
 I2C SR2 Register Bit Definitions, 202
I2C_SR2_PEC_Pos
 I2C SR2 Register Bit Definitions, 202
I2C_SR2_SMBDEFAULT
 I2C SR2 Register Bit Definitions, 202
 Peripheral_Registers_Bits_Definition, 858
I2C_SR2_SMBDEFAULT_Msk
 I2C SR2 Register Bit Definitions, 202
I2C_SR2_SMBDEFAULT_Pos

I2C_SR2_Register Bit Definitions, [202](#)
I2C_SR2_SMBHOST
 I2C SR2 Register Bit Definitions, [202](#)
 Peripheral_Registers_Bits_Definition, [858](#)
I2C_SR2_SMBHOST_Msk
 I2C SR2 Register Bit Definitions, [202](#)
I2C_SR2_SMBHOST_Pos
 I2C SR2 Register Bit Definitions, [202](#)
I2C_SR2_TRA
 I2C SR2 Register Bit Definitions, [203](#)
 Peripheral_Registers_Bits_Definition, [859](#)
I2C_SR2_TRA_Msk
 I2C SR2 Register Bit Definitions, [203](#)
I2C_SR2_TRA_Pos
 I2C SR2 Register Bit Definitions, [203](#)
I2C_stretchmode
 S_I2C_Config_t, [1040](#)
I2C_StretchMode_disabled
 I2C Stretch Mode Definitions, [273](#)
I2C_StretchMode_enabled
 I2C Stretch Mode Definitions, [273](#)
I2C_TRISE_TRISE
 I2C TRISE Register Bit Definitions, [205](#)
 Peripheral_Registers_Bits_Definition, [859](#)
I2C_TRISE_TRISE_Msk
 I2C TRISE Register Bit Definitions, [205](#)
I2C_TRISE_TRISE_Pos
 I2C TRISE Register Bit Definitions, [205](#)
I2C_TypeDef, 1019
 CCR, [1020](#)
 CR1, [1020](#)
 CR2, [1020](#)
 DR, [1020](#)
 OAR1, [1020](#)
 OAR2, [1020](#)
 SR1, [1020](#)
 SR2, [1021](#)
 TRISE, [1021](#)
I2SCFGR
 CMSIS, [488](#)
 SPI_TypeDef, [1051](#)
I2SPR
 CMSIS, [488](#)
 SPI_TypeDef, [1051](#)
IABR
 ITM Functions, [135](#)
 NVIC_TypeDef, [1030](#)
ICER
 ITM Functions, [135](#)
 NVIC_TypeDef, [1030](#)
ICPR
 ITM Functions, [135](#)
 NVIC_TypeDef, [1030](#)
ICR
 CMSIS, [488, 489](#)
ICSR
 ITM Functions, [135](#)
 SCB_TypeDef, [1047](#)
ICTR
 ITM Functions, [135](#)
ID_Check_Result
 Enum representing the result of ID check., [360](#)
ID_Found
 Enum representing the result of ID check., [360](#)
ID_NOT_Found
 Enum representing the result of ID check., [360](#)
IDCODE
 CMSIS, [489](#)
Idle_STATE
 Application States, [352](#)
IDR
 CMSIS, [489](#)
 CRC_TypeDef, [1005](#)
 GPIO_TypeDef, [1017](#)
IER
 CMSIS, [489, 490](#)
IFCR
 CMSIS, [490](#)
IMCR
 ITM Functions, [135](#)
IMR
 CMSIS, [490](#)
 EXTI_TypeDef, [1013](#)
IMSCR
 CMSIS, [490](#)
Inc/app_states.h, 1223
Inc/ecu.h, 1225
Includes, 354
INIT
 CRC_TypeDef, [1005](#)
Init_STATE
 Application States, [352](#)
Instrumentation Trace Macrocell (ITM), 73
 ITM_IMCR_INTEGRATION_Msk, [74](#)
 ITM_IMCR_INTEGRATION_Pos, [74](#)
 ITM_IRR_ATREADYM_Msk, [75](#)
 ITM_IRR_ATREADYM_Pos, [75](#)
 ITM_IWR_ATVALIDDM_Msk, [75](#)
 ITM_IWR_ATVALIDDM_Pos, [75](#)
 ITM_LSR_Access_Msk, [75](#)
 ITM_LSR_Access_Pos, [75](#)
 ITM_LSR_BytAcc_Msk, [75](#)
 ITM_LSR_BytAcc_Pos, [75](#)
 ITM_LSR_Present_Msk, [76](#)
 ITM_LSR_Present_Pos, [76](#)
 ITM_TCR_BUSY_Msk, [76](#)
 ITM_TCR_BUSY_Pos, [76](#)
 ITM_TCR_DWTENA_Msk, [76](#)
 ITM_TCR_DWTENA_Pos, [76](#)
 ITM_TCR_GTSFREQ_Msk, [76](#)
 ITM_TCR_GTSFREQ_Pos, [76](#)
 ITM_TCR_ITMENA_Msk, [77](#)
 ITM_TCR_ITMENA_Pos, [77](#)
 ITM_TCR_SWOENA_Msk, [77](#)
 ITM_TCR_SWOENA_Pos, [77](#)
 ITM_TCR_SYNCENA_Msk, [77](#)

ITM_TCR_SYNCENA_Pos, 77
ITM_TCR_TraceBusID_Msk, 77
ITM_TCR_TraceBusID_Pos, 77
ITM_TCR_TSENA_Msk, 78
ITM_TCR_TSENA_Pos, 78
ITM_TCR_TSPPrescale_Msk, 78
ITM_TCR_TSPPrescale_Pos, 78
ITM_TPR_PRIVMASK_Msk, 78
ITM_TPR_PRIVMASK_Pos, 78
Interrupt Priorities, 289
Interrupt Requests, 290
Interrupt Status, 288
interrupt_config
 STK_config_t, 1052
IP
 ITM Functions, 135
 NVIC_TypeDef, 1030
IPSR_ISR_Msk
 CMSIS, 417
 Status and Control Registers, 46
IPSR_ISR_Pos
 CMSIS, 417
 Status and Control Registers, 46
IPSR_Type, 1021
IRQ_Enable
 S_SPI_Config_t, 1044
IRQn
 Configuration_section_for_CMSIS, 524
IRR
 ITM Functions, 136
ISAR
 ITM Functions, 136
ISER
 ITM Functions, 136
 NVIC_TypeDef, 1030
ISPR
 ITM Functions, 136
 NVIC_TypeDef, 1030
ISR
 CMSIS, 490, 491
 ITM Functions, 136, 137
IT
 ITM Functions, 137
ITATBCTR0
 ITM Functions, 137
ITATBCTR2
 ITM Functions, 137
ITCTRL
 ITM Functions, 137
ITM
 CMSIS, 417
 Core Definitions, 109
ITM Functions, 119
 _reserved0, 124, 125
 _reserved1, 125, 126
ACPR, 126
ACTLR, 126
ADR, 126
AFSR, 126
AIRCR, 127
b, 127
BFAR, 127
C, 127, 128
CALIB, 128
CCR, 128
CFSR, 128
CID0, 128
CID1, 129
CID2, 129
CID3, 129
CLAIMCLR, 129
CLAIMSET, 129
COMP0, 129
COMP1, 129
COMP2, 130
COMP3, 130
CPACR, 130
CPICNT, 130
CPUID, 130
CSPSR, 130
CTRL, 130, 131
CYCCNT, 131
DCRDR, 131
DCRSR, 131
DEMCR, 131
DEVID, 131
DEVTYPE, 131
DFR, 132
DFSR, 132
DHCSR, 132
EXCCNT, 132
FFCR, 132
FFSR, 132
FIFO0, 132
FIFO1, 133
FOLDCNT, 133
FPCA, 133
FSCR, 133
FUNCTION0, 133
FUNCTION1, 133
FUNCTION2, 134
FUNCTION3, 134
GE, 134
HFSR, 134
IABR, 135
ICER, 135
ICPR, 135
ICSR, 135
ICTR, 135
IMCR, 135
IP, 135
IRR, 136
ISAR, 136
ISER, 136
ISPR, 136
ISR, 136, 137

IT, 137
 ITATBCTR0, 137
 ITATBCTR2, 137
 ITCTRL, 137
 ITM_CheckChar, 123
 ITM_ReceiveChar, 123
 ITM_RxBuffer, 137
 ITM_RXBUFFER_EMPTY, 123
 ITM_SendChar, 124
 IWR, 138
 LAR, 138
 LOAD, 138
 LSR, 138
 LSUCNT, 138
 MASK0, 138
 MASK1, 138
 MASK2, 139
 MASK3, 139
 MMFAR, 139
 MMFR, 139
 N, 139, 140
 nPRIV, 140
 PCSR, 140
 PFR, 140
 PID0, 140
 PID1, 140
 PID2, 141
 PID3, 141
 PID4, 141
 PID5, 141
 PID6, 141
 PID7, 141
 PORT, 141
 Q, 141, 142
 SCR, 142
 SHCSR, 142
 SHP, 142
 SLEEPCNT, 142
 SPPR, 143
 SPSEL, 143
 SSPSR, 143
 STIR, 143
 T, 143
 TCR, 144
 TER, 144
 TPR, 144
 TRIGGER, 144
 u16, 144
 u32, 144, 145
 u8, 145
 V, 145
 VAL, 146
 VTOR, 146
 w, 146
 Z, 146, 147
 ITM_BASE
 CMSIS, 418
 Core Definitions, 109
 ITM_CheckChar
 CMSIS, 453
 ITM Functions, 123
 ITM_IMCR_INTEGRATION_Msk
 CMSIS, 418
 Instrumentation Trace Macrocell (ITM), 74
 ITM_IMCR_INTEGRATION_Pos
 CMSIS, 418
 Instrumentation Trace Macrocell (ITM), 74
 ITM_IRR_ATREADYM_Msk
 CMSIS, 418
 Instrumentation Trace Macrocell (ITM), 75
 ITM_IRR_ATREADYM_Pos
 CMSIS, 418
 Instrumentation Trace Macrocell (ITM), 75
 ITM_IWR_ATVALIDM_Msk
 CMSIS, 418
 Instrumentation Trace Macrocell (ITM), 75
 ITM_IWR_ATVALIDM_Pos
 CMSIS, 418
 Instrumentation Trace Macrocell (ITM), 75
 ITM_LSR_Access_Msk
 CMSIS, 419
 Instrumentation Trace Macrocell (ITM), 75
 ITM_LSR_Access_Pos
 CMSIS, 419
 Instrumentation Trace Macrocell (ITM), 75
 ITM_LSR_BytAcc_Msk
 CMSIS, 419
 Instrumentation Trace Macrocell (ITM), 75
 ITM_LSR_BytAcc_Pos
 CMSIS, 419
 Instrumentation Trace Macrocell (ITM), 75
 ITM_LSR_Present_Msk
 CMSIS, 419
 Instrumentation Trace Macrocell (ITM), 76
 ITM_LSR_Present_Pos
 CMSIS, 419
 Instrumentation Trace Macrocell (ITM), 76
 ITM_ReceiveChar
 CMSIS, 453
 ITM Functions, 123
 ITM_RxBuffer
 CMSIS, 491
 ITM Functions, 137
 ITM_RXBUFFER_EMPTY
 CMSIS, 419
 ITM Functions, 123
 ITM_SendChar
 CMSIS, 453
 ITM Functions, 124
 ITM_TCR_BUSY_Msk
 CMSIS, 419
 Instrumentation Trace Macrocell (ITM), 76
 ITM_TCR_BUSY_Pos
 CMSIS, 420
 Instrumentation Trace Macrocell (ITM), 76
 ITM_TCR_DWTENA_Msk

CMSIS, 420
Instrumentation Trace Macrocell (ITM), 76

ITM_TCR_DWTENA_Pos
CMSIS, 420
Instrumentation Trace Macrocell (ITM), 76

ITM_TCR_GTSFREQ_Msk
CMSIS, 420
Instrumentation Trace Macrocell (ITM), 76

ITM_TCR_GTSFREQ_Pos
CMSIS, 420
Instrumentation Trace Macrocell (ITM), 76

ITM_TCR_ITMENA_Msk
CMSIS, 420
Instrumentation Trace Macrocell (ITM), 77

ITM_TCR_ITMENA_Pos
CMSIS, 420
Instrumentation Trace Macrocell (ITM), 77

ITM_TCR_SWOENA_Msk
CMSIS, 420
Instrumentation Trace Macrocell (ITM), 77

ITM_TCR_SWOENA_Pos
CMSIS, 421
Instrumentation Trace Macrocell (ITM), 77

ITM_TCR_SYNCENA_Msk
CMSIS, 421
Instrumentation Trace Macrocell (ITM), 77

ITM_TCR_SYNCENA_Pos
CMSIS, 421
Instrumentation Trace Macrocell (ITM), 77

ITM_TCR_TraceBusID_Msk
CMSIS, 421
Instrumentation Trace Macrocell (ITM), 77

ITM_TCR_TraceBusID_Pos
CMSIS, 421
Instrumentation Trace Macrocell (ITM), 77

ITM_TCR_TSENA_Msk
CMSIS, 421
Instrumentation Trace Macrocell (ITM), 78

ITM_TCR_TSENA_Pos
CMSIS, 421
Instrumentation Trace Macrocell (ITM), 78

ITM_TCR_TSPrescale_Msk
CMSIS, 421
Instrumentation Trace Macrocell (ITM), 78

ITM_TCR_TSPrescale_Pos
CMSIS, 422
Instrumentation Trace Macrocell (ITM), 78

ITM_TPR_PRIVMASK_Msk
CMSIS, 422
Instrumentation Trace Macrocell (ITM), 78

ITM_TPR_PRIVMASK_Pos
CMSIS, 422
Instrumentation Trace Macrocell (ITM), 78

ITM_Type, 1022

IV0LR
CMSIS, 491

IV0RR
CMSIS, 491

IV1LR
CMSIS, 491

IV1RR
CMSIS, 491

IWDG_KR_KEY
Peripheral_Registers_Bits_Definition, 859

IWDG_PR_PR
Peripheral_Registers_Bits_Definition, 859

IWDG_PR_PR_0
Peripheral_Registers_Bits_Definition, 859

IWDG_PR_PR_1
Peripheral_Registers_Bits_Definition, 859

IWDG_PR_PR_2
Peripheral_Registers_Bits_Definition, 859

IWDG_RLR_RL
Peripheral_Registers_Bits_Definition, 860

IWDG_SR_PVU
Peripheral_Registers_Bits_Definition, 860

IWDG_SR_RVU
Peripheral_Registers_Bits_Definition, 860

IWDG_TypeDef, 1023

IWR
ITM Functions, 138

JDR1
CMSIS, 492

JDR2
CMSIS, 492

JDR3
CMSIS, 492

JDR4
CMSIS, 492

JOFR1
CMSIS, 492

JOFR2
CMSIS, 492

JOFR3
CMSIS, 492

JOFR4
CMSIS, 492

JSQR
CMSIS, 493

K0LR
CMSIS, 493

K0RR
CMSIS, 493

K1LR
CMSIS, 493

K1RR
CMSIS, 493

K2LR
CMSIS, 493

K2RR
CMSIS, 493

K3LR
CMSIS, 493

K3RR
CMSIS, 494

KEYPAD Driver, 18
 KEYPAD_PORT, 19
keypad_Get_Pressed_Key
 API Function Prototypes, 20
keypad_init
 API Function Prototypes, 21
KEYPAD_PORT
 KEYPAD Driver, 19
KEYR
 CMSIS, 494
KR
 CMSIS, 494

LAR
 ITM Functions, 138
LCD Commands, 23
LCD Driver, 22
 LCD_2ROWS, 23
 LCD_4BIT, 23
 LCD_4ROWS, 23
 LCD_8BIT, 23
 LCD_MODE_t, 22
 LCD_ROWS_t, 23
LCD Row Positions, 24
LCD_2ROWS
 LCD Driver, 23
LCD_4BIT
 LCD Driver, 23
LCD_4ROWS
 LCD Driver, 23
LCD_8BIT
 LCD Driver, 23
LCD_Init
 API Function Prototypes, 25
LCD_MODE_t
 LCD Driver, 22
LCD_ROWS_t
 LCD Driver, 23
LCD_Send_Char
 API Function Prototypes, 26
LCD_Send_Char_Pos
 API Function Prototypes, 27
LCD_Send_Command
 API Function Prototypes, 27
LCD_Send_Enable_Signal
 API Function Prototypes, 28
LCD_Send_String
 API Function Prototypes, 29
LCD_Send_String_Pos
 API Function Prototypes, 30
LCD_Set_Cursor
 API Function Prototypes, 30
LCD_t, 1023
 D0_PIN, 1024
 D1_PIN, 1024
 D2_PIN, 1024
 D3_PIN, 1024
 D4_PIN, 1025
 D5_PIN, 1025

D6_PIN, 1025
D7_PIN, 1025
Display_Mode, 1025
EN_PIN, 1025
Entry_Mode, 1025
GPIO_PORT, 1025
Mode, 1026
Rows, 1026
RS_PIN, 1026

LCKR
 CMSIS, 494
 GPIO_TypeDef, 1017
LED Driver, 31
LED_Active_High
 Enum for specifying the LED mode., 33
LED_Active_Low
 Enum for specifying the LED mode., 33
LED_cfg_t, 1026
 LED_Mode, 1027
 LED_Pin, 1027
 LED_Port, 1027
LED_Init
 API Function Prototypes, 34
LED_Mode
 LED_cfg_t, 1027
LED_Mode_max
 Enum for specifying the LED mode., 33
LED_Mode_t
 Enum for specifying the LED mode., 33
LED_Pin
 LED_cfg_t, 1027
LED_Port
 LED_cfg_t, 1027
LED_Toggle
 API Function Prototypes, 35
LED_TurnOff
 API Function Prototypes, 35
LED_TurnOn
 API Function Prototypes, 36
Library_configuration_section, 520
 __STM32F4XX_STDPERIPH_VERSION, 520
 __STM32F4XX_STDPERIPH_VERSION_MAIN,
 521
 __STM32F4XX_STDPERIPH_VERSION_RC, 521
 __STM32F4XX_STDPERIPH_VERSION_SUB1,
 521
 __STM32F4XX_STDPERIPH_VERSION_SUB2,
 521
 HSE_STARTUP_TIMEOUT, 521
 HSE_VALUE, 521
 HSI_VALUE, 522

LIFCR
 CMSIS, 494
LIPCR
 CMSIS, 494
LISR
 CMSIS, 494
LOAD

ITM Functions, [138](#)
STK_TypeDef, [1053](#)

LOW_BYTE_FIRST
Platform Standard Types, [151](#)

LSB_FIRST
Platform Standard Types, [151](#)

LSR
ITM Functions, [138](#)

LSUCNT
ITM Functions, [138](#)

LTDC_AWCR_AAH
Peripheral_Registers_Bits_Definition, [860](#)

LTDC_AWCR_AAW
Peripheral_Registers_Bits_Definition, [860](#)

LTDC_BCCR_BCBLUE
Peripheral_Registers_Bits_Definition, [860](#)

LTDC_BCCR_BCGREEN
Peripheral_Registers_Bits_Definition, [860](#)

LTDC_BCCR_BCRED
Peripheral_Registers_Bits_Definition, [861](#)

LTDC_BPCR_AHBP
Peripheral_Registers_Bits_Definition, [861](#)

LTDC_BPCR_AVBP
Peripheral_Registers_Bits_Definition, [861](#)

LTDC_CDSR_HDES
Peripheral_Registers_Bits_Definition, [861](#)

LTDC_CDSR_HSYNCS
Peripheral_Registers_Bits_Definition, [861](#)

LTDC_CDSR_VDES
Peripheral_Registers_Bits_Definition, [861](#)

LTDC_CDSR_VSYNCS
Peripheral_Registers_Bits_Definition, [861](#)

LTDC_CPSR_CXPOS
Peripheral_Registers_Bits_Definition, [861](#)

LTDC_CPSR_CYPOS
Peripheral_Registers_Bits_Definition, [862](#)

LTDC_GCR_DBW
Peripheral_Registers_Bits_Definition, [862](#)

LTDC_GCR_DEN
Peripheral_Registers_Bits_Definition, [862](#)

LTDC_GCR_DEPOL
Peripheral_Registers_Bits_Definition, [862](#)

LTDC_GCR_DGW
Peripheral_Registers_Bits_Definition, [862](#)

LTDC_GCR_DRW
Peripheral_Registers_Bits_Definition, [862](#)

LTDC_GCR_HSPOL
Peripheral_Registers_Bits_Definition, [862](#)

LTDC_GCR_LTDCEN
Peripheral_Registers_Bits_Definition, [862](#)

LTDC_GCR_PCPOL
Peripheral_Registers_Bits_Definition, [863](#)

LTDC_GCR_VSPOL
Peripheral_Registers_Bits_Definition, [863](#)

LTDC_ICR_CFUIF
Peripheral_Registers_Bits_Definition, [863](#)

LTDC_ICR_CLIF
Peripheral_Registers_Bits_Definition, [863](#)

LTDC_ICR_CRRIF
Peripheral_Registers_Bits_Definition, [863](#)

LTDC_ICR_CTERRIF
Peripheral_Registers_Bits_Definition, [863](#)

LTDC_IER_FUIE
Peripheral_Registers_Bits_Definition, [863](#)

LTDC_IER_LIE
Peripheral_Registers_Bits_Definition, [863](#)

LTDC_IER_RRIE
Peripheral_Registers_Bits_Definition, [864](#)

LTDC_IER_TERRIE
Peripheral_Registers_Bits_Definition, [864](#)

LTDC_ISR_FUIF
Peripheral_Registers_Bits_Definition, [864](#)

LTDC_ISR_LIF
Peripheral_Registers_Bits_Definition, [864](#)

LTDC_ISR_RRIF
Peripheral_Registers_Bits_Definition, [864](#)

LTDC_ISR_TERRIF
Peripheral_Registers_Bits_Definition, [864](#)

LTDC_Layer_TypeDef, [1027](#)

LTDC_LIPCR_LIPOS
Peripheral_Registers_Bits_Definition, [864](#)

LTDC_LxBFCR_BF1
Peripheral_Registers_Bits_Definition, [864](#)

LTDC_LxBFCR_BF2
Peripheral_Registers_Bits_Definition, [865](#)

LTDC_LxCACR_CONSTA
Peripheral_Registers_Bits_Definition, [865](#)

LTDC_LxCFBAR_CFBADD
Peripheral_Registers_Bits_Definition, [865](#)

LTDC_LxCFBLNR_CFBLNBR
Peripheral_Registers_Bits_Definition, [865](#)

LTDC_LxCFBLR_CFBLL
Peripheral_Registers_Bits_Definition, [865](#)

LTDC_LxCFBLR_CFBP
Peripheral_Registers_Bits_Definition, [865](#)

LTDC_LxCKCR_CKBLUE
Peripheral_Registers_Bits_Definition, [865](#)

LTDC_LxCKCR_CKGREEN
Peripheral_Registers_Bits_Definition, [865](#)

LTDC_LxCKCR_CKRED
Peripheral_Registers_Bits_Definition, [866](#)

LTDC_LxCLUTWR_BLUE
Peripheral_Registers_Bits_Definition, [866](#)

LTDC_LxCLUTWR_CLUTADD
Peripheral_Registers_Bits_Definition, [866](#)

LTDC_LxCLUTWR_GREEN
Peripheral_Registers_Bits_Definition, [866](#)

LTDC_LxCLUTWR_RED
Peripheral_Registers_Bits_Definition, [866](#)

LTDC_LxCR_CLUTEN
Peripheral_Registers_Bits_Definition, [866](#)

LTDC_LxCR_COLKEN
Peripheral_Registers_Bits_Definition, [866](#)

LTDC_LxCR_LEN
Peripheral_Registers_Bits_Definition, [866](#)

LTDC_LxDCCR_DCALPHA

Peripheral_Registers_Bits_Definition, [867](#)
LTDC_LxDCCR_DCBLUE
 Peripheral_Registers_Bits_Definition, [867](#)
LTDC_LxDCCR_DCGREEN
 Peripheral_Registers_Bits_Definition, [867](#)
LTDC_LxDCCR_DCRED
 Peripheral_Registers_Bits_Definition, [867](#)
LTDC_LxPFPCR_PF
 Peripheral_Registers_Bits_Definition, [867](#)
LTDC_LxWHPCR_WHSPPOS
 Peripheral_Registers_Bits_Definition, [867](#)
LTDC_LxWHPCR_WHSTPOS
 Peripheral_Registers_Bits_Definition, [867](#)
LTDC_LxWVPCR_WVSPPOS
 Peripheral_Registers_Bits_Definition, [867](#)
LTDC_LxWVPCR_WVSTPOS
 Peripheral_Registers_Bits_Definition, [868](#)
LTDC_SRCR_IMR
 Peripheral_Registers_Bits_Definition, [868](#)
LTDC_SRCR_VBR
 Peripheral_Registers_Bits_Definition, [868](#)
LTDC_SSCR_HSW
 Peripheral_Registers_Bits_Definition, [868](#)
LTDC_SSCR_VSH
 Peripheral_Registers_Bits_Definition, [868](#)
LTDC_TWCR_TOTALH
 Peripheral_Registers_Bits_Definition, [868](#)
LTDC_TWCR_TOTALW
 Peripheral_Registers_Bits_Definition, [868](#)
LTDC_TypeDef, [1028](#)
LTR
 CMSIS, [495](#)
LWR
 CMSIS, [495](#)
M0AR
 CMSIS, [495](#)
M1AR
 CMSIS, [495](#)
 Macros for the servo motor directions, [38](#)
 Macros for USART Configuration, [345](#)
MASK
 CMSIS, [495](#)
MASK0
 ITM Functions, [138](#)
MASK1
 ITM Functions, [138](#)
MASK2
 ITM Functions, [139](#)
MASK3
 ITM Functions, [139](#)
Master_Byte_Transmitting
 I2C Driver, [271](#)
MCAL_GPIO_DelInit
 API Function Prototypes, [259](#)
MCAL_GPIO_Init
 API Function Prototypes, [260](#)
MCAL_GPIO_LockPin
 API Function Prototypes, [261](#)
MCAL_GPIO_ReadPin
 API Function Prototypes, [262](#)
MCAL_GPIO_ReadPort
 API Function Prototypes, [263](#)
MCAL_GPIO_TogglePin
 API Function Prototypes, [264](#)
MCAL_GPIO_TogglePort
 API Function Prototypes, [264](#)
MCAL_GPIO_WritePin
 API Function Prototypes, [265](#)
MCAL_GPIO_WritePort
 API Function Prototypes, [266](#)
MCAL_I2C_Deinit
 API Function Prototypes, [280](#)
MCAL_I2C_Init
 API Function Prototypes, [281](#)
MCAL_I2C_Master_Rx
 API Function Prototypes, [282](#)
MCAL_I2C_Master_Tx
 API Function Prototypes, [282](#)
MCAL_I2C_Set_GPIO
 API Function Prototypes, [283](#)
MCAL_I2C_Slave_SendData
 API Function Prototypes, [284](#)
MCAL_I2C_SlaveReceiveData
 API Function Prototypes, [284](#)
MCAL_NVIC_ClearPendingIRQ
 API Function Prototypes, [292](#)
MCAL_NVIC_DisableIRQ
 API Function Prototypes, [293](#)
MCAL_NVIC_EnableIRQ
 API Function Prototypes, [293](#)
MCAL_NVIC_GetActive
 API Function Prototypes, [294](#)
MCAL_NVIC_GetPendingIRQ
 API Function Prototypes, [295](#)
MCAL_NVIC_GetPriority
 API Function Prototypes, [295](#)
MCAL_NVIC_GetPriorityGrouping
 API Function Prototypes, [296](#)
MCAL_NVIC_SetPendingIRQ
 API Function Prototypes, [296](#)
MCAL_NVIC_SetPriority
 API Function Prototypes, [298](#)
MCAL_NVIC_SetPriorityGrouping
 API Function Prototypes, [299](#)
MCAL_NVIC_SystemReset
 API Function Prototypes, [299](#)
MCAL_RCC_Enable_Peripheral
 API Function Prototypes, [311](#)
MCAL_RCC_GetHCLKFreq
 API Function Prototypes, [311](#)
MCAL_RCC_GetPCLK1Freq
 API Function Prototypes, [312](#)
MCAL_RCC_GetPCLK2Freq
 API Function Prototypes, [312](#)
MCAL_RCC_GetSYS_CLKFreq
 API Function Prototypes, [313](#)

MCAL_RCC_Reset_Peripheral
 API Function Prototypes, 314

MCAL_RCC_Select_Clock
 API Function Prototypes, 314

MCAL_SPI_DelInit
 API Function Prototypes, 326

MCAL_SPI_GPIO_Set_Pins
 API Function Prototypes, 327

MCAL_SPI_Init
 API Function Prototypes, 328

MCAL_SPI_ReceiveData
 API Function Prototypes, 328

MCAL_SPI_SendData
 API Function Prototypes, 329

MCAL_SPI_Tx_Rx
 API Function Prototypes, 330

MCAL_STK_Config
 API Function Prototypes, 335

MCAL_STK_Delay
 API Function Prototypes, 336

MCAL_STK_Delay1ms
 API Function Prototypes, 336

MCAL_STK_SetCallback
 API Function Prototypes, 337

MCAL_STK_SetReload
 API Function Prototypes, 338

MCAL_STK_StartTimer
 API Function Prototypes, 338

MCAL_STK_StopTimer
 API Function Prototypes, 339

MCAL_USART_DelInit
 API Function Prototypes, 347

MCAL_USART_GPIO_Set_Pins
 API Function Prototypes, 347

MCAL_USART_Init
 API Function Prototypes, 348

MCAL_USART_ReceiveBuffer
 API Function Prototypes, 348

MCAL_USART_ReceiveData
 API Function Prototypes, 349

MCAL_USART_SendData
 API Function Prototypes, 349

MCAL_USART_SendString
 API Function Prototypes, 350

MCR
 CMSIS, 495

Memory Base Addresses, 157

MemoryManagement_IRQn
 Configuration_section_for\CMSIS, 524

MEMRMP
 CMSIS, 495

SYSCFG_RegDef_t, 1055

MISR
 CMSIS, 496

MMAR
 SCB_TypeDef, 1047

MMFAR
 ITM Functions, 139

MMFR
 ITM Functions, 139

Mode
 LCD_t, 1026

MODER
 CMSIS, 496

 GPIO_TypeDef, 1017

MSB_FIRST
 Platform Standard Types, 151

MSR
 CMSIS, 496

N
 ITM Functions, 139, 140

NDTR
 CMSIS, 496

Nested Vectored Interrupt Controller (NVIC), 49

 NVIC_STIR_INTID_Msk, 50

 NVIC_STIR_INTID_Pos, 50

NLR
 CMSIS, 496

NonMaskableInt_IRQn
 Configuration_section_for\CMSIS, 524

nPRIV
 ITM Functions, 140

NSS
 S_SPI_Config_t, 1044

NVIC
 CMSIS, 422

 Core Definitions, 110

 NVIC Instance, 169

 NVIC Configuration Macros, 287

 NVIC Driver, 285

 NVIC Functions, 112

 NVIC_ClearPendingIRQ, 113

 NVIC_DecodePriority, 113

 NVIC_DisableIRQ, 113

 NVIC_EnableIRQ, 114

 NVIC_EncodePriority, 114

 NVIC_GetActive, 115

 NVIC_GetPendingIRQ, 115

 NVIC_GetPriority, 115

 NVIC_GetPriorityGrouping, 116

 NVIC_SetPendingIRQ, 116

 NVIC_SetPriority, 116

 NVIC_SetPriorityGrouping, 117

 NVIC_SystemReset, 117

 NVIC Instance, 169

 NVIC, 169

 NVIC IRQ Enable/Disable Macros, 300

 NVIC IRQ10_EXTI4_Disable, 301

 NVIC IRQ10_EXTI4_Enable, 301

 NVIC IRQ23_EXTI5_9_Disable, 301

 NVIC IRQ23_EXTI5_9_Enable, 302

 NVIC IRQ31_I2C1_EV_Disable, 302

 NVIC IRQ31_I2C1_EV_Enable, 302

 NVIC IRQ32_I2C1_ER_Disable, 302

NVIC_IRQ32_I2C1_ER_Enable, 302
 NVIC_IRQ33_I2C2_EV_Disable, 302
 NVIC_IRQ33_I2C2_EV_Enable, 302
 NVIC_IRQ34_I2C2_ER_Enable, 303
 NVIC_IRQ35_SPI1_Disable, 303
 NVIC_IRQ35_SPI1_Enable, 303
 NVIC_IRQ36_SPI2_Enable, 303
 NVIC_IRQ37_USART1_Disable, 303
 NVIC_IRQ37_USART1_Enable, 303
 NVIC_IRQ38_USART2_Disable, 303
 NVIC_IRQ38_USART2_Enable, 304
 NVIC_IRQ39_USART3_Enable, 304
 NVIC_IRQ40_EXTI10_15_Enable, 304
 NVIC_IRQ6_EXTI0_Disable, 304
 NVIC_IRQ6_EXTI0_Enable, 304
 NVIC_IRQ7_EXTI1_Disable, 304
 NVIC_IRQ7_EXTI1_Enable, 304
 NVIC_IRQ8_EXTI2_Disable, 305
 NVIC_IRQ8_EXTI2_Enable, 305
 NVIC_IRQ9_EXTI3_Disable, 305
 NVIC_IRQ9_EXTI3_Enable, 305
 NVIC Register Definitions, 287
 NVIC Register Map, 161
 NVIC_BASE
 CMSIS, 422
 Core Definitions, 110
 NVIC_ClearPendingIRQ
 CMSIS, 454
 NVIC Functions, 113
 NVIC_DecodePriority
 CMSIS, 454
 NVIC Functions, 113
 NVIC_DisableIRQ
 CMSIS, 455
 NVIC Functions, 113
 NVIC_EnableIRQ
 CMSIS, 455
 NVIC Functions, 114
 NVIC_EncodePriority
 CMSIS, 455
 NVIC Functions, 114
 NVIC_GetActive
 CMSIS, 457
 NVIC Functions, 115
 NVIC_GetPendingIRQ
 CMSIS, 457
 NVIC Functions, 115
 NVIC_GetPriority
 CMSIS, 458
 NVIC Functions, 115
 NVIC_GetPriorityGrouping
 CMSIS, 458
 NVIC Functions, 116
 NVIC IRQ10_EXTI4_Disable
 NVIC IRQ Enable/Disable Macros, 301
 NVIC IRQ10_EXTI4_Enable
 NVIC IRQ Enable/Disable Macros, 301
 NVIC IRQ23_EXTI5_9_Disable
 NVIC IRQ Enable/Disable Macros, 301
 NVIC IRQ23_EXTI5_9_Enable
 NVIC IRQ Enable/Disable Macros, 302
 NVIC IRQ31_I2C1_EV_Disable
 NVIC IRQ Enable/Disable Macros, 302
 NVIC IRQ31_I2C1_EV_Enable
 NVIC IRQ Enable/Disable Macros, 302
 NVIC IRQ32_I2C1_ER_Disable
 NVIC IRQ Enable/Disable Macros, 302
 NVIC IRQ32_I2C1_ER_Enable
 NVIC IRQ Enable/Disable Macros, 302
 NVIC IRQ33_I2C2_EV_Disable
 NVIC IRQ Enable/Disable Macros, 302
 NVIC IRQ33_I2C2_EV_Enable
 NVIC IRQ Enable/Disable Macros, 302
 NVIC IRQ34_I2C2_ER_Disable
 NVIC IRQ Enable/Disable Macros, 302
 NVIC IRQ34_I2C2_ER_Enable
 NVIC IRQ Enable/Disable Macros, 303
 NVIC IRQ35_SPI1_Disable
 NVIC IRQ Enable/Disable Macros, 303
 NVIC IRQ35_SPI1_Enable
 NVIC IRQ Enable/Disable Macros, 303
 NVIC IRQ36_SPI2_Disable
 NVIC IRQ Enable/Disable Macros, 303
 NVIC IRQ37_USART1_Disable
 NVIC IRQ Enable/Disable Macros, 303
 NVIC IRQ37_USART1_Enable
 NVIC IRQ Enable/Disable Macros, 303
 NVIC IRQ38_USART2_Disable
 NVIC IRQ Enable/Disable Macros, 304
 NVIC IRQ38_USART2_Enable
 NVIC IRQ Enable/Disable Macros, 304
 NVIC IRQ39_USART3_Disable
 NVIC IRQ Enable/Disable Macros, 304
 NVIC IRQ39_USART3_Enable
 NVIC IRQ Enable/Disable Macros, 304
 NVIC IRQ40_EXTI10_15_Enable
 NVIC IRQ Enable/Disable Macros, 304
 NVIC IRQ6_EXTI0_Disable
 NVIC IRQ Enable/Disable Macros, 304
 NVIC IRQ6_EXTI0_Enable
 NVIC IRQ Enable/Disable Macros, 304
 NVIC IRQ7_EXTI1_Disable
 NVIC IRQ Enable/Disable Macros, 304
 NVIC IRQ7_EXTI1_Enable
 NVIC IRQ Enable/Disable Macros, 304
 NVIC IRQ8_EXTI2_Disable
 NVIC IRQ Enable/Disable Macros, 305
 NVIC IRQ8_EXTI2_Enable
 NVIC IRQ Enable/Disable Macros, 305
 NVIC IRQ9_EXTI3_Disable
 NVIC IRQ Enable/Disable Macros, 305
 NVIC IRQ9_EXTI3_Enable
 NVIC IRQ Enable/Disable Macros, 305
 NVIC_SetPendingIRQ
 CMSIS, 458
 NVIC Functions, 116
 NVIC_SetPriority
 CMSIS, 459
 NVIC Functions, 116
 NVIC_SetPriorityGrouping

CMSIS, 459
NVIC Functions, 117
NVIC_STIR_INTID_Msk
 CMSIS, 422
 Nested Vectored Interrupt Controller (NVIC), 50
NVIC_STIR_INTID_Pos
 CMSIS, 422
 Nested Vectored Interrupt Controller (NVIC), 50
NVIC_SystemReset
 CMSIS, 459
 NVIC Functions, 117
NVIC_Type, 1029
NVIC_TypeDef, 1029
 IABR, 1030
 ICER, 1030
 ICPR, 1030
 IP, 1030
 ISER, 1030
 ISPR, 1030
 RESERVED0, 1030
 RESERVED1, 1030
 RESERVED2, 1031
 RESERVED3, 1031
 RESERVED4, 1031
 RESERVED5, 1031
 STIR, 1031

OAR1
 CMSIS, 496
 I2C_TypeDef, 1020

OAR2
 CMSIS, 497
 I2C_TypeDef, 1020

OCOLR
 CMSIS, 497

ODR
 CMSIS, 497
 GPIO_TypeDef, 1017

OMAR
 CMSIS, 497

OOR
 CMSIS, 497

OPFCCR
 CMSIS, 497

OPTCR
 CMSIS, 497

OPTCR1
 CMSIS, 498

OPTKEYR
 CMSIS, 498

OR
 CMSIS, 498
 TIM1_TypeDef, 1059

OSPEEDR
 CMSIS, 498
 GPIO_TypeDef, 1017

OTG_FS IRQn
 Configuration_section_for_CMSIS, 526
OTG_FS_WKUP IRQn
 Configuration_section_for_CMSIS, 526

OTYPER
 CMSIS, 498
 GPIO_TypeDef, 1017

P_IRQ_CallBack
 S_SPI_Config_t, 1044
P_Slave_Event_CallBack
 S_I2C_Config_t, 1040

PAR
 CMSIS, 498

Payload_Length
 S_SPI_Config_t, 1044

PCSR
 ITM Functions, 140

PendSV_IRQn
 Configuration_section_for_CMSIS, 524

PERIPH_BASE
 Peripheral_memory_map, 535

PERIPH_BB_BASE
 Peripheral_memory_map, 535

Peripheral Instances, 167

Peripheral Register Definitions, 160

Peripheral Selection, 308
 RCC_CRC, 308
 RCC_GPIOA, 308
 RCC_GPIOB, 308
 RCC_GPIOC, 309
 RCC_GPIOD, 309
 RCC_GPIOE, 309
 RCC_I2C1, 309
 RCC_I2C2, 309
 RCC_SPI1, 309
 RCC_SPI2, 309
 RCC_SYSCFG, 309
 RCC_TIM2, 310
 RCC_USART1, 310
 RCC_USART2, 310
 RCC_USART6, 310

Peripheral_declaration, 536

Peripheral_memory_map, 532
 AHB2PERIPH_BASE, 534
 BKPSRAM_BASE, 534
 BKPSRAM_BB_BASE, 534
 CCMDATARAM_BASE, 534
 CCMDATARAM_BB_BASE, 535
 DMA2D_BASE, 535
 FLASH_BASE, 535
 GPIOA_BASE, 535
 PERIPH_BASE, 535
 PERIPH_BB_BASE, 535
 SRAM1_BASE, 535
 SRAM1_BB_BASE, 536
 SRAM_BB_BASE, 536
 UART8_BASE, 536

Peripheral_Registers_Bits_Definition, 539
 ADC_CCR_ADCPRE, 623
 ADC_CCR_ADCPRE_0, 623
 ADC_CCR_ADCPRE_1, 623

ADC_CCR_DDS, 624
ADC_CCR_DELAY, 624
ADC_CCR_DELAY_0, 624
ADC_CCR_DELAY_1, 624
ADC_CCR_DELAY_2, 624
ADC_CCR_DELAY_3, 624
ADC_CCR_DMA, 624
ADC_CCR_DMA_0, 624
ADC_CCR_DMA_1, 625
ADC_CCR_MULTI, 625
ADC_CCR_MULTI_0, 625
ADC_CCR_MULTI_1, 625
ADC_CCR_MULTI_2, 625
ADC_CCR_MULTI_3, 625
ADC_CCR_MULTI_4, 625
ADC_CCR_TSVREFE, 625
ADC_CCR_VBATE, 626
ADC_CDR_DATA1, 626
ADC_CDR_DATA2, 626
ADC_CR1_AWDCH, 626
ADC_CR1_AWDCH_0, 626
ADC_CR1_AWDCH_1, 626
ADC_CR1_AWDCH_2, 626
ADC_CR1_AWDCH_3, 626
ADC_CR1_AWDCH_4, 627
ADC_CR1_AWDEN, 627
ADC_CR1_AWDIE, 627
ADC_CR1_AWDSGL, 627
ADC_CR1_DISCEN, 627
ADC_CR1_DISCNUM, 627
ADC_CR1_DISCNUM_0, 627
ADC_CR1_DISCNUM_1, 628
ADC_CR1_DISCNUM_2, 628
ADC_CR1_EOCIE, 628
ADC_CR1_JAUTO, 628
ADC_CR1_JAWDEN, 628
ADC_CR1_JDISCEN, 628
ADC_CR1_JEOCIE, 628
ADC_CR1_OVRIE, 629
ADC_CR1_RES, 629
ADC_CR1_RES_0, 629
ADC_CR1_RES_1, 629
ADC_CR1_SCAN, 629
ADC_CR2_ADON, 629
ADC_CR2_ALIGN, 629
ADC_CR2_CONT, 630
ADC_CR2_DDS, 630
ADC_CR2_DMA, 630
ADC_CR2_EOCS, 630
ADC_CR2_EXTEN, 630
ADC_CR2_EXTEN_0, 630
ADC_CR2_EXTEN_1, 630
ADC_CR2_EXTSEL, 631
ADC_CR2_EXTSEL_0, 631
ADC_CR2_EXTSEL_1, 631
ADC_CR2_EXTSEL_2, 631
ADC_CR2_EXTSEL_3, 631
ADC_CR2_JEXTEN, 631
ADC_CR2_JEXTEN_0, 631
ADC_CR2_JEXTEN_1, 631
ADC_CR2_JEXTSEL, 632
ADC_CR2_JEXTSEL_0, 632
ADC_CR2_JEXTSEL_1, 632
ADC_CR2_JEXTSEL_2, 632
ADC_CR2_JEXTSEL_3, 632
ADC_CR2_JSWSTART, 632
ADC_CR2_SWSTART, 632
ADC_CSR_AWD1, 632
ADC_CSR_AWD2, 633
ADC_CSR_AWD3, 633
ADC_CSR_EOC1, 633
ADC_CSR_EOC2, 633
ADC_CSR_EOC3, 633
ADC_CSR_JEOC1, 633
ADC_CSR_JEOC2, 633
ADC_CSR_JEOC3, 633
ADC_CSR_JSTRT1, 634
ADC_CSR_JSTRT2, 634
ADC_CSR_JSTRT3, 634
ADC_CSR_OVR1, 634
ADC_CSR_OVR2, 634
ADC_CSR_OVR3, 634
ADC_CSR_STRT1, 634
ADC_CSR_STRT2, 634
ADC_CSR_STRT3, 635
ADC_DR_ADC2DATA, 635
ADC_DR_DATA, 635
ADC_HTR_HT, 635
ADC_JDR1_JDATA, 635
ADC_JDR2_JDATA, 635
ADC_JDR3_JDATA, 635
ADC_JDR4_JDATA, 635
ADC_JOFR1_JOFFSET1, 636
ADC_JOFR2_JOFFSET2, 636
ADC_JOFR3_JOFFSET3, 636
ADC_JOFR4_JOFFSET4, 636
ADC_JSQR_JL, 636
ADC_JSQR_JL_0, 636
ADC_JSQR_JL_1, 636
ADC_JSQR_JSQ1, 636
ADC_JSQR_JSQ1_0, 637
ADC_JSQR_JSQ1_1, 637
ADC_JSQR_JSQ1_2, 637
ADC_JSQR_JSQ1_3, 637
ADC_JSQR_JSQ1_4, 637
ADC_JSQR_JSQ2, 637
ADC_JSQR_JSQ2_0, 637
ADC_JSQR_JSQ2_1, 637
ADC_JSQR_JSQ2_2, 638
ADC_JSQR_JSQ2_3, 638
ADC_JSQR_JSQ2_4, 638
ADC_JSQR_JSQ3, 638
ADC_JSQR_JSQ3_0, 638
ADC_JSQR_JSQ3_1, 638
ADC_JSQR_JSQ3_2, 638
ADC_JSQR_JSQ3_3, 638

ADC_JSQR_JSQ3_4, 639
ADC_JSQR_JSQ4, 639
ADC_JSQR_JSQ4_0, 639
ADC_JSQR_JSQ4_1, 639
ADC_JSQR_JSQ4_2, 639
ADC_JSQR_JSQ4_3, 639
ADC_JSQR_JSQ4_4, 639
ADC_LTR_LT, 639
ADC_SMPR1_SMP10, 640
ADC_SMPR1_SMP10_0, 640
ADC_SMPR1_SMP10_1, 640
ADC_SMPR1_SMP10_2, 640
ADC_SMPR1_SMP11, 640
ADC_SMPR1_SMP11_0, 640
ADC_SMPR1_SMP11_1, 640
ADC_SMPR1_SMP11_2, 640
ADC_SMPR1_SMP12, 641
ADC_SMPR1_SMP12_0, 641
ADC_SMPR1_SMP12_1, 641
ADC_SMPR1_SMP12_2, 641
ADC_SMPR1_SMP13, 641
ADC_SMPR1_SMP13_0, 641
ADC_SMPR1_SMP13_1, 641
ADC_SMPR1_SMP13_2, 641
ADC_SMPR1_SMP14, 642
ADC_SMPR1_SMP14_0, 642
ADC_SMPR1_SMP14_1, 642
ADC_SMPR1_SMP14_2, 642
ADC_SMPR1_SMP15, 642
ADC_SMPR1_SMP15_0, 642
ADC_SMPR1_SMP15_1, 642
ADC_SMPR1_SMP15_2, 642
ADC_SMPR1_SMP16, 643
ADC_SMPR1_SMP16_0, 643
ADC_SMPR1_SMP16_1, 643
ADC_SMPR1_SMP16_2, 643
ADC_SMPR1_SMP17, 643
ADC_SMPR1_SMP17_0, 643
ADC_SMPR1_SMP17_1, 643
ADC_SMPR1_SMP17_2, 643
ADC_SMPR1_SMP18, 644
ADC_SMPR1_SMP18_0, 644
ADC_SMPR1_SMP18_1, 644
ADC_SMPR1_SMP18_2, 644
ADC_SMPR2_SMP0, 644
ADC_SMPR2_SMP0_0, 644
ADC_SMPR2_SMP0_1, 644
ADC_SMPR2_SMP0_2, 644
ADC_SMPR2_SMP1, 645
ADC_SMPR2_SMP1_0, 645
ADC_SMPR2_SMP1_1, 645
ADC_SMPR2_SMP1_2, 645
ADC_SMPR2_SMP2, 645
ADC_SMPR2_SMP2_0, 645
ADC_SMPR2_SMP2_1, 645
ADC_SMPR2_SMP2_2, 645
ADC_SMPR2_SMP3, 646
ADC_SMPR2_SMP3_0, 646
ADC_SMPR2_SMP3_1, 646
ADC_SMPR2_SMP3_2, 646
ADC_SMPR2_SMP4, 646
ADC_SMPR2_SMP4_0, 646
ADC_SMPR2_SMP4_1, 646
ADC_SMPR2_SMP4_2, 646
ADC_SMPR2_SMP5, 647
ADC_SMPR2_SMP5_0, 647
ADC_SMPR2_SMP5_1, 647
ADC_SMPR2_SMP5_2, 647
ADC_SMPR2_SMP6, 647
ADC_SMPR2_SMP6_0, 647
ADC_SMPR2_SMP6_1, 647
ADC_SMPR2_SMP6_2, 647
ADC_SMPR2_SMP7, 648
ADC_SMPR2_SMP7_0, 648
ADC_SMPR2_SMP7_1, 648
ADC_SMPR2_SMP7_2, 648
ADC_SMPR2_SMP8, 648
ADC_SMPR2_SMP8_0, 648
ADC_SMPR2_SMP8_1, 648
ADC_SMPR2_SMP8_2, 648
ADC_SMPR2_SMP9, 649
ADC_SMPR2_SMP9_0, 649
ADC_SMPR2_SMP9_1, 649
ADC_SMPR2_SMP9_2, 649
ADC_SQR1_L, 649
ADC_SQR1_L_0, 649
ADC_SQR1_L_1, 649
ADC_SQR1_L_2, 649
ADC_SQR1_L_3, 650
ADC_SQR1_SQ13, 650
ADC_SQR1_SQ13_0, 650
ADC_SQR1_SQ13_1, 650
ADC_SQR1_SQ13_2, 650
ADC_SQR1_SQ13_3, 650
ADC_SQR1_SQ13_4, 650
ADC_SQR1_SQ14, 650
ADC_SQR1_SQ14_0, 651
ADC_SQR1_SQ14_1, 651
ADC_SQR1_SQ14_2, 651
ADC_SQR1_SQ14_3, 651
ADC_SQR1_SQ14_4, 651
ADC_SQR1_SQ15, 651
ADC_SQR1_SQ15_0, 651
ADC_SQR1_SQ15_1, 651
ADC_SQR1_SQ15_2, 652
ADC_SQR1_SQ15_3, 652
ADC_SQR1_SQ15_4, 652
ADC_SQR1_SQ16, 652
ADC_SQR1_SQ16_0, 652
ADC_SQR1_SQ16_1, 652
ADC_SQR1_SQ16_2, 652
ADC_SQR1_SQ16_3, 652
ADC_SQR1_SQ16_4, 653
ADC_SQR2_SQ10, 653
ADC_SQR2_SQ10_0, 653
ADC_SQR2_SQ10_1, 653

ADC_SQR2_SQ10_2, 653
ADC_SQR2_SQ10_3, 653
ADC_SQR2_SQ10_4, 653
ADC_SQR2_SQ11, 653
ADC_SQR2_SQ11_0, 654
ADC_SQR2_SQ11_1, 654
ADC_SQR2_SQ11_2, 654
ADC_SQR2_SQ11_3, 654
ADC_SQR2_SQ11_4, 654
ADC_SQR2_SQ12, 654
ADC_SQR2_SQ12_0, 654
ADC_SQR2_SQ12_1, 654
ADC_SQR2_SQ12_2, 655
ADC_SQR2_SQ12_3, 655
ADC_SQR2_SQ12_4, 655
ADC_SQR2_SQ7, 655
ADC_SQR2_SQ7_0, 655
ADC_SQR2_SQ7_1, 655
ADC_SQR2_SQ7_2, 655
ADC_SQR2_SQ7_3, 655
ADC_SQR2_SQ7_4, 656
ADC_SQR2_SQ8, 656
ADC_SQR2_SQ8_0, 656
ADC_SQR2_SQ8_1, 656
ADC_SQR2_SQ8_2, 656
ADC_SQR2_SQ8_3, 656
ADC_SQR2_SQ8_4, 656
ADC_SQR2_SQ9, 656
ADC_SQR2_SQ9_0, 657
ADC_SQR2_SQ9_1, 657
ADC_SQR2_SQ9_2, 657
ADC_SQR2_SQ9_3, 657
ADC_SQR2_SQ9_4, 657
ADC_SQR3_SQ1, 657
ADC_SQR3_SQ1_0, 657
ADC_SQR3_SQ1_1, 657
ADC_SQR3_SQ1_2, 658
ADC_SQR3_SQ1_3, 658
ADC_SQR3_SQ1_4, 658
ADC_SQR3_SQ2, 658
ADC_SQR3_SQ2_0, 658
ADC_SQR3_SQ2_1, 658
ADC_SQR3_SQ2_2, 658
ADC_SQR3_SQ2_3, 658
ADC_SQR3_SQ2_4, 659
ADC_SQR3_SQ3, 659
ADC_SQR3_SQ3_0, 659
ADC_SQR3_SQ3_1, 659
ADC_SQR3_SQ3_2, 659
ADC_SQR3_SQ3_3, 659
ADC_SQR3_SQ3_4, 659
ADC_SQR3_SQ4, 659
ADC_SQR3_SQ4_0, 660
ADC_SQR3_SQ4_1, 660
ADC_SQR3_SQ4_2, 660
ADC_SQR3_SQ4_3, 660
ADC_SQR3_SQ4_4, 660
ADC_SQR3_SQ5, 660
ADC_SQR3_SQ5_0, 660
ADC_SQR3_SQ5_1, 660
ADC_SQR3_SQ5_2, 661
ADC_SQR3_SQ5_3, 661
ADC_SQR3_SQ5_4, 661
ADC_SQR3_SQ6, 661
ADC_SQR3_SQ6_0, 661
ADC_SQR3_SQ6_1, 661
ADC_SQR3_SQ6_2, 661
ADC_SQR3_SQ6_3, 661
ADC_SQR3_SQ6_4, 662
ADC_SR_AWD, 662
ADC_SR_EOC, 662
ADC_SR_JEOC, 662
ADC_SR_JSTRT, 662
ADC_SR_OVR, 662
ADC_SR_STRT, 662
CAN_BTR_BRP, 663
CAN_BTR_LBKM, 663
CAN_BTR_SILM, 663
CAN_BTR_SJW, 663
CAN_BTR_TS1, 663
CAN_BTR_TS2, 663
CAN_ESR_BOFF, 663
CAN_ESR_EPVF, 664
CAN_ESR_EWGF, 664
CAN_ESR_LEC, 664
CAN_ESR_LEC_0, 664
CAN_ESR_LEC_1, 664
CAN_ESR_LEC_2, 664
CAN_ESR_REC, 664
CAN_ESR_TEC, 664
CAN_F0R1_FB0, 665
CAN_F0R1_FB1, 665
CAN_F0R1_FB10, 665
CAN_F0R1_FB11, 665
CAN_F0R1_FB12, 665
CAN_F0R1_FB13, 665
CAN_F0R1_FB14, 665
CAN_F0R1_FB15, 665
CAN_F0R1_FB16, 666
CAN_F0R1_FB17, 666
CAN_F0R1_FB18, 666
CAN_F0R1_FB19, 666
CAN_F0R1_FB2, 666
CAN_F0R1_FB20, 666
CAN_F0R1_FB21, 666
CAN_F0R1_FB22, 666
CAN_F0R1_FB23, 667
CAN_F0R1_FB24, 667
CAN_F0R1_FB25, 667
CAN_F0R1_FB26, 667
CAN_F0R1_FB27, 667
CAN_F0R1_FB28, 667
CAN_F0R1_FB29, 667
CAN_F0R1_FB3, 667
CAN_F0R1_FB30, 668
CAN_F0R1_FB31, 668

CAN_F0R1_FB4, 668
CAN_F0R1_FB5, 668
CAN_F0R1_FB6, 668
CAN_F0R1_FB7, 668
CAN_F0R1_FB8, 668
CAN_F0R1_FB9, 668
CAN_F0R2_FB0, 669
CAN_F0R2_FB1, 669
CAN_F0R2_FB10, 669
CAN_F0R2_FB11, 669
CAN_F0R2_FB12, 669
CAN_F0R2_FB13, 669
CAN_F0R2_FB14, 669
CAN_F0R2_FB15, 669
CAN_F0R2_FB16, 670
CAN_F0R2_FB17, 670
CAN_F0R2_FB18, 670
CAN_F0R2_FB19, 670
CAN_F0R2_FB2, 670
CAN_F0R2_FB20, 670
CAN_F0R2_FB21, 670
CAN_F0R2_FB22, 670
CAN_F0R2_FB23, 671
CAN_F0R2_FB24, 671
CAN_F0R2_FB25, 671
CAN_F0R2_FB26, 671
CAN_F0R2_FB27, 671
CAN_F0R2_FB28, 671
CAN_F0R2_FB29, 671
CAN_F0R2_FB3, 671
CAN_F0R2_FB30, 672
CAN_F0R2_FB31, 672
CAN_F0R2_FB4, 672
CAN_F0R2_FB5, 672
CAN_F0R2_FB6, 672
CAN_F0R2_FB7, 672
CAN_F0R2_FB8, 672
CAN_F0R2_FB9, 672
CAN_F10R1_FB0, 673
CAN_F10R1_FB1, 673
CAN_F10R1_FB10, 673
CAN_F10R1_FB11, 673
CAN_F10R1_FB12, 673
CAN_F10R1_FB13, 673
CAN_F10R1_FB14, 673
CAN_F10R1_FB15, 673
CAN_F10R1_FB16, 674
CAN_F10R1_FB17, 674
CAN_F10R1_FB18, 674
CAN_F10R1_FB19, 674
CAN_F10R1_FB2, 674
CAN_F10R1_FB20, 674
CAN_F10R1_FB21, 674
CAN_F10R1_FB22, 674
CAN_F10R1_FB23, 675
CAN_F10R1_FB24, 675
CAN_F10R1_FB25, 675
CAN_F10R1_FB26, 675
CAN_F10R1_FB27, 675
CAN_F10R1_FB28, 675
CAN_F10R1_FB29, 675
CAN_F10R1_FB3, 675
CAN_F10R1_FB30, 676
CAN_F10R1_FB31, 676
CAN_F10R1_FB4, 676
CAN_F10R1_FB5, 676
CAN_F10R1_FB6, 676
CAN_F10R1_FB7, 676
CAN_F10R1_FB8, 676
CAN_F10R1_FB9, 676
CAN_F10R2_FB0, 677
CAN_F10R2_FB1, 677
CAN_F10R2_FB10, 677
CAN_F10R2_FB11, 677
CAN_F10R2_FB12, 677
CAN_F10R2_FB13, 677
CAN_F10R2_FB14, 677
CAN_F10R2_FB15, 677
CAN_F10R2_FB16, 678
CAN_F10R2_FB17, 678
CAN_F10R2_FB18, 678
CAN_F10R2_FB19, 678
CAN_F10R2_FB2, 678
CAN_F10R2_FB20, 678
CAN_F10R2_FB21, 678
CAN_F10R2_FB22, 678
CAN_F10R2_FB23, 679
CAN_F10R2_FB24, 679
CAN_F10R2_FB25, 679
CAN_F10R2_FB26, 679
CAN_F10R2_FB27, 679
CAN_F10R2_FB28, 679
CAN_F10R2_FB29, 679
CAN_F10R2_FB3, 679
CAN_F10R2_FB30, 680
CAN_F10R2_FB31, 680
CAN_F10R2_FB4, 680
CAN_F10R2_FB5, 680
CAN_F10R2_FB6, 680
CAN_F10R2_FB7, 680
CAN_F10R2_FB8, 680
CAN_F10R2_FB9, 680
CAN_F11R1_FB0, 681
CAN_F11R1_FB1, 681
CAN_F11R1_FB10, 681
CAN_F11R1_FB11, 681
CAN_F11R1_FB12, 681
CAN_F11R1_FB13, 681
CAN_F11R1_FB14, 681
CAN_F11R1_FB15, 681
CAN_F11R1_FB16, 682
CAN_F11R1_FB17, 682
CAN_F11R1_FB18, 682
CAN_F11R1_FB19, 682
CAN_F11R1_FB2, 682
CAN_F11R1_FB20, 682

CAN_F11R1_FB21, 682
CAN_F11R1_FB22, 682
CAN_F11R1_FB23, 683
CAN_F11R1_FB24, 683
CAN_F11R1_FB25, 683
CAN_F11R1_FB26, 683
CAN_F11R1_FB27, 683
CAN_F11R1_FB28, 683
CAN_F11R1_FB29, 683
CAN_F11R1_FB3, 683
CAN_F11R1_FB30, 684
CAN_F11R1_FB31, 684
CAN_F11R1_FB4, 684
CAN_F11R1_FB5, 684
CAN_F11R1_FB6, 684
CAN_F11R1_FB7, 684
CAN_F11R1_FB8, 684
CAN_F11R1_FB9, 684
CAN_F11R2_FB0, 685
CAN_F11R2_FB1, 685
CAN_F11R2_FB10, 685
CAN_F11R2_FB11, 685
CAN_F11R2_FB12, 685
CAN_F11R2_FB13, 685
CAN_F11R2_FB14, 685
CAN_F11R2_FB15, 685
CAN_F11R2_FB16, 686
CAN_F11R2_FB17, 686
CAN_F11R2_FB18, 686
CAN_F11R2_FB19, 686
CAN_F11R2_FB2, 686
CAN_F11R2_FB20, 686
CAN_F11R2_FB21, 686
CAN_F11R2_FB22, 686
CAN_F11R2_FB23, 687
CAN_F11R2_FB24, 687
CAN_F11R2_FB25, 687
CAN_F11R2_FB26, 687
CAN_F11R2_FB27, 687
CAN_F11R2_FB28, 687
CAN_F11R2_FB29, 687
CAN_F11R2_FB3, 687
CAN_F11R2_FB30, 688
CAN_F11R2_FB31, 688
CAN_F11R2_FB4, 688
CAN_F11R2_FB5, 688
CAN_F11R2_FB6, 688
CAN_F11R2_FB7, 688
CAN_F11R2_FB8, 688
CAN_F11R2_FB9, 688
CAN_F12R1_FB0, 689
CAN_F12R1_FB1, 689
CAN_F12R1_FB10, 689
CAN_F12R1_FB11, 689
CAN_F12R1_FB12, 689
CAN_F12R1_FB13, 689
CAN_F12R1_FB14, 689
CAN_F12R1_FB15, 689
CAN_F12R1_FB16, 690
CAN_F12R1_FB17, 690
CAN_F12R1_FB18, 690
CAN_F12R1_FB19, 690
CAN_F12R1_FB2, 690
CAN_F12R1_FB20, 690
CAN_F12R1_FB21, 690
CAN_F12R1_FB22, 690
CAN_F12R1_FB23, 691
CAN_F12R1_FB24, 691
CAN_F12R1_FB25, 691
CAN_F12R1_FB26, 691
CAN_F12R1_FB27, 691
CAN_F12R1_FB28, 691
CAN_F12R1_FB29, 691
CAN_F12R1_FB3, 691
CAN_F12R1_FB30, 692
CAN_F12R1_FB31, 692
CAN_F12R1_FB4, 692
CAN_F12R1_FB5, 692
CAN_F12R1_FB6, 692
CAN_F12R1_FB7, 692
CAN_F12R1_FB8, 692
CAN_F12R1_FB9, 692
CAN_F12R2_FB0, 693
CAN_F12R2_FB1, 693
CAN_F12R2_FB10, 693
CAN_F12R2_FB11, 693
CAN_F12R2_FB12, 693
CAN_F12R2_FB13, 693
CAN_F12R2_FB14, 693
CAN_F12R2_FB15, 693
CAN_F12R2_FB16, 694
CAN_F12R2_FB17, 694
CAN_F12R2_FB18, 694
CAN_F12R2_FB19, 694
CAN_F12R2_FB2, 694
CAN_F12R2_FB20, 694
CAN_F12R2_FB21, 694
CAN_F12R2_FB22, 694
CAN_F12R2_FB23, 695
CAN_F12R2_FB24, 695
CAN_F12R2_FB25, 695
CAN_F12R2_FB26, 695
CAN_F12R2_FB27, 695
CAN_F12R2_FB28, 695
CAN_F12R2_FB29, 695
CAN_F12R2_FB3, 695
CAN_F12R2_FB30, 696
CAN_F12R2_FB31, 696
CAN_F12R2_FB4, 696
CAN_F12R2_FB5, 696
CAN_F12R2_FB6, 696
CAN_F12R2_FB7, 696
CAN_F12R2_FB8, 696
CAN_F12R2_FB9, 696
CAN_F13R1_FB0, 697
CAN_F13R1_FB1, 697

CAN_F13R1_FB10, 697
CAN_F13R1_FB11, 697
CAN_F13R1_FB12, 697
CAN_F13R1_FB13, 697
CAN_F13R1_FB14, 697
CAN_F13R1_FB15, 697
CAN_F13R1_FB16, 698
CAN_F13R1_FB17, 698
CAN_F13R1_FB18, 698
CAN_F13R1_FB19, 698
CAN_F13R1_FB2, 698
CAN_F13R1_FB20, 698
CAN_F13R1_FB21, 698
CAN_F13R1_FB22, 698
CAN_F13R1_FB23, 699
CAN_F13R1_FB24, 699
CAN_F13R1_FB25, 699
CAN_F13R1_FB26, 699
CAN_F13R1_FB27, 699
CAN_F13R1_FB28, 699
CAN_F13R1_FB29, 699
CAN_F13R1_FB3, 699
CAN_F13R1_FB30, 700
CAN_F13R1_FB31, 700
CAN_F13R1_FB4, 700
CAN_F13R1_FB5, 700
CAN_F13R1_FB6, 700
CAN_F13R1_FB7, 700
CAN_F13R1_FB8, 700
CAN_F13R1_FB9, 700
CAN_F13R2_FB0, 701
CAN_F13R2_FB1, 701
CAN_F13R2_FB10, 701
CAN_F13R2_FB11, 701
CAN_F13R2_FB12, 701
CAN_F13R2_FB13, 701
CAN_F13R2_FB14, 701
CAN_F13R2_FB15, 701
CAN_F13R2_FB16, 702
CAN_F13R2_FB17, 702
CAN_F13R2_FB18, 702
CAN_F13R2_FB19, 702
CAN_F13R2_FB2, 702
CAN_F13R2_FB20, 702
CAN_F13R2_FB21, 702
CAN_F13R2_FB22, 702
CAN_F13R2_FB23, 703
CAN_F13R2_FB24, 703
CAN_F13R2_FB25, 703
CAN_F13R2_FB26, 703
CAN_F13R2_FB27, 703
CAN_F13R2_FB28, 703
CAN_F13R2_FB29, 703
CAN_F13R2_FB3, 703
CAN_F13R2_FB30, 704
CAN_F13R2_FB31, 704
CAN_F13R2_FB4, 704
CAN_F13R2_FB5, 704
CAN_F13R2_FB6, 704
CAN_F13R2_FB7, 704
CAN_F13R2_FB8, 704
CAN_F13R2_FB9, 704
CAN_F1R1_FB0, 705
CAN_F1R1_FB1, 705
CAN_F1R1_FB10, 705
CAN_F1R1_FB11, 705
CAN_F1R1_FB12, 705
CAN_F1R1_FB13, 705
CAN_F1R1_FB14, 705
CAN_F1R1_FB15, 705
CAN_F1R1_FB16, 706
CAN_F1R1_FB17, 706
CAN_F1R1_FB18, 706
CAN_F1R1_FB19, 706
CAN_F1R1_FB2, 706
CAN_F1R1_FB20, 706
CAN_F1R1_FB21, 706
CAN_F1R1_FB22, 706
CAN_F1R1_FB23, 707
CAN_F1R1_FB24, 707
CAN_F1R1_FB25, 707
CAN_F1R1_FB26, 707
CAN_F1R1_FB27, 707
CAN_F1R1_FB28, 707
CAN_F1R1_FB29, 707
CAN_F1R1_FB3, 707
CAN_F1R1_FB30, 708
CAN_F1R1_FB31, 708
CAN_F1R1_FB4, 708
CAN_F1R1_FB5, 708
CAN_F1R1_FB6, 708
CAN_F1R1_FB7, 708
CAN_F1R1_FB8, 708
CAN_F1R1_FB9, 708
CAN_F1R2_FB0, 709
CAN_F1R2_FB1, 709
CAN_F1R2_FB10, 709
CAN_F1R2_FB11, 709
CAN_F1R2_FB12, 709
CAN_F1R2_FB13, 709
CAN_F1R2_FB14, 709
CAN_F1R2_FB15, 709
CAN_F1R2_FB16, 710
CAN_F1R2_FB17, 710
CAN_F1R2_FB18, 710
CAN_F1R2_FB19, 710
CAN_F1R2_FB2, 710
CAN_F1R2_FB20, 710
CAN_F1R2_FB21, 710
CAN_F1R2_FB22, 710
CAN_F1R2_FB23, 711
CAN_F1R2_FB24, 711
CAN_F1R2_FB25, 711
CAN_F1R2_FB26, 711
CAN_F1R2_FB27, 711
CAN_F1R2_FB28, 711

CAN_F1R2_FB29, 711
CAN_F1R2_FB3, 711
CAN_F1R2_FB30, 712
CAN_F1R2_FB31, 712
CAN_F1R2_FB4, 712
CAN_F1R2_FB5, 712
CAN_F1R2_FB6, 712
CAN_F1R2_FB7, 712
CAN_F1R2_FB8, 712
CAN_F1R2_FB9, 712
CAN_F2R1_FB0, 713
CAN_F2R1_FB1, 713
CAN_F2R1_FB10, 713
CAN_F2R1_FB11, 713
CAN_F2R1_FB12, 713
CAN_F2R1_FB13, 713
CAN_F2R1_FB14, 713
CAN_F2R1_FB15, 713
CAN_F2R1_FB16, 714
CAN_F2R1_FB17, 714
CAN_F2R1_FB18, 714
CAN_F2R1_FB19, 714
CAN_F2R1_FB2, 714
CAN_F2R1_FB20, 714
CAN_F2R1_FB21, 714
CAN_F2R1_FB22, 714
CAN_F2R1_FB23, 715
CAN_F2R1_FB24, 715
CAN_F2R1_FB25, 715
CAN_F2R1_FB26, 715
CAN_F2R1_FB27, 715
CAN_F2R1_FB28, 715
CAN_F2R1_FB29, 715
CAN_F2R1_FB3, 715
CAN_F2R1_FB30, 716
CAN_F2R1_FB31, 716
CAN_F2R1_FB4, 716
CAN_F2R1_FB5, 716
CAN_F2R1_FB6, 716
CAN_F2R1_FB7, 716
CAN_F2R1_FB8, 716
CAN_F2R1_FB9, 716
CAN_F2R2_FB0, 717
CAN_F2R2_FB1, 717
CAN_F2R2_FB10, 717
CAN_F2R2_FB11, 717
CAN_F2R2_FB12, 717
CAN_F2R2_FB13, 717
CAN_F2R2_FB14, 717
CAN_F2R2_FB15, 717
CAN_F2R2_FB16, 718
CAN_F2R2_FB17, 718
CAN_F2R2_FB18, 718
CAN_F2R2_FB19, 718
CAN_F2R2_FB2, 718
CAN_F2R2_FB20, 718
CAN_F2R2_FB21, 718
CAN_F2R2_FB22, 718
CAN_F2R2_FB23, 719
CAN_F2R2_FB24, 719
CAN_F2R2_FB25, 719
CAN_F2R2_FB26, 719
CAN_F2R2_FB27, 719
CAN_F2R2_FB28, 719
CAN_F2R2_FB29, 719
CAN_F2R2_FB3, 719
CAN_F2R2_FB30, 720
CAN_F2R2_FB31, 720
CAN_F2R2_FB4, 720
CAN_F2R2_FB5, 720
CAN_F2R2_FB6, 720
CAN_F2R2_FB7, 720
CAN_F2R2_FB8, 720
CAN_F2R2_FB9, 720
CAN_F3R1_FB0, 721
CAN_F3R1_FB1, 721
CAN_F3R1_FB10, 721
CAN_F3R1_FB11, 721
CAN_F3R1_FB12, 721
CAN_F3R1_FB13, 721
CAN_F3R1_FB14, 721
CAN_F3R1_FB15, 721
CAN_F3R1_FB16, 722
CAN_F3R1_FB17, 722
CAN_F3R1_FB18, 722
CAN_F3R1_FB19, 722
CAN_F3R1_FB2, 722
CAN_F3R1_FB20, 722
CAN_F3R1_FB21, 722
CAN_F3R1_FB22, 722
CAN_F3R1_FB23, 723
CAN_F3R1_FB24, 723
CAN_F3R1_FB25, 723
CAN_F3R1_FB26, 723
CAN_F3R1_FB27, 723
CAN_F3R1_FB28, 723
CAN_F3R1_FB29, 723
CAN_F3R1_FB3, 723
CAN_F3R1_FB30, 724
CAN_F3R1_FB31, 724
CAN_F3R1_FB4, 724
CAN_F3R1_FB5, 724
CAN_F3R1_FB6, 724
CAN_F3R1_FB7, 724
CAN_F3R1_FB8, 724
CAN_F3R1_FB9, 724
CAN_F3R2_FB0, 725
CAN_F3R2_FB1, 725
CAN_F3R2_FB10, 725
CAN_F3R2_FB11, 725
CAN_F3R2_FB12, 725
CAN_F3R2_FB13, 725
CAN_F3R2_FB14, 725
CAN_F3R2_FB15, 725
CAN_F3R2_FB16, 726
CAN_F3R2_FB17, 726

CAN_F3R2_FB18, 726
CAN_F3R2_FB19, 726
CAN_F3R2_FB2, 726
CAN_F3R2_FB20, 726
CAN_F3R2_FB21, 726
CAN_F3R2_FB22, 726
CAN_F3R2_FB23, 727
CAN_F3R2_FB24, 727
CAN_F3R2_FB25, 727
CAN_F3R2_FB26, 727
CAN_F3R2_FB27, 727
CAN_F3R2_FB28, 727
CAN_F3R2_FB29, 727
CAN_F3R2_FB3, 727
CAN_F3R2_FB30, 728
CAN_F3R2_FB31, 728
CAN_F3R2_FB4, 728
CAN_F3R2_FB5, 728
CAN_F3R2_FB6, 728
CAN_F3R2_FB7, 728
CAN_F3R2_FB8, 728
CAN_F3R2_FB9, 728
CAN_F4R1_FB0, 729
CAN_F4R1_FB1, 729
CAN_F4R1_FB10, 729
CAN_F4R1_FB11, 729
CAN_F4R1_FB12, 729
CAN_F4R1_FB13, 729
CAN_F4R1_FB14, 729
CAN_F4R1_FB15, 729
CAN_F4R1_FB16, 730
CAN_F4R1_FB17, 730
CAN_F4R1_FB18, 730
CAN_F4R1_FB19, 730
CAN_F4R1_FB2, 730
CAN_F4R1_FB20, 730
CAN_F4R1_FB21, 730
CAN_F4R1_FB22, 730
CAN_F4R1_FB23, 731
CAN_F4R1_FB24, 731
CAN_F4R1_FB25, 731
CAN_F4R1_FB26, 731
CAN_F4R1_FB27, 731
CAN_F4R1_FB28, 731
CAN_F4R1_FB29, 731
CAN_F4R1_FB3, 731
CAN_F4R1_FB30, 732
CAN_F4R1_FB31, 732
CAN_F4R1_FB4, 732
CAN_F4R1_FB5, 732
CAN_F4R1_FB6, 732
CAN_F4R1_FB7, 732
CAN_F4R1_FB8, 732
CAN_F4R1_FB9, 732
CAN_F4R2_FB0, 733
CAN_F4R2_FB1, 733
CAN_F4R2_FB10, 733
CAN_F4R2_FB11, 733
CAN_F4R2_FB12, 733
CAN_F4R2_FB13, 733
CAN_F4R2_FB14, 733
CAN_F4R2_FB15, 733
CAN_F4R2_FB16, 734
CAN_F4R2_FB17, 734
CAN_F4R2_FB18, 734
CAN_F4R2_FB19, 734
CAN_F4R2_FB2, 734
CAN_F4R2_FB20, 734
CAN_F4R2_FB21, 734
CAN_F4R2_FB22, 734
CAN_F4R2_FB23, 735
CAN_F4R2_FB24, 735
CAN_F4R2_FB25, 735
CAN_F4R2_FB26, 735
CAN_F4R2_FB27, 735
CAN_F4R2_FB28, 735
CAN_F4R2_FB29, 735
CAN_F4R2_FB3, 735
CAN_F4R2_FB30, 736
CAN_F4R2_FB31, 736
CAN_F4R2_FB4, 736
CAN_F4R2_FB5, 736
CAN_F4R2_FB6, 736
CAN_F4R2_FB7, 736
CAN_F4R2_FB8, 736
CAN_F4R2_FB9, 736
CAN_F5R1_FB0, 737
CAN_F5R1_FB1, 737
CAN_F5R1_FB10, 737
CAN_F5R1_FB11, 737
CAN_F5R1_FB12, 737
CAN_F5R1_FB13, 737
CAN_F5R1_FB14, 737
CAN_F5R1_FB15, 737
CAN_F5R1_FB16, 738
CAN_F5R1_FB17, 738
CAN_F5R1_FB18, 738
CAN_F5R1_FB19, 738
CAN_F5R1_FB2, 738
CAN_F5R1_FB20, 738
CAN_F5R1_FB21, 738
CAN_F5R1_FB22, 738
CAN_F5R1_FB23, 739
CAN_F5R1_FB24, 739
CAN_F5R1_FB25, 739
CAN_F5R1_FB26, 739
CAN_F5R1_FB27, 739
CAN_F5R1_FB28, 739
CAN_F5R1_FB29, 739
CAN_F5R1_FB3, 739
CAN_F5R1_FB30, 740
CAN_F5R1_FB31, 740
CAN_F5R1_FB4, 740
CAN_F5R1_FB5, 740
CAN_F5R1_FB6, 740
CAN_F5R1_FB7, 740

CAN_F5R1_FB8, 740
CAN_F5R1_FB9, 740
CAN_F5R2_FB0, 741
CAN_F5R2_FB1, 741
CAN_F5R2_FB10, 741
CAN_F5R2_FB11, 741
CAN_F5R2_FB12, 741
CAN_F5R2_FB13, 741
CAN_F5R2_FB14, 741
CAN_F5R2_FB15, 741
CAN_F5R2_FB16, 742
CAN_F5R2_FB17, 742
CAN_F5R2_FB18, 742
CAN_F5R2_FB19, 742
CAN_F5R2_FB2, 742
CAN_F5R2_FB20, 742
CAN_F5R2_FB21, 742
CAN_F5R2_FB22, 742
CAN_F5R2_FB23, 743
CAN_F5R2_FB24, 743
CAN_F5R2_FB25, 743
CAN_F5R2_FB26, 743
CAN_F5R2_FB27, 743
CAN_F5R2_FB28, 743
CAN_F5R2_FB29, 743
CAN_F5R2_FB3, 743
CAN_F5R2_FB30, 744
CAN_F5R2_FB31, 744
CAN_F5R2_FB4, 744
CAN_F5R2_FB5, 744
CAN_F5R2_FB6, 744
CAN_F5R2_FB7, 744
CAN_F5R2_FB8, 744
CAN_F5R2_FB9, 744
CAN_F6R1_FB0, 745
CAN_F6R1_FB1, 745
CAN_F6R1_FB10, 745
CAN_F6R1_FB11, 745
CAN_F6R1_FB12, 745
CAN_F6R1_FB13, 745
CAN_F6R1_FB14, 745
CAN_F6R1_FB15, 745
CAN_F6R1_FB16, 746
CAN_F6R1_FB17, 746
CAN_F6R1_FB18, 746
CAN_F6R1_FB19, 746
CAN_F6R1_FB2, 746
CAN_F6R1_FB20, 746
CAN_F6R1_FB21, 746
CAN_F6R1_FB22, 746
CAN_F6R1_FB23, 747
CAN_F6R1_FB24, 747
CAN_F6R1_FB25, 747
CAN_F6R1_FB26, 747
CAN_F6R1_FB27, 747
CAN_F6R1_FB28, 747
CAN_F6R1_FB29, 747
CAN_F6R1_FB3, 747
CAN_F6R1_FB30, 748
CAN_F6R1_FB31, 748
CAN_F6R1_FB4, 748
CAN_F6R1_FB5, 748
CAN_F6R1_FB6, 748
CAN_F6R1_FB7, 748
CAN_F6R1_FB8, 748
CAN_F6R1_FB9, 748
CAN_F6R2_FB0, 749
CAN_F6R2_FB1, 749
CAN_F6R2_FB10, 749
CAN_F6R2_FB11, 749
CAN_F6R2_FB12, 749
CAN_F6R2_FB13, 749
CAN_F6R2_FB14, 749
CAN_F6R2_FB15, 749
CAN_F6R2_FB16, 750
CAN_F6R2_FB17, 750
CAN_F6R2_FB18, 750
CAN_F6R2_FB19, 750
CAN_F6R2_FB2, 750
CAN_F6R2_FB20, 750
CAN_F6R2_FB21, 750
CAN_F6R2_FB22, 750
CAN_F6R2_FB23, 751
CAN_F6R2_FB24, 751
CAN_F6R2_FB25, 751
CAN_F6R2_FB26, 751
CAN_F6R2_FB27, 751
CAN_F6R2_FB28, 751
CAN_F6R2_FB29, 751
CAN_F6R2_FB3, 751
CAN_F6R2_FB30, 752
CAN_F6R2_FB31, 752
CAN_F6R2_FB4, 752
CAN_F6R2_FB5, 752
CAN_F6R2_FB6, 752
CAN_F6R2_FB7, 752
CAN_F6R2_FB8, 752
CAN_F6R2_FB9, 752
CAN_F7R1_FB0, 753
CAN_F7R1_FB1, 753
CAN_F7R1_FB10, 753
CAN_F7R1_FB11, 753
CAN_F7R1_FB12, 753
CAN_F7R1_FB13, 753
CAN_F7R1_FB14, 753
CAN_F7R1_FB15, 753
CAN_F7R1_FB16, 754
CAN_F7R1_FB17, 754
CAN_F7R1_FB18, 754
CAN_F7R1_FB19, 754
CAN_F7R1_FB2, 754
CAN_F7R1_FB20, 754
CAN_F7R1_FB21, 754
CAN_F7R1_FB22, 754
CAN_F7R1_FB23, 755
CAN_F7R1_FB24, 755

CAN_F7R1_FB25, 755
CAN_F7R1_FB26, 755
CAN_F7R1_FB27, 755
CAN_F7R1_FB28, 755
CAN_F7R1_FB29, 755
CAN_F7R1_FB3, 755
CAN_F7R1_FB30, 756
CAN_F7R1_FB31, 756
CAN_F7R1_FB4, 756
CAN_F7R1_FB5, 756
CAN_F7R1_FB6, 756
CAN_F7R1_FB7, 756
CAN_F7R1_FB8, 756
CAN_F7R1_FB9, 756
CAN_F7R2_FB0, 757
CAN_F7R2_FB1, 757
CAN_F7R2_FB10, 757
CAN_F7R2_FB11, 757
CAN_F7R2_FB12, 757
CAN_F7R2_FB13, 757
CAN_F7R2_FB14, 757
CAN_F7R2_FB15, 757
CAN_F7R2_FB16, 758
CAN_F7R2_FB17, 758
CAN_F7R2_FB18, 758
CAN_F7R2_FB19, 758
CAN_F7R2_FB2, 758
CAN_F7R2_FB20, 758
CAN_F7R2_FB21, 758
CAN_F7R2_FB22, 758
CAN_F7R2_FB23, 759
CAN_F7R2_FB24, 759
CAN_F7R2_FB25, 759
CAN_F7R2_FB26, 759
CAN_F7R2_FB27, 759
CAN_F7R2_FB28, 759
CAN_F7R2_FB29, 759
CAN_F7R2_FB3, 759
CAN_F7R2_FB30, 760
CAN_F7R2_FB31, 760
CAN_F7R2_FB4, 760
CAN_F7R2_FB5, 760
CAN_F7R2_FB6, 760
CAN_F7R2_FB7, 760
CAN_F7R2_FB8, 760
CAN_F7R2_FB9, 760
CAN_F8R1_FB0, 761
CAN_F8R1_FB1, 761
CAN_F8R1_FB10, 761
CAN_F8R1_FB11, 761
CAN_F8R1_FB12, 761
CAN_F8R1_FB13, 761
CAN_F8R1_FB14, 761
CAN_F8R1_FB15, 761
CAN_F8R1_FB16, 762
CAN_F8R1_FB17, 762
CAN_F8R1_FB18, 762
CAN_F8R1_FB19, 762
CAN_F8R1_FB2, 762
CAN_F8R1_FB20, 762
CAN_F8R1_FB21, 762
CAN_F8R1_FB22, 762
CAN_F8R1_FB23, 763
CAN_F8R1_FB24, 763
CAN_F8R1_FB25, 763
CAN_F8R1_FB26, 763
CAN_F8R1_FB27, 763
CAN_F8R1_FB28, 763
CAN_F8R1_FB29, 763
CAN_F8R1_FB3, 763
CAN_F8R1_FB30, 764
CAN_F8R1_FB31, 764
CAN_F8R1_FB4, 764
CAN_F8R1_FB5, 764
CAN_F8R1_FB6, 764
CAN_F8R1_FB7, 764
CAN_F8R1_FB8, 764
CAN_F8R1_FB9, 764
CAN_F8R2_FB0, 765
CAN_F8R2_FB1, 765
CAN_F8R2_FB10, 765
CAN_F8R2_FB11, 765
CAN_F8R2_FB12, 765
CAN_F8R2_FB13, 765
CAN_F8R2_FB14, 765
CAN_F8R2_FB15, 765
CAN_F8R2_FB16, 766
CAN_F8R2_FB17, 766
CAN_F8R2_FB18, 766
CAN_F8R2_FB19, 766
CAN_F8R2_FB2, 766
CAN_F8R2_FB20, 766
CAN_F8R2_FB21, 766
CAN_F8R2_FB22, 766
CAN_F8R2_FB23, 767
CAN_F8R2_FB24, 767
CAN_F8R2_FB25, 767
CAN_F8R2_FB26, 767
CAN_F8R2_FB27, 767
CAN_F8R2_FB28, 767
CAN_F8R2_FB29, 767
CAN_F8R2_FB3, 767
CAN_F8R2_FB30, 768
CAN_F8R2_FB31, 768
CAN_F8R2_FB4, 768
CAN_F8R2_FB5, 768
CAN_F8R2_FB6, 768
CAN_F8R2_FB7, 768
CAN_F8R2_FB8, 768
CAN_F8R2_FB9, 768
CAN_F9R1_FB0, 769
CAN_F9R1_FB1, 769
CAN_F9R1_FB10, 769
CAN_F9R1_FB11, 769
CAN_F9R1_FB12, 769
CAN_F9R1_FB13, 769

CAN_F9R1_FB14, 769
CAN_F9R1_FB15, 769
CAN_F9R1_FB16, 770
CAN_F9R1_FB17, 770
CAN_F9R1_FB18, 770
CAN_F9R1_FB19, 770
CAN_F9R1_FB2, 770
CAN_F9R1_FB20, 770
CAN_F9R1_FB21, 770
CAN_F9R1_FB22, 770
CAN_F9R1_FB23, 771
CAN_F9R1_FB24, 771
CAN_F9R1_FB25, 771
CAN_F9R1_FB26, 771
CAN_F9R1_FB27, 771
CAN_F9R1_FB28, 771
CAN_F9R1_FB29, 771
CAN_F9R1_FB3, 771
CAN_F9R1_FB30, 772
CAN_F9R1_FB31, 772
CAN_F9R1_FB4, 772
CAN_F9R1_FB5, 772
CAN_F9R1_FB6, 772
CAN_F9R1_FB7, 772
CAN_F9R1_FB8, 772
CAN_F9R1_FB9, 772
CAN_F9R2_FB0, 773
CAN_F9R2_FB1, 773
CAN_F9R2_FB10, 773
CAN_F9R2_FB11, 773
CAN_F9R2_FB12, 773
CAN_F9R2_FB13, 773
CAN_F9R2_FB14, 773
CAN_F9R2_FB15, 773
CAN_F9R2_FB16, 774
CAN_F9R2_FB17, 774
CAN_F9R2_FB18, 774
CAN_F9R2_FB19, 774
CAN_F9R2_FB2, 774
CAN_F9R2_FB20, 774
CAN_F9R2_FB21, 774
CAN_F9R2_FB22, 774
CAN_F9R2_FB23, 775
CAN_F9R2_FB24, 775
CAN_F9R2_FB25, 775
CAN_F9R2_FB26, 775
CAN_F9R2_FB27, 775
CAN_F9R2_FB28, 775
CAN_F9R2_FB29, 775
CAN_F9R2_FB3, 775
CAN_F9R2_FB30, 776
CAN_F9R2_FB31, 776
CAN_F9R2_FB4, 776
CAN_F9R2_FB5, 776
CAN_F9R2_FB6, 776
CAN_F9R2_FB7, 776
CAN_F9R2_FB8, 776
CAN_F9R2_FB9, 776
CAN_FA1R_FACT, 777
CAN_FA1R_FACT0, 777
CAN_FA1R_FACT1, 777
CAN_FA1R_FACT10, 777
CAN_FA1R_FACT11, 777
CAN_FA1R_FACT12, 777
CAN_FA1R_FACT13, 777
CAN_FA1R_FACT2, 777
CAN_FA1R_FACT3, 778
CAN_FA1R_FACT4, 778
CAN_FA1R_FACT5, 778
CAN_FA1R_FACT6, 778
CAN_FA1R_FACT7, 778
CAN_FA1R_FACT8, 778
CAN_FA1R_FACT9, 778
CAN_FFA1R_FFA, 778
CAN_FFA1R_FFA0, 779
CAN_FFA1R_FFA1, 779
CAN_FFA1R_FFA10, 779
CAN_FFA1R_FFA11, 779
CAN_FFA1R_FFA12, 779
CAN_FFA1R_FFA13, 779
CAN_FFA1R_FFA2, 779
CAN_FFA1R_FFA3, 779
CAN_FFA1R_FFA4, 780
CAN_FFA1R_FFA5, 780
CAN_FFA1R_FFA6, 780
CAN_FFA1R_FFA7, 780
CAN_FFA1R_FFA8, 780
CAN_FFA1R_FFA9, 780
CAN_FM1R_FBM, 780
CAN_FM1R_FBM0, 780
CAN_FM1R_FBM1, 781
CAN_FM1R_FBM10, 781
CAN_FM1R_FBM11, 781
CAN_FM1R_FBM12, 781
CAN_FM1R_FBM13, 781
CAN_FM1R_FBM2, 781
CAN_FM1R_FBM3, 781
CAN_FM1R_FBM4, 781
CAN_FM1R_FBM5, 782
CAN_FM1R_FBM6, 782
CAN_FM1R_FBM7, 782
CAN_FM1R_FBM8, 782
CAN_FM1R_FBM9, 782
CAN_FMR_INIT, 782
CAN_FS1R_FSC, 782
CAN_FS1R_FSC0, 782
CAN_FS1R_FSC1, 783
CAN_FS1R_FSC10, 783
CAN_FS1R_FSC11, 783
CAN_FS1R_FSC12, 783
CAN_FS1R_FSC13, 783
CAN_FS1R_FSC2, 783
CAN_FS1R_FSC3, 783
CAN_FS1R_FSC4, 783
CAN_FS1R_FSC5, 784
CAN_FS1R_FSC6, 784

CAN_FS1R_FSC7, 784
CAN_FS1R_FSC8, 784
CAN_FS1R_FSC9, 784
CAN_IER_BOFIE, 784
CAN_IER_EPVIE, 784
CAN_IER_ERRIE, 784
CAN_IER_EWGIE, 785
CAN_IER_FFIE0, 785
CAN_IER_FFIE1, 785
CAN_IER_FMPIE0, 785
CAN_IER_FMPIE1, 785
CAN_IER_FOVIE0, 785
CAN_IER_FOVIE1, 785
CAN_IER_LECIE, 785
CAN_IER_SLKIE, 786
CAN_IER_TMEIE, 786
CAN_IER_WKUIE, 786
CAN_MCR_ABOM, 786
CAN_MCR_AWUM, 786
CAN_MCR_INRQ, 786
CAN_MCR_NART, 786
CAN_MCR_RESET, 786
CAN_MCR_RFLM, 787
CAN_MCR_SLEEP, 787
CAN_MCR_TTCM, 787
CAN_MCR_TXFP, 787
CAN_MSR_ERRI, 787
CAN_MSR_INAK, 787
CAN_MSR_RX, 787
CAN_MSR_RXM, 787
CAN_MSR_SAMP, 788
CAN_MSR_SLAK, 788
CAN_MSR_SLAKI, 788
CAN_MSR_TXM, 788
CAN_MSR_WKUI, 788
CAN_RDH0R_DATA4, 788
CAN_RDH0R_DATA5, 788
CAN_RDH0R_DATA6, 788
CAN_RDH0R_DATA7, 789
CAN_RDH1R_DATA4, 789
CAN_RDH1R_DATA5, 789
CAN_RDH1R_DATA6, 789
CAN_RDH1R_DATA7, 789
CAN_RDL0R_DATA0, 789
CAN_RDL0R_DATA1, 789
CAN_RDL0R_DATA2, 789
CAN_RDL0R_DATA3, 790
CAN_RDL1R_DATA0, 790
CAN_RDL1R_DATA1, 790
CAN_RDL1R_DATA2, 790
CAN_RDL1R_DATA3, 790
CAN_RDT0R_DLC, 790
CAN_RDT0R_FMI, 790
CAN_RDT0R_TIME, 790
CAN_RDT1R_DLC, 791
CAN_RDT1R_FMI, 791
CAN_RDT1R_TIME, 791
CAN_RF0R_FMP0, 791
CAN_RF0R_FOVR0, 791
CAN_RF0R_FULL0, 791
CAN_RF0R_RFOM0, 791
CAN_RF1R_FMP1, 791
CAN_RF1R_FOVR1, 792
CAN_RF1R_FULL1, 792
CAN_RF1R_RFOM1, 792
CAN_RI0R_EXID, 792
CAN_RI0R_IDE, 792
CAN_RI0R_RTR, 792
CAN_RI0R_STID, 792
CAN_RI1R_EXID, 792
CAN_RI1R_IDE, 793
CAN_RI1R_RTR, 793
CAN_RI1R_STID, 793
CAN_TDH0R_DATA4, 793
CAN_TDH0R_DATA5, 793
CAN_TDH0R_DATA6, 793
CAN_TDH0R_DATA7, 793
CAN_TDH1R_DATA4, 793
CAN_TDH1R_DATA5, 794
CAN_TDH1R_DATA6, 794
CAN_TDH1R_DATA7, 794
CAN_TDH2R_DATA4, 794
CAN_TDH2R_DATA5, 794
CAN_TDH2R_DATA6, 794
CAN_TDH2R_DATA7, 794
CAN_TDL0R_DATA0, 794
CAN_TDL0R_DATA1, 795
CAN_TDL0R_DATA2, 795
CAN_TDL0R_DATA3, 795
CAN_TDL1R_DATA0, 795
CAN_TDL1R_DATA1, 795
CAN_TDL1R_DATA2, 795
CAN_TDL1R_DATA3, 795
CAN_TDL2R_DATA0, 795
CAN_TDL2R_DATA1, 796
CAN_TDL2R_DATA2, 796
CAN_TDL2R_DATA3, 796
CAN_TDT0R_DLC, 796
CAN_TDT0R_TGT, 796
CAN_TDT0R_TIME, 796
CAN_TDT1R_DLC, 796
CAN_TDT1R_TGT, 796
CAN_TDT1R_TIME, 797
CAN_TDT2R_DLC, 797
CAN_TDT2R_TGT, 797
CAN_TDT2R_TIME, 797
CAN_TI0R_EXID, 797
CAN_TI0R_IDE, 797
CAN_TI0R_RTR, 797
CAN_TI0R_STID, 797
CAN_TI0R_TXRQ, 798
CAN_TI1R_EXID, 798
CAN_TI1R_IDE, 798
CAN_TI1R_RTR, 798
CAN_TI1R_STID, 798
CAN_TI1R_TXRQ, 798

CAN_TI2R_EXID, 798
CAN_TI2R_IDE, 798
CAN_TI2R_RTR, 799
CAN_TI2R_STID, 799
CAN_TI2R_TXRQ, 799
CAN_TSR_ABRQ0, 799
CAN_TSR_ABRQ1, 799
CAN_TSR_ABRQ2, 799
CAN_TSR_ALST0, 799
CAN_TSR_ALST1, 799
CAN_TSR_ALST2, 800
CAN_TSR_CODE, 800
CAN_TSR_LOW, 800
CAN_TSR_LOW0, 800
CAN_TSR_LOW1, 800
CAN_TSR_LOW2, 800
CAN_TSR_RQCP0, 800
CAN_TSR_RQCP1, 800
CAN_TSR_RQCP2, 801
CAN_TSR_TERR0, 801
CAN_TSR_TERR1, 801
CAN_TSR_TERR2, 801
CAN_TSR_TME, 801
CAN_TSR_TME0, 801
CAN_TSR_TME1, 801
CAN_TSR_TME2, 801
CAN_TSR_TXOK0, 802
CAN_TSR_TXOK1, 802
CAN_TSR_TXOK2, 802
CRC_CR_RESET, 802
CRC_DR_DR, 802
CRC_IDR_IDR, 802
DAC_CR_BOFF1, 802
DAC_CR_BOFF2, 802
DAC_CR_DMAEN1, 803
DAC_CR_DMAEN2, 803
DAC_CR_DMAUDRIE1, 803
DAC_CR_DMAUDRIE2, 803
DAC_CR_EN1, 803
DAC_CR_EN2, 803
DAC_CR_MAMP1, 803
DAC_CR_MAMP1_0, 803
DAC_CR_MAMP1_1, 804
DAC_CR_MAMP1_2, 804
DAC_CR_MAMP1_3, 804
DAC_CR_MAMP2, 804
DAC_CR_MAMP2_0, 804
DAC_CR_MAMP2_1, 804
DAC_CR_MAMP2_2, 804
DAC_CR_MAMP2_3, 804
DAC_CR_TEN1, 805
DAC_CR_TEN2, 805
DAC_CR_TSEL1, 805
DAC_CR_TSEL1_0, 805
DAC_CR_TSEL1_1, 805
DAC_CR_TSEL1_2, 805
DAC_CR_TSEL2, 805
DAC_CR_TSEL2_0, 805
DAC_CR_TSEL2_1, 806
DAC_CR_TSEL2_2, 806
DAC_CR_WAVE1, 806
DAC_CR_WAVE1_0, 806
DAC_CR_WAVE1_1, 806
DAC_CR_WAVE2, 806
DAC_CR_WAVE2_0, 806
DAC_CR_WAVE2_1, 806
DAC_DHR12L1_DACC1DHR, 807
DAC_DHR12L2_DACC2DHR, 807
DAC_DHR12LD_DACC1DHR, 807
DAC_DHR12LD_DACC2DHR, 807
DAC_DHR12R1_DACC1DHR, 807
DAC_DHR12R2_DACC2DHR, 807
DAC_DHR12RD_DACC1DHR, 807
DAC_DHR12RD_DACC2DHR, 807
DAC_DHR8R1_DACC1DHR, 808
DAC_DHR8R2_DACC2DHR, 808
DAC_DHR8RD_DACC1DHR, 808
DAC_DHR8RD_DACC2DHR, 808
DAC_DOR1_DACC1DOR, 808
DAC_DOR2_DACC2DOR, 808
DAC_SR_DMAUDR1, 808
DAC_SR_DMAUDR2, 808
DAC_SWTRIGR_SWTRIG1, 809
DAC_SWTRIGR_SWTRIG2, 809
DBGMCU_CR_TRACE_MODE_0, 809
DBGMCU_CR_TRACE_MODE_1, 809
DFSDM_CHAWSCDR_AWFORD, 809
DFSDM_CHAWSCDR_AWFORD_0, 809
DFSDM_CHAWSCDR_AWFORD_1, 809
DFSDM_CHAWSCDR_AWFOSR, 809
DFSDM_CHAWSCDR_BKSCD, 810
DFSDM_CHAWSCDR_SCDT, 810
DFSDM_CHCFGR1_CHEN, 810
DFSDM_CHCFGR1_CHINSEL, 810
DFSDM_CHCFGR1_CKABEN, 810
DFSDM_CHCFGR1_CKOUTDIV, 810
DFSDM_CHCFGR1_CKOUTSRC, 810
DFSDM_CHCFGR1_DATMPX, 810
DFSDM_CHCFGR1_DATMPX_0, 811
DFSDM_CHCFGR1_DATMPX_1, 811
DFSDM_CHCFGR1_DATPACK, 811
DFSDM_CHCFGR1_DATPACK_0, 811
DFSDM_CHCFGR1_DATPACK_1, 811
DFSDM_CHCFGR1_DFSDMEN, 811
DFSDM_CHCFGR1_SCDEN, 811
DFSDM_CHCFGR1_SITP, 811
DFSDM_CHCFGR1_SITP_0, 812
DFSDM_CHCFGR1_SITP_1, 812
DFSDM_CHCFGR1_SPICKSEL, 812
DFSDM_CHCFGR1_SPICKSEL_0, 812
DFSDM_CHCFGR1_SPICKSEL_1, 812
DFSDM_CHCFGR2_DTRBS, 812
DFSDM_CHCFGR2_OFFSET, 812
DFSDM_CHDATINR_INDAT0, 812
DFSDM_CHDATINR_INDAT1, 813
DFSDM_CHWDATR_WDATA, 813

DFSDM_FLTAWCFR_CLRAWHTF, 813
DFSDM_FLTAWCFR_CLRAWLTF, 813
DFSDM_FLTAWHTR_AWHT, 813
DFSDM_FLTAWHTR_BKAWH, 813
DFSDM_FLTAWLTR_AWLT, 813
DFSDM_FLTAWLTR_BKAWL, 813
DFSDM_FLTAWSR_AWHTF, 814
DFSDM_FLTAWSR_AWLTF, 814
DFSDM_FLTCNVTIMR_CNV_CNT, 814
DFSDM_FLTCR1_AWFSEL, 814
DFSDM_FLTCR1_DFEN, 814
DFSDM_FLTCR1_FAST, 814
DFSDM_FLTCR1_JDMAEN, 814
DFSDM_FLTCR1_JEXTEN, 814
DFSDM_FLTCR1_JEXTEN_0, 815
DFSDM_FLTCR1_JEXTEN_1, 815
DFSDM_FLTCR1_JEXTSEL, 815
DFSDM_FLTCR1_JEXTSEL_0, 815
DFSDM_FLTCR1_JEXTSEL_1, 815
DFSDM_FLTCR1_JEXTSEL_2, 815
DFSDM_FLTCR1_JSCAN, 815
DFSDM_FLTCR1_JSWSTART, 815
DFSDM_FLTCR1_JSYNC, 816
DFSDM_FLTCR1_RCH, 816
DFSDM_FLTCR1_RCONT, 816
DFSDM_FLTCR1_RDMAEN, 816
DFSDM_FLTCR1_RSWSTART, 816
DFSDM_FLTCR1_RSYNC, 816
DFSDM_FLTCR2_AWDCH, 816
DFSDM_FLTCR2_AWDIE, 817
DFSDM_FLTCR2_CKABIE, 817
DFSDM_FLTCR2_EXCH, 817
DFSDM_FLTCR2_JEOCIE, 817
DFSDM_FLTCR2_JOVRIE, 817
DFSDM_FLTCR2_REOCIE, 817
DFSDM_FLTCR2_ROVRIE, 817
DFSDM_FLTCR2_SCDIE, 817
DFSDM_FLTEXMAX_EXMAX, 818
DFSDM_FLTEXMAX_EXMAXCH, 818
DFSDM_FLTEXMIN_EXMIN, 818
DFSDM_FLTEXMIN_EXMINCH, 818
DFSDM_FLTFCR_FORD, 818
DFSDM_FLTFCR_FORD_0, 818
DFSDM_FLTFCR_FORD_1, 818
DFSDM_FLTFCR_FORD_2, 818
DFSDM_FLTFCR_FOSR, 819
DFSDM_FLTFCR_IOSR, 819
DFSDM_FLTICR_CLRCKABF, 819
DFSDM_FLTICR_CLRJOVRF, 819
DFSDM_FLTICR_CLRROVRF, 819
DFSDM_FLTICR_CLRSCSDF, 819
DFSDM_FLTISR_AWDF, 819
DFSDM_FLTISR_CKABF, 819
DFSDM_FLTISR_JCIP, 820
DFSDM_FLTISR_JEOCF, 820
DFSDM_FLTISR_JOVRF, 820
DFSDM_FLTISR_RCIP, 820
DFSDM_FLTISR_ReOcf, 820
DFSDM_FLTISR_ROVRF, 820
DFSDM_FLTISR_SCDF, 820
DFSDM_FLTJCHGR_JCHG, 820
DFSDM_FLTJDATAR_JDATA, 821
DFSDM_FLTJDATAR_JDATACH, 821
DFSDM_FLTRDATAR_RDATA, 821
DFSDM_FLTRDATAR_RDATAL, 821
DFSDM_FLTRDATAR_RPEND, 821
DMA2D_AMTCR_DT, 821
DMA2D_AMTCR_EN, 821
DMA2D_BGCMAR_MA, 821
DMA2D_BGCCRL_BLUE, 822
DMA2D_BGCCRL_GREEN, 822
DMA2D_BGCCRL_RED, 822
DMA2D_BGMAR_MA, 822
DMA2D_BGOR_LO, 822
DMA2D_BGPCCR_ALPHA, 822
DMA2D_BGPCCR_AM, 822
DMA2D_BGPCCR_AM_0, 822
DMA2D_BGPCCR_AM_1, 823
DMA2D_BGPCCR_CCM, 823
DMA2D_BGPCCR_CM, 823
DMA2D_BGPCCR_CM_0, 823
DMA2D_BGPCCR_CM_1, 823
DMA2D_BGPCCR_CM_2, 823
DMA2D_BGPCCR_CS, 823
DMA2D_BGPCCR_START, 823
DMA2D_CR_ABORT, 824
DMA2D_CR_CAEIE, 824
DMA2D_CR_CEIE, 824
DMA2D_CR_CTCIE, 824
DMA2D_CR_MODE, 824
DMA2D_CR_START, 824
DMA2D_CR_SUSP, 824
DMA2D_CR_TCIE, 824
DMA2D_CR_TEIE, 825
DMA2D_CR_TWIE, 825
DMA2D_FGCMAR_MA, 825
DMA2D_FGCCRL_BLUE, 825
DMA2D_FGCCRL_GREEN, 825
DMA2D_FGCCRL_RED, 825
DMA2D_FGMAR_MA, 825
DMA2D_FGOR_LO, 825
DMA2D_FGPCCR_ALPHA, 826
DMA2D_FGPCCR_AM, 826
DMA2D_FGPCCR_AM_0, 826
DMA2D_FGPCCR_AM_1, 826
DMA2D_FGPCCR_CCM, 826
DMA2D_FGPCCR_CM, 826
DMA2D_FGPCCR_CM_0, 826
DMA2D_FGPCCR_CM_1, 826
DMA2D_FGPCCR_CM_2, 827
DMA2D_FGPCCR_CM_3, 827
DMA2D_FGPCCR_CS, 827
DMA2D_FGPCCR_START, 827
DMA2D_IFCR_CAEIF, 827
DMA2D_IFCR_CCEIF, 827
DMA2D_IFCR_CCTCIF, 828

DMA2D_IFCR_CTCIF, 828
DMA2D_IFCR_CTEIF, 828
DMA2D_IFCR_CTWIF, 828
DMA2D_IFSR_CCAEIF, 828
DMA2D_IFSR_CCEIF, 828
DMA2D_IFSR_CCTCIF, 828
DMA2D_IFSR_CTCIF, 829
DMA2D_IFSR_CTEIF, 829
DMA2D_IFSR_CTWIF, 829
DMA2D_ISR_CAEIF, 829
DMA2D_ISR_CEIF, 829
DMA2D_ISR_CTCIF, 829
DMA2D_ISR_TCIF, 829
DMA2D_ISR_TEIF, 830
DMA2D_ISR_TWIF, 830
DMA2D_LWR_LW, 830
DMA2D_NLR_NL, 830
DMA2D_NLR_PL, 830
DMA2D_OCOLR_ALPHA_1, 830
DMA2D_OCOLR_ALPHA_3, 830
DMA2D_OCOLR_ALPHA_4, 830
DMA2D_OCOLR_BLUE_1, 831
DMA2D_OCOLR_BLUE_2, 831
DMA2D_OCOLR_BLUE_3, 831
DMA2D_OCOLR_BLUE_4, 831
DMA2D_OCOLR_GREEN_1, 831
DMA2D_OCOLR_GREEN_2, 831
DMA2D_OCOLR_GREEN_3, 831
DMA2D_OCOLR_GREEN_4, 832
DMA2D_OCOLR_RED_1, 832
DMA2D_OCOLR_RED_2, 832
DMA2D_OCOLR_RED_3, 832
DMA2D_OCOLR_RED_4, 832
DMA2D_OMAR_MA, 832
DMA2D_OOR_LO, 832
DMA2D_OPFCCR_CM, 833
DMA2D_OPFCCR_CM_0, 833
DMA2D_OPFCCR_CM_1, 833
DMA2D_OPFCCR_CM_2, 833
ETH_MACCR_BL, 833
EXTI_EMR_MR0, 833
EXTI_EMR_MR1, 833
EXTI_EMR_MR10, 834
EXTI_EMR_MR11, 834
EXTI_EMR_MR12, 834
EXTI_EMR_MR13, 834
EXTI_EMR_MR14, 834
EXTI_EMR_MR15, 834
EXTI_EMR_MR16, 834
EXTI_EMR_MR17, 834
EXTI_EMR_MR18, 835
EXTI_EMR_MR19, 835
EXTI_EMR_MR2, 835
EXTI_EMR_MR23, 835
EXTI_EMR_MR3, 835
EXTI_EMR_MR4, 835
EXTI_EMR_MR5, 835
EXTI_EMR_MR6, 835
EXTI_EMR_MR7, 836
EXTI_EMR_MR8, 836
EXTI_EMR_MR9, 836
EXTI_FTSR_TR0, 836
EXTI_FTSR_TR1, 836
EXTI_FTSR_TR10, 836
EXTI_FTSR_TR11, 836
EXTI_FTSR_TR12, 836
EXTI_FTSR_TR13, 837
EXTI_FTSR_TR14, 837
EXTI_FTSR_TR15, 837
EXTI_FTSR_TR16, 837
EXTI_FTSR_TR17, 837
EXTI_FTSR_TR18, 837
EXTI_FTSR_TR19, 837
EXTI_FTSR_TR2, 837
EXTI_FTSR_TR23, 838
EXTI_FTSR_TR3, 838
EXTI_FTSR_TR4, 838
EXTI_FTSR_TR5, 838
EXTI_FTSR_TR6, 838
EXTI_FTSR_TR7, 838
EXTI_FTSR_TR8, 838
EXTI_FTSR_TR9, 838
EXTI_IMR_MR0, 839
EXTI_IMR_MR1, 839
EXTI_IMR_MR10, 839
EXTI_IMR_MR11, 839
EXTI_IMR_MR12, 839
EXTI_IMR_MR13, 839
EXTI_IMR_MR14, 839
EXTI_IMR_MR15, 839
EXTI_IMR_MR16, 840
EXTI_IMR_MR17, 840
EXTI_IMR_MR18, 840
EXTI_IMR_MR19, 840
EXTI_IMR_MR2, 840
EXTI_IMR_MR23, 840
EXTI_IMR_MR3, 840
EXTI_IMR_MR4, 840
EXTI_IMR_MR5, 841
EXTI_IMR_MR6, 841
EXTI_IMR_MR7, 841
EXTI_IMR_MR8, 841
EXTI_IMR_MR9, 841
EXTI_PR_PR0, 841
EXTI_PR_PR1, 841
EXTI_PR_PR10, 841
EXTI_PR_PR11, 842
EXTI_PR_PR12, 842
EXTI_PR_PR13, 842
EXTI_PR_PR14, 842
EXTI_PR_PR15, 842
EXTI_PR_PR16, 842
EXTI_PR_PR17, 842
EXTI_PR_PR18, 842
EXTI_PR_PR19, 843
EXTI_PR_PR2, 843

EXTI_PR_PR23, 843
EXTI_PR_PR3, 843
EXTI_PR_PR4, 843
EXTI_PR_PR5, 843
EXTI_PR_PR6, 843
EXTI_PR_PR7, 843
EXTI_PR_PR8, 844
EXTI_PR_PR9, 844
EXTI_RTSR_TR0, 844
EXTI_RTSR_TR1, 844
EXTI_RTSR_TR10, 844
EXTI_RTSR_TR11, 844
EXTI_RTSR_TR12, 844
EXTI_RTSR_TR13, 844
EXTI_RTSR_TR14, 845
EXTI_RTSR_TR15, 845
EXTI_RTSR_TR16, 845
EXTI_RTSR_TR17, 845
EXTI_RTSR_TR18, 845
EXTI_RTSR_TR19, 845
EXTI_RTSR_TR2, 845
EXTI_RTSR_TR23, 845
EXTI_RTSR_TR3, 846
EXTI_RTSR_TR4, 846
EXTI_RTSR_TR5, 846
EXTI_RTSR_TR6, 846
EXTI_RTSR_TR7, 846
EXTI_RTSR_TR8, 846
EXTI_RTSR_TR9, 846
EXTI_SWIER_SWIER0, 846
EXTI_SWIER_SWIER1, 847
EXTI_SWIER_SWIER10, 847
EXTI_SWIER_SWIER11, 847
EXTI_SWIER_SWIER12, 847
EXTI_SWIER_SWIER13, 847
EXTI_SWIER_SWIER14, 847
EXTI_SWIER_SWIER15, 847
EXTI_SWIER_SWIER16, 847
EXTI_SWIER_SWIER17, 848
EXTI_SWIER_SWIER18, 848
EXTI_SWIER_SWIER19, 848
EXTI_SWIER_SWIER2, 848
EXTI_SWIER_SWIER23, 848
EXTI_SWIER_SWIER3, 848
EXTI_SWIER_SWIER4, 848
EXTI_SWIER_SWIER5, 848
EXTI_SWIER_SWIER6, 849
EXTI_SWIER_SWIER7, 849
EXTI_SWIER_SWIER8, 849
EXTI_SWIER_SWIER9, 849
I2C_CCR_CCR, 849
I2C_CCR_DUTY, 849
I2C_CCR_FS, 849
I2C_CR1_ACK, 850
I2C_CR1_ALERT, 850
I2C_CR1_ENARP, 850
I2C_CR1_ENGC, 850
I2C_CR1_ENPEC, 850
I2C_CR1_NOSTRETCH, 850
I2C_CR1_PE, 850
I2C_CR1_PEC, 851
I2C_CR1_POS, 851
I2C_CR1_SMBTYPE, 851
I2C_CR1_SMBUS, 851
I2C_CR1_START, 851
I2C_CR1_STOP, 851
I2C_CR1_SWRST, 851
I2C_CR2_DMAEN, 852
I2C_CR2_FREQ, 852
I2C_CR2_FREQ_0, 852
I2C_CR2_FREQ_1, 852
I2C_CR2_FREQ_2, 852
I2C_CR2_FREQ_3, 852
I2C_CR2_FREQ_4, 852
I2C_CR2_FREQ_5, 853
I2C_CR2_ITBUFEN, 853
I2C_CR2_ITERREN, 853
I2C_CR2_IYTEVTEN, 853
I2C_CR2_LAST, 853
I2C_DR_DR, 853
I2C_FLTR_ANOFF, 853
I2C_FLTR_DNF, 854
I2C_OAR1_ADD0, 854
I2C_OAR1_ADD1, 854
I2C_OAR1_ADD1_7, 854
I2C_OAR1_ADD2, 854
I2C_OAR1_ADD3, 854
I2C_OAR1_ADD4, 854
I2C_OAR1_ADD5, 854
I2C_OAR1_ADD6, 855
I2C_OAR1_ADD7, 855
I2C_OAR1_ADD8, 855
I2C_OAR1_ADD8_9, 855
I2C_OAR1_ADD9, 855
I2C_OAR1_ADDMODE, 855
I2C_OAR2_ADD2, 855
I2C_OAR2_ENDUAL, 855
I2C_SR1_ADD10, 856
I2C_SR1_ADDR, 856
I2C_SR1_AF, 856
I2C_SR1_ARLO, 856
I2C_SR1_BERR, 856
I2C_SR1_BTF, 856
I2C_SR1_OVR, 856
I2C_SR1_PECERR, 857
I2C_SR1_RXNE, 857
I2C_SR1_SB, 857
I2C_SR1_SMBALERT, 857
I2C_SR1_STOPF, 857
I2C_SR1_TIMEOUT, 857
I2C_SR1_TXE, 857
I2C_SR2_BUSY, 858
I2C_SR2_DUALF, 858
I2C_SR2_GENCALL, 858
I2C_SR2_MSL, 858
I2C_SR2_PEC, 858

I2C_SR2_SMBDEFAULT, 858
I2C_SR2_SMBHOST, 858
I2C_SR2_TRA, 859
I2C_TRISE_TRISE, 859
IWDG_KR_KEY, 859
IWDG_PR_PR, 859
IWDG_PR_PR_0, 859
IWDG_PR_PR_1, 859
IWDG_PR_PR_2, 859
IWDG_RLR_RL, 860
IWDG_SR_PVU, 860
IWDG_SR_RVU, 860
LTDC_AWCR_AAH, 860
LTDC_AWCR_AAW, 860
LTDC_BCCR_BCBLUE, 860
LTDC_BCCR_BCGREEN, 860
LTDC_BCCR_BCRED, 861
LTDC_BPCR_AHBP, 861
LTDC_BPCR_AVBP, 861
LTDC_CDSR_HDES, 861
LTDC_CDSR_HSYNCS, 861
LTDC_CDSR_VDES, 861
LTDC_CDSR_VSYNCS, 861
LTDC_CPSR_CXPOS, 861
LTDC_CPSR_CYPOS, 862
LTDC_GCR_DBW, 862
LTDC_GCR_DEN, 862
LTDC_GCR_DEPOL, 862
LTDC_GCR_DGW, 862
LTDC_GCR_DRW, 862
LTDC_GCR_HSPOL, 862
LTDC_GCR_LTDCEN, 862
LTDC_GCR_PCPOL, 863
LTDC_GCR_VSPOL, 863
LTDC_ICR_CFUIF, 863
LTDC_ICR_CLIF, 863
LTDC_ICR_CRRIF, 863
LTDC_ICR_CTERRIF, 863
LTDC_IER_FUIE, 863
LTDC_IER_LIE, 863
LTDC_IER_RRIE, 864
LTDC_IER_TERRIE, 864
LTDC_ISR_FUIF, 864
LTDC_ISR_LIF, 864
LTDC_ISR_RRIF, 864
LTDC_ISR_TERRIF, 864
LTDC_LIPCR_LIPOS, 864
LTDC_LxBFCR_BF1, 864
LTDC_LxBFCR_BF2, 865
LTDC_LxCACR_CONSTA, 865
LTDC_LxCFBAR_CFBADD, 865
LTDC_LxCFBLNR_CFBLNBR, 865
LTDC_LxCFBLR_CFBLL, 865
LTDC_LxCFBLR_CFBP, 865
LTDC_LxCKCR_CKBLUE, 865
LTDC_LxCKCR_CKGREEN, 865
LTDC_LxCKCR_CKRED, 866
LTDC_LxCLUTWR_BLUE, 866
LTDC_LxCLUTWR_CLUTADD, 866
LTDC_LxCLUTWR_GREEN, 866
LTDC_LxCLUTWR_RED, 866
LTDC_LxCR_CLUTEN, 866
LTDC_LxCR_COLKEN, 866
LTDC_LxCR_LEN, 866
LTDC_LxDCCR_DCALPHA, 867
LTDC_LxDCCR_DCBLUE, 867
LTDC_LxDCCR_DCGREEN, 867
LTDC_LxDCCR_DCRED, 867
LTDC_LxFPCR_PF, 867
LTDC_LxWHPCR_WHSPPOS, 867
LTDC_LxWHPCR_WHSTPOS, 867
LTDC_LxWVPCR_WVSPPPOS, 867
LTDC_LxWVPCR_WVSTPOS, 868
LTDC_SRCR_IMR, 868
LTDC_SRCR_VBR, 868
LTDC_SSCR_HSW, 868
LTDC_SSCR_VSH, 868
LTDC_TWCR_TOTALH, 868
LTDC_TWCR_TOTALW, 868
PWR_CR_ADCDC1, 868
PWR_CR_CSBF, 869
PWR_CR_CWUF, 869
PWR_CR_DBP, 869
PWR_CR_FISSR, 869
PWR_CR_FMSSR, 869
PWR_CR_FPPDS, 869
PWR_CR_LPDS, 869
PWR_CR_LPLVDS, 870
PWR_CR_LPUDS, 870
PWR_CR_MRLVDS, 870
PWR_CR_MRUDS, 870
PWR_CR_ODEN, 870
PWR_CR_ODSWEN, 870
PWR_CR_PDDS, 870
PWR_CR_PLS, 871
PWR_CR_PLS_0, 871
PWR_CR_PLS_1, 871
PWR_CR_PLS_2, 871
PWR_CR_PLSLEV0, 871
PWR_CR_PLSLEV1, 871
PWR_CR_PLSLEV2, 871
PWR_CR_PLSLEV3, 872
PWR_CR_PLSLEV4, 872
PWR_CR_PLSLEV5, 872
PWR_CR_PLSLEV6, 872
PWR_CR_PLSLEV7, 872
PWR_CR_PVDE, 872
PWR_CR_UDEN, 872
PWR_CR_UDEN_0, 873
PWR_CR_UDEN_1, 873
PWR_CR_VOS, 873
PWR_CR_VOS_0, 873
PWR_CR_VOS_1, 873
PWR_CSR_BRE, 873
PWR_CSR_BRR, 873
PWR_CSR_EWUP, 874

PWR_CSR_ODRDY, 874
PWR_CSR_ODSWRDY, 874
PWR_CSR_PVDO, 874
PWR_CSR_SBF, 874
PWR_CSR_UDSWRDY, 874
PWR_CSR_VOSRDY, 874
PWR_CSR_WUF, 875
PWR_CSR_WUPP, 875
RCC_CFGR_HPRE, 875
RCC_CFGR_HPRE_0, 875
RCC_CFGR_HPRE_1, 875
RCC_CFGR_HPRE_2, 875
RCC_CFGR_HPRE_3, 875
RCC_CFGR_HPRE_DIV1, 876
RCC_CFGR_HPRE_DIV128, 876
RCC_CFGR_HPRE_DIV16, 876
RCC_CFGR_HPRE_DIV2, 876
RCC_CFGR_HPRE_DIV256, 876
RCC_CFGR_HPRE_DIV4, 876
RCC_CFGR_HPRE_DIV512, 876
RCC_CFGR_HPRE_DIV64, 876
RCC_CFGR_HPRE_DIV8, 877
RCC_CFGR_PPREG1, 877
RCC_CFGR_PPREG1_0, 877
RCC_CFGR_PPREG1_1, 877
RCC_CFGR_PPREG1_2, 877
RCC_CFGR_PPREG1_DIV1, 877
RCC_CFGR_PPREG1_DIV16, 877
RCC_CFGR_PPREG1_DIV2, 877
RCC_CFGR_PPREG1_DIV4, 878
RCC_CFGR_PPREG1_DIV8, 878
RCC_CFGR_PPREG2, 878
RCC_CFGR_PPREG2_0, 878
RCC_CFGR_PPREG2_1, 878
RCC_CFGR_PPREG2_2, 878
RCC_CFGR_PPREG2_DIV1, 878
RCC_CFGR_PPREG2_DIV16, 878
RCC_CFGR_PPREG2_DIV2, 879
RCC_CFGR_PPREG2_DIV4, 879
RCC_CFGR_PPREG2_DIV8, 879
RCC_CFGR_RTCPPREG_4, 879
RCC_CFGR_SW, 879
RCC_CFGR_SW_0, 879
RCC_CFGR_SW_1, 879
RCC_CFGR_SW_HSE, 879
RCC_CFGR_SW_HSI, 880
RCC_CFGR_SW_PLL, 880
RCC_CFGR_SWS, 880
RCC_CFGR_SWS_0, 880
RCC_CFGR_SWS_1, 880
RCC_CFGR_SWS_HSE, 880
RCC_CFGR_SWS_HSI, 880
RCC_CFGR_SWS_PLL, 880
RCC_CR_HSICAL_0, 881
RCC_CR_HSICAL_1, 881
RCC_CR_HSICAL_2, 881
RCC_CR_HSICAL_3, 881
RCC_CR_HSICAL_4, 881
RCC_CR_HSICAL_5, 881
RCC_CR_HSICAL_6, 881
RCC_CR_HSICAL_7, 882
RCC_CR_HSITRIM_0, 882
RCC_CR_HSITRIM_1, 882
RCC_CR_HSITRIM_2, 882
RCC_CR_HSITRIM_3, 882
RCC_CR_HSITRIM_4, 882
SAI_GCR_SYNCIN, 882
SAI_GCR_SYNCIN_0, 882
SAI_GCR_SYNCIN_1, 883
SAI_GCR_SYNCOUT, 883
SAI_GCR_SYNCOUT_0, 883
SAI_GCR_SYNCOUT_1, 883
SAI_xCLRFR_CAFSDET, 883
SAI_xCLRFR_CCNRDY, 883
SAI_xCLRFR_CFREQ, 883
SAI_xCLRFR_CLFSDET, 884
SAI_xCLRFR_CMUTEDDET, 884
SAI_xCLRFR_COVRUDR, 884
SAI_xCLRFR_CWCKCFG, 884
SAI_xCR1_CKSTR, 884
SAI_xCR1_DMAEN, 884
SAI_xCR1_DS, 884
SAI_xCR1_DS_0, 885
SAI_xCR1_DS_1, 885
SAI_xCR1_DS_2, 885
SAI_xCR1_LSBFIRST, 885
SAI_xCR1_MCKDIV, 885
SAI_xCR1_MCKDIV_0, 885
SAI_xCR1_MCKDIV_1, 885
SAI_xCR1_MCKDIV_2, 886
SAI_xCR1_MCKDIV_3, 886
SAI_xCR1_MODE, 886
SAI_xCR1_MODE_0, 886
SAI_xCR1_MODE_1, 886
SAI_xCR1_MONO, 886
SAI_xCR1_NODIV, 886
SAI_xCR1_OUTDRIV, 887
SAI_xCR1_PRTCFG, 887
SAI_xCR1_PRTCFG_0, 887
SAI_xCR1_PRTCFG_1, 887
SAI_xCR1_SAIEN, 887
SAI_xCR1_SYNCEN, 887
SAI_xCR1_SYNCEN_0, 887
SAI_xCR1_SYNCEN_1, 888
SAI_xCR2_COMP, 888
SAI_xCR2_COMP_0, 888
SAI_xCR2_COMP_1, 888
SAI_xCR2_CPL, 888
SAI_xCR2_FFLUSH, 888
SAI_xCR2_FTH, 888
SAI_xCR2_FTH_0, 889
SAI_xCR2_FTH_1, 889
SAI_xCR2_MUTE, 889
SAI_xCR2_MUTECNT, 889
SAI_xCR2_MUTECNT_0, 889
SAI_xCR2_MUTECNT_1, 889

SAI_xCR2_MUTECNT_2, 889
 SAI_xCR2_MUTECNT_3, 890
 SAI_xCR2_MUTECNT_4, 890
 SAI_xCR2_MUTECNT_5, 890
 SAI_xCR2_MUTEVAL, 890
 SAI_xCR2_TRIS, 890
 SAI_xFRCR_FRL, 890
 SAI_xFRCR_FRL_0, 890
 SAI_xFRCR_FRL_1, 891
 SAI_xFRCR_FRL_2, 891
 SAI_xFRCR_FRL_3, 891
 SAI_xFRCR_FRL_4, 891
 SAI_xFRCR_FRL_5, 891
 SAI_xFRCR_FRL_6, 891
 SAI_xFRCR_FRL_7, 891
 SAI_xFRCR_FSALL, 891
 SAI_xFRCR_FSALL_0, 892
 SAI_xFRCR_FSALL_1, 892
 SAI_xFRCR_FSALL_2, 892
 SAI_xFRCR_FSALL_3, 892
 SAI_xFRCR_FSALL_4, 892
 SAI_xFRCR_FSALL_5, 892
 SAI_xFRCR_FSALL_6, 892
 SAI_xFRCR_FSDEF, 893
 SAI_xFRCR_FSOFF, 893
 SAI_xFRCR_FSPOL, 893
 SAI_xIMR_AFSDETIE, 893
 SAI_xIMR_CNRDYIE, 893
 SAI_xIMR_FREQIE, 893
 SAI_xIMR_LFSDETIE, 893
 SAI_xIMR_MUTEDETIE, 894
 SAI_xIMR_OVRUDRIE, 894
 SAI_xIMR_WCKCFGIE, 894
 SAI_xSLOTR_FBOFF, 894
 SAI_xSLOTR_FBOFF_0, 894
 SAI_xSLOTR_FBOFF_1, 894
 SAI_xSLOTR_FBOFF_2, 894
 SAI_xSLOTR_FBOFF_3, 895
 SAI_xSLOTR_FBOFF_4, 895
 SAI_xSLOTR_NBSLOT, 895
 SAI_xSLOTR_NBSLOT_0, 895
 SAI_xSLOTR_NBSLOT_1, 895
 SAI_xSLOTR_NBSLOT_2, 895
 SAI_xSLOTR_NBSLOT_3, 895
 SAI_xSLOTR_SLOTEN, 896
 SAI_xSLOTR_SLOTsz, 896
 SAI_xSLOTR_SLOTsz_0, 896
 SAI_xSLOTR_SLOTsz_1, 896
 SAI_xSR_AFSDET, 896
 SAI_xSR_CNRDY, 896
 SAI_xSR_FLVL, 896
 SAI_xSR_FLVL_0, 897
 SAI_xSR_FLVL_1, 897
 SAI_xSR_FLVL_2, 897
 SAI_xSR_FREQ, 897
 SAI_xSR_LFSDET, 897
 SAI_xSR_MUTEDET, 897
 SAI_xSR_OVRUDR, 897
 SAI_xSR_WCKCFG, 898
 SDIO_ARG_CMDARG, 898
 SDIO_CLKCR_BYPASS, 898
 SDIO_CLKCR_CLKDIV, 898
 SDIO_CLKCR_CLKEN, 898
 SDIO_CLKCR_HWFC_EN, 898
 SDIO_CLKCR_NEGEDGE, 898
 SDIO_CLKCR_PWRSAV, 899
 SDIO_CLKCR_WIDBUS, 899
 SDIO_CLKCR_WIDBUS_0, 899
 SDIO_CLKCR_WIDBUS_1, 899
 SDIO_CMD_CEATACMD, 899
 SDIO_CMD_CMDINDEX, 899
 SDIO_CMD_CPSMEN, 899
 SDIO_CMD_ENCMDCOMPL, 900
 SDIO_CMD_NIEN, 900
 SDIO_CMD_SDIOSUSPEND, 900
 SDIO_CMD_WAITINT, 900
 SDIO_CMD_WAITPEND, 900
 SDIO_CMD_WAITRESP, 900
 SDIO_CMD_WAITRESP_0, 900
 SDIO_CMD_WAITRESP_1, 901
 SDIO_DCOUNT_DATACOUNT, 901
 SDIO_DCTRL_DBLOCKSIZE, 901
 SDIO_DCTRL_DBLOCKSIZE_0, 901
 SDIO_DCTRL_DBLOCKSIZE_1, 901
 SDIO_DCTRL_DBLOCKSIZE_2, 901
 SDIO_DCTRL_DBLOCKSIZE_3, 901
 SDIO_DCTRL_DMAEN, 901
 SDIO_DCTRL_DTDIR, 902
 SDIO_DCTRL_DTEN, 902
 SDIO_DCTRL_DTMODE, 902
 SDIO_DCTRL_RWMOD, 902
 SDIO_DCTRL_RWSTART, 902
 SDIO_DCTRL_RWSTOP, 902
 SDIO_DCTRL_SDIOEN, 902
 SDIO_DLLEN_DATALENGTH, 903
 SDIO_DTIMER_DATETIME, 903
 SDIO_FIFO_FIFODATA, 903
 SDIO_FIFOCNT_FIFOCOUNT, 903
 SDIO_ICR_CCRCFAILC, 903
 SDIO_ICR_CEATAENDC, 903
 SDIO_ICR_CMDRENDC, 903
 SDIO_ICR_CMDSENTC, 904
 SDIO_ICR_CTIMEUTC, 904
 SDIO_ICR_DATAENDC, 904
 SDIO_ICR_DBCKENDC, 904
 SDIO_ICR_DCRCFAILC, 904
 SDIO_ICR_DTIMEUTC, 904
 SDIO_ICR_RXOVERRC, 904
 SDIO_ICR_SDIOITC, 905
 SDIO_ICR_STBITERRC, 905
 SDIO_ICR_TXUNDERRC, 905
 SDIO_MASK_CCRCFAILIE, 905
 SDIO_MASK_CEATAENDIE, 905
 SDIO_MASK_CMDACTIE, 905
 SDIO_MASK_CMDRENDIE, 905
 SDIO_MASK_CMDSENTIE, 906

SDIO_MASK_CTIMEOUTIE, 906
SDIO_MASK_DATAENDIE, 906
SDIO_MASK_DBCKENDIE, 906
SDIO_MASK_DCRCFAILIE, 906
SDIO_MASK_DTIMEOUTIE, 906
SDIO_MASK_RXACTIE, 906
SDIO_MASK_RXDAVLIE, 907
SDIO_MASK_RXFIFOEIE, 907
SDIO_MASK_RXFIFOFIE, 907
SDIO_MASK_RXOVERRIE, 907
SDIO_MASK_SDIOITIE, 907
SDIO_MASK_STBITERRIE, 907
SDIO_MASK_TXACTIE, 908
SDIO_MASK_TXDAVLIE, 908
SDIO_MASK_TXFIFOEIE, 908
SDIO_MASK_TXFIFOFIE, 908
SDIO_MASK_TXFIFOHEIE, 908
SDIO_MASK_TXUNDERRIE, 908
SDIO_POWER_PWRCTRL, 908
SDIO_POWER_PWRCTRL_0, 909
SDIO_POWER_PWRCTRL_1, 909
SDIO_RESP0_CARDSTATUS0, 909
SDIO_RESP1_CARDSTATUS1, 909
SDIO_RESP2_CARDSTATUS2, 909
SDIO_RESP3_CARDSTATUS3, 909
SDIO_RESP4_CARDSTATUS4, 909
SDIO_RESPCMD_RESPCMD, 909
SDIO_STA_CCRCFAIL, 910
SDIO_STA_CEATAEND, 910
SDIO_STA_CMDACT, 910
SDIO_STA_CMDREND, 910
SDIO_STA_CMDSENT, 910
SDIO_STA_CTIMEOUT, 910
SDIO_STA_DATAEND, 910
SDIO_STA_DBCKEND, 911
SDIO_STA_DCRCFAIL, 911
SDIO_STA_DTIMEOUT, 911
SDIO_STA_RXACT, 911
SDIO_STA_RXDAVL, 911
SDIO_STA_RXFIFOE, 911
SDIO_STA_RXFIFOF, 911
SDIO_STA_RXFIFOHF, 912
SDIO_STA_RXOVERR, 912
SDIO_STA_SDIOIT, 912
SDIO_STA_STBITERR, 912
SDIO_STA_TXACT, 912
SDIO_STA_TXDAVL, 912
SDIO_STA_TXFIFOE, 912
SDIO_STA_TXFIFOF, 913
SDIO_STA_TXFIFOHE, 913
SDIO_STA_TXUNDERR, 913
SPI_CR1_BIDIMODE, 913
SPI_CR1_BIDIOE, 913
SPI_CR1_BR, 913
SPI_CR1_BR_0, 913
SPI_CR1_BR_1, 914
SPI_CR1_BR_2, 914
SPI_CR1_CPHA, 914
SPI_CR1_CPOL, 914
SPI_CR1_CRCEN, 914
SPI_CR1_CRCNEXT, 914
SPI_CR1_DFF, 914
SPI_CR1_LSBFIRST, 915
SPI_CR1_MSTR, 915
SPI_CR1_RXONLY, 915
SPI_CR1_SPE, 915
SPI_CR1_SSI, 915
SPI_CR1_SSM, 915
SPI_CR2_ERRIE, 915
SPI_CR2_RXDMAEN, 916
SPI_CR2_RXNEIE, 916
SPI_CR2_SSOE, 916
SPI_CR2_RXDMAEN, 916
SPI_CR2_TXEIE, 916
SPI_CRCPR_CRCPOLY, 916
SPI_DR_DR, 916
SPI_I2SCFGR_CHLEN, 917
SPI_I2SCFGR_CKPOL, 917
SPI_I2SCFGR_DATLEN, 917
SPI_I2SCFGR_DATLEN_0, 917
SPI_I2SCFGR_DATLEN_1, 917
SPI_I2SCFGR_I2SCFG, 917
SPI_I2SCFGR_I2SCFG_0, 917
SPI_I2SCFGR_I2SCFG_1, 918
SPI_I2SCFGR_I2SE, 918
SPI_I2SCFGR_I2SMOD, 918
SPI_I2SCFGR_I2SSTD, 918
SPI_I2SCFGR_I2SSTD_0, 918
SPI_I2SCFGR_I2SSTD_1, 918
SPI_I2SCFGR_PCMSYNC, 918
SPI_I2SPR_I2SDIV, 919
SPI_I2SPR_MCKOE, 919
SPI_I2SPR_ODD, 919
SPI_RXCRCR_RXCRC, 919
SPI_SR_BSY, 919
SPI_SR_CHSIDE, 919
SPI_SR_CRCERR, 919
SPI_SR_MODF, 920
SPI_SR_OVR, 920
SPI_SR_RXNE, 920
SPI_SR_TXE, 920
SPI_SR_UDR, 920
SPI_TXCRCR_TXCRC, 920
SYSCFG_CMPCR_CMP_PD, 920
SYSCFG_CMPCR_READY, 921
SYSCFG_EXTICR1_EXTI0, 921
SYSCFG_EXTICR1_EXTI0_PA, 921
SYSCFG_EXTICR1_EXTI0_PB, 921
SYSCFG_EXTICR1_EXTI0_PC, 921
SYSCFG_EXTICR1_EXTI0_PD, 921
SYSCFG_EXTICR1_EXTI0_PE, 921
SYSCFG_EXTICR1_EXTI0_PF, 922
SYSCFG_EXTICR1_EXTI0_PG, 922
SYSCFG_EXTICR1_EXTI0_PH, 922
SYSCFG_EXTICR1_EXTI0_PI, 922

SYSCFG_EXTICR1_EXTI0_PJ, 922
SYSCFG_EXTICR1_EXTI0_PK, 922
SYSCFG_EXTICR1_EXTI1, 922
SYSCFG_EXTICR1_EXTI1_PA, 922
SYSCFG_EXTICR1_EXTI1_PB, 923
SYSCFG_EXTICR1_EXTI1_PC, 923
SYSCFG_EXTICR1_EXTI1_PD, 923
SYSCFG_EXTICR1_EXTI1_PE, 923
SYSCFG_EXTICR1_EXTI1_PF, 923
SYSCFG_EXTICR1_EXTI1_PG, 923
SYSCFG_EXTICR1_EXTI1_PH, 923
SYSCFG_EXTICR1_EXTI1_PI, 924
SYSCFG_EXTICR1_EXTI1_PJ, 924
SYSCFG_EXTICR1_EXTI1_PK, 924
SYSCFG_EXTICR1_EXTI2, 924
SYSCFG_EXTICR1_EXTI2_PA, 924
SYSCFG_EXTICR1_EXTI2_PB, 924
SYSCFG_EXTICR1_EXTI2_PC, 924
SYSCFG_EXTICR1_EXTI2_PD, 925
SYSCFG_EXTICR1_EXTI2_PE, 925
SYSCFG_EXTICR1_EXTI2_PF, 925
SYSCFG_EXTICR1_EXTI2_PG, 925
SYSCFG_EXTICR1_EXTI2_PH, 925
SYSCFG_EXTICR1_EXTI2_PI, 925
SYSCFG_EXTICR1_EXTI2_PJ, 925
SYSCFG_EXTICR1_EXTI2_PK, 925
SYSCFG_EXTICR1_EXTI3, 926
SYSCFG_EXTICR1_EXTI3_PA, 926
SYSCFG_EXTICR1_EXTI3_PB, 926
SYSCFG_EXTICR1_EXTI3_PC, 926
SYSCFG_EXTICR1_EXTI3_PD, 926
SYSCFG_EXTICR1_EXTI3_PE, 926
SYSCFG_EXTICR1_EXTI3_PF, 926
SYSCFG_EXTICR1_EXTI3_PG, 927
SYSCFG_EXTICR1_EXTI3_PH, 927
SYSCFG_EXTICR1_EXTI3_PI, 927
SYSCFG_EXTICR1_EXTI3_PJ, 927
SYSCFG_EXTICR1_EXTI3_PK, 927
SYSCFG_EXTICR2_EXTI4, 927
SYSCFG_EXTICR2_EXTI4_PA, 927
SYSCFG_EXTICR2_EXTI4_PB, 928
SYSCFG_EXTICR2_EXTI4_PC, 928
SYSCFG_EXTICR2_EXTI4_PD, 928
SYSCFG_EXTICR2_EXTI4_PE, 928
SYSCFG_EXTICR2_EXTI4_PF, 928
SYSCFG_EXTICR2_EXTI4_PG, 928
SYSCFG_EXTICR2_EXTI4_PH, 928
SYSCFG_EXTICR2_EXTI4_PI, 929
SYSCFG_EXTICR2_EXTI4_PJ, 929
SYSCFG_EXTICR2_EXTI4_PK, 929
SYSCFG_EXTICR2_EXTI5, 929
SYSCFG_EXTICR2_EXTI5_PA, 929
SYSCFG_EXTICR2_EXTI5_PB, 929
SYSCFG_EXTICR2_EXTI5_PC, 929
SYSCFG_EXTICR2_EXTI5_PD, 930
SYSCFG_EXTICR2_EXTI5_PE, 930
SYSCFG_EXTICR2_EXTI5_PF, 930
SYSCFG_EXTICR2_EXTI5_PG, 930
SYSCFG_EXTICR2_EXTI5_PH, 930
SYSCFG_EXTICR2_EXTI5_PI, 930
SYSCFG_EXTICR2_EXTI5_PJ, 930
SYSCFG_EXTICR2_EXTI5_PK, 930
SYSCFG_EXTICR2_EXTI6, 931
SYSCFG_EXTICR2_EXTI6_PA, 931
SYSCFG_EXTICR2_EXTI6_PB, 931
SYSCFG_EXTICR2_EXTI6_PC, 931
SYSCFG_EXTICR2_EXTI6_PD, 931
SYSCFG_EXTICR2_EXTI6_PE, 931
SYSCFG_EXTICR2_EXTI6_PF, 931
SYSCFG_EXTICR2_EXTI6_PG, 932
SYSCFG_EXTICR2_EXTI6_PH, 932
SYSCFG_EXTICR2_EXTI6_PI, 932
SYSCFG_EXTICR2_EXTI6_PJ, 932
SYSCFG_EXTICR2_EXTI6_PK, 932
SYSCFG_EXTICR2_EXTI7, 932
SYSCFG_EXTICR2_EXTI7_PA, 932
SYSCFG_EXTICR2_EXTI7_PB, 933
SYSCFG_EXTICR2_EXTI7_PC, 933
SYSCFG_EXTICR2_EXTI7_PD, 933
SYSCFG_EXTICR2_EXTI7_PE, 933
SYSCFG_EXTICR2_EXTI7_PF, 933
SYSCFG_EXTICR2_EXTI7_PG, 933
SYSCFG_EXTICR2_EXTI7_PH, 933
SYSCFG_EXTICR2_EXTI7_PI, 934
SYSCFG_EXTICR2_EXTI7_PJ, 934
SYSCFG_EXTICR2_EXTI7_PK, 934
SYSCFG_EXTICR3_EXTI10, 934
SYSCFG_EXTICR3_EXTI10_PA, 934
SYSCFG_EXTICR3_EXTI10_PB, 934
SYSCFG_EXTICR3_EXTI10_PC, 934
SYSCFG_EXTICR3_EXTI10_PD, 935
SYSCFG_EXTICR3_EXTI10_PE, 935
SYSCFG_EXTICR3_EXTI10_PF, 935
SYSCFG_EXTICR3_EXTI10_PG, 935
SYSCFG_EXTICR3_EXTI10_PH, 935
SYSCFG_EXTICR3_EXTI10_PI, 935
SYSCFG_EXTICR3_EXTI10_PJ, 935
SYSCFG_EXTICR3_EXTI11, 935
SYSCFG_EXTICR3_EXTI11_PA, 936
SYSCFG_EXTICR3_EXTI11_PB, 936
SYSCFG_EXTICR3_EXTI11_PC, 936
SYSCFG_EXTICR3_EXTI11_PD, 936
SYSCFG_EXTICR3_EXTI11_PE, 936
SYSCFG_EXTICR3_EXTI11_PF, 936
SYSCFG_EXTICR3_EXTI11_PG, 936
SYSCFG_EXTICR3_EXTI11_PH, 937
SYSCFG_EXTICR3_EXTI11_PI, 937
SYSCFG_EXTICR3_EXTI11_PJ, 937
SYSCFG_EXTICR3_EXTI18, 937
SYSCFG_EXTICR3_EXTI18_PA, 937
SYSCFG_EXTICR3_EXTI18_PB, 937
SYSCFG_EXTICR3_EXTI18_PC, 937
SYSCFG_EXTICR3_EXTI18_PD, 938
SYSCFG_EXTICR3_EXTI18_PE, 938
SYSCFG_EXTICR3_EXTI18_PF, 938
SYSCFG_EXTICR3_EXTI18_PG, 938

SYSCFG_EXTICR3_EXTI8_PH, 938
SYSCFG_EXTICR3_EXTI8_PI, 938
SYSCFG_EXTICR3_EXTI8_PJ, 938
SYSCFG_EXTICR3_EXTI9, 938
SYSCFG_EXTICR3_EXTI9_PA, 939
SYSCFG_EXTICR3_EXTI9_PB, 939
SYSCFG_EXTICR3_EXTI9_PC, 939
SYSCFG_EXTICR3_EXTI9_PD, 939
SYSCFG_EXTICR3_EXTI9_PE, 939
SYSCFG_EXTICR3_EXTI9_PF, 939
SYSCFG_EXTICR3_EXTI9_PG, 939
SYSCFG_EXTICR3_EXTI9_PH, 940
SYSCFG_EXTICR3_EXTI9_PI, 940
SYSCFG_EXTICR3_EXTI9_PJ, 940
SYSCFG_EXTICR4_EXTI12, 940
SYSCFG_EXTICR4_EXTI12_PA, 940
SYSCFG_EXTICR4_EXTI12_PB, 940
SYSCFG_EXTICR4_EXTI12_PC, 940
SYSCFG_EXTICR4_EXTI12_PD, 941
SYSCFG_EXTICR4_EXTI12_PE, 941
SYSCFG_EXTICR4_EXTI12_PF, 941
SYSCFG_EXTICR4_EXTI12_PG, 941
SYSCFG_EXTICR4_EXTI12_PH, 941
SYSCFG_EXTICR4_EXTI12_PI, 941
SYSCFG_EXTICR4_EXTI12_PJ, 941
SYSCFG_EXTICR4_EXTI13, 941
SYSCFG_EXTICR4_EXTI13_PA, 942
SYSCFG_EXTICR4_EXTI13_PB, 942
SYSCFG_EXTICR4_EXTI13_PC, 942
SYSCFG_EXTICR4_EXTI13_PD, 942
SYSCFG_EXTICR4_EXTI13_PE, 942
SYSCFG_EXTICR4_EXTI13_PF, 942
SYSCFG_EXTICR4_EXTI13_PG, 942
SYSCFG_EXTICR4_EXTI13_PH, 943
SYSCFG_EXTICR4_EXTI13_PI, 943
SYSCFG_EXTICR4_EXTI13_PJ, 943
SYSCFG_EXTICR4_EXTI14, 943
SYSCFG_EXTICR4_EXTI14_PA, 943
SYSCFG_EXTICR4_EXTI14_PB, 943
SYSCFG_EXTICR4_EXTI14_PC, 943
SYSCFG_EXTICR4_EXTI14_PD, 944
SYSCFG_EXTICR4_EXTI14_PE, 944
SYSCFG_EXTICR4_EXTI14_PF, 944
SYSCFG_EXTICR4_EXTI14_PG, 944
SYSCFG_EXTICR4_EXTI14_PH, 944
SYSCFG_EXTICR4_EXTI14_PI, 944
SYSCFG_EXTICR4_EXTI14_PJ, 944
SYSCFG_EXTICR4_EXTI14, 944
SYSCFG_EXTICR4_EXTI15, 945
SYSCFG_EXTICR4_EXTI15_PA, 945
SYSCFG_EXTICR4_EXTI15_PB, 945
SYSCFG_EXTICR4_EXTI15_PC, 945
SYSCFG_EXTICR4_EXTI15_PD, 945
SYSCFG_EXTICR4_EXTI15_PE, 945
SYSCFG_EXTICR4_EXTI15_PF, 945
SYSCFG_EXTICR4_EXTI15_PG, 945
SYSCFG_EXTICR4_EXTI15_PH, 946
SYSCFG_EXTICR4_EXTI15_PI, 946
SYSCFG_EXTICR4_EXTI15_PJ, 946
SYSCFG_MEMRMP_FB_MODE, 946
SYSCFG_MEMRMP_MEM_MODE, 946
SYSCFG_MEMRMP_MEM_MODE_0, 946
SYSCFG_MEMRMP_MEM_MODE_1, 946
SYSCFG_MEMRMP_MEM_MODE_2, 946
SYSCFG_MEMRMP_SWP_FMC, 947
SYSCFG_MEMRMP_SWP_FMC_0, 947
SYSCFG_MEMRMP_SWP_FMC_1, 947
SYSCFG_PMC_ADC1DC2, 947
SYSCFG_PMC_ADC2DC2, 947
SYSCFG_PMC_ADC3DC2, 947
SYSCFG_PMC_ADCxDC2, 947
SYSCFG_PMC_MII_RMII_SEL, 948
TIM_ARR_ARR, 948
TIM_BDTR_AOE, 948
TIM_BDTR_BKE, 948
TIM_BDTR_BKP, 948
TIM_BDTR_DTG, 948
TIM_BDTR_DTG_0, 948
TIM_BDTR_DTG_1, 949
TIM_BDTR_DTG_2, 949
TIM_BDTR_DTG_3, 949
TIM_BDTR_DTG_4, 949
TIM_BDTR_DTG_5, 949
TIM_BDTR_DTG_6, 949
TIM_BDTR_DTG_7, 949
TIM_BDTR_LOCK, 949
TIM_BDTR_LOCK_0, 950
TIM_BDTR_LOCK_1, 950
TIM_BDTR_MOE, 950
TIM_BDTR_OSSI, 950
TIM_BDTR_OSSR, 950
TIM_CCER_CC1E, 950
TIM_CCER_CC1NE, 950
TIM_CCER_CC1NP, 951
TIM_CCER_CC1P, 951
TIM_CCER_CC2E, 951
TIM_CCER_CC2NE, 951
TIM_CCER_CC2NP, 951
TIM_CCER_CC2P, 951
TIM_CCER_CC3E, 951
TIM_CCER_CC3NE, 952
TIM_CCER_CC3NP, 952
TIM_CCER_CC3P, 952
TIM_CCER_CC4E, 952
TIM_CCER_CC4NP, 952
TIM_CCER_CC4P, 952
TIM_CCMR1_CC1S, 952
TIM_CCMR1_CC1S_0, 953
TIM_CCMR1_CC1S_1, 953
TIM_CCMR1_CC2S, 953
TIM_CCMR1_CC2S_0, 953
TIM_CCMR1_CC2S_1, 953
TIM_CCMR1_IC1F, 953
TIM_CCMR1_IC1F_0, 953
TIM_CCMR1_IC1F_1, 954
TIM_CCMR1_IC1F_2, 954
TIM_CCMR1_IC1F_3, 954

TIM_CCMR1_IC1PSC, 954
TIM_CCMR1_IC1PSC_0, 954
TIM_CCMR1_IC1PSC_1, 954
TIM_CCMR1_IC2F, 954
TIM_CCMR1_IC2F_0, 954
TIM_CCMR1_IC2F_1, 955
TIM_CCMR1_IC2F_2, 955
TIM_CCMR1_IC2F_3, 955
TIM_CCMR1_IC2PSC, 955
TIM_CCMR1_IC2PSC_0, 955
TIM_CCMR1_IC2PSC_1, 955
TIM_CCMR1_OC1CE, 955
TIM_CCMR1_OC1FE, 956
TIM_CCMR1_OC1M, 956
TIM_CCMR1_OC1M_0, 956
TIM_CCMR1_OC1M_1, 956
TIM_CCMR1_OC1M_2, 956
TIM_CCMR1_OC1PE, 956
TIM_CCMR1_OC2CE, 956
TIM_CCMR1_OC2FE, 957
TIM_CCMR1_OC2M, 957
TIM_CCMR1_OC2M_0, 957
TIM_CCMR1_OC2M_1, 957
TIM_CCMR1_OC2M_2, 957
TIM_CCMR1_OC2PE, 957
TIM_CCMR2_CC3S, 957
TIM_CCMR2_CC3S_0, 958
TIM_CCMR2_CC3S_1, 958
TIM_CCMR2_CC4S, 958
TIM_CCMR2_CC4S_0, 958
TIM_CCMR2_CC4S_1, 958
TIM_CCMR2_IC3F, 958
TIM_CCMR2_IC3F_0, 958
TIM_CCMR2_IC3F_1, 959
TIM_CCMR2_IC3F_2, 959
TIM_CCMR2_IC3F_3, 959
TIM_CCMR2_IC3PSC, 959
TIM_CCMR2_IC3PSC_0, 959
TIM_CCMR2_IC3PSC_1, 959
TIM_CCMR2_IC4F, 959
TIM_CCMR2_IC4F_0, 959
TIM_CCMR2_IC4F_1, 960
TIM_CCMR2_IC4F_2, 960
TIM_CCMR2_IC4F_3, 960
TIM_CCMR2_IC4PSC, 960
TIM_CCMR2_IC4PSC_0, 960
TIM_CCMR2_IC4PSC_1, 960
TIM_CCMR2_OC3CE, 960
TIM_CCMR2_OC3FE, 960
TIM_CCMR2_OC3M, 961
TIM_CCMR2_OC3M_0, 961
TIM_CCMR2_OC3M_1, 961
TIM_CCMR2_OC3M_2, 961
TIM_CCMR2_OC3PE, 961
TIM_CCMR2_OC4CE, 961
TIM_CCMR2_OC4FE, 961
TIM_CCMR2_OC4M, 962
TIM_CCMR2_OC4M_0, 962
TIM_CCMR2_OC4M_1, 962
TIM_CCMR2_OC4M_2, 962
TIM_CCMR2_OC4PE, 962
TIM_CCR1_CCR1, 962
TIM_CCR2_CCR2, 962
TIM_CCR3_CCR3, 963
TIM_CCR4_CCR4, 963
TIM_CNT_CNT, 963
TIM_CR1_ARPE, 963
TIM_CR1_CEN, 963
TIM_CR1_CKD, 963
TIM_CR1_CKD_0, 963
TIM_CR1_CKD_1, 964
TIM_CR1_CMS, 964
TIM_CR1_CMS_0, 964
TIM_CR1_CMS_1, 964
TIM_CR1_DIR, 964
TIM_CR1_OPM, 964
TIM_CR1_UDIS, 964
TIM_CR1_URS, 965
TIM_CR2_CCDS, 965
TIM_CR2_CCPC, 965
TIM_CR2_CCUS, 965
TIM_CR2_MMS, 965
TIM_CR2_MMS_0, 965
TIM_CR2_MMS_1, 965
TIM_CR2_MMS_2, 966
TIM_CR2_OIS1, 966
TIM_CR2_OIS1N, 966
TIM_CR2_OIS2, 966
TIM_CR2_OIS2N, 966
TIM_CR2_OIS3, 966
TIM_CR2_OIS3N, 966
TIM_CR2_OIS4, 967
TIM_CR2_TI1S, 967
TIM_DCR_DBAA, 967
TIM_DCR_DBAA_0, 967
TIM_DCR_DBAA_1, 967
TIM_DCR_DBAA_2, 967
TIM_DCR_DBAA_3, 967
TIM_DCR_DBAA_4, 968
TIM_DCR_DBBL, 968
TIM_DCR_DBBL_0, 968
TIM_DCR_DBBL_1, 968
TIM_DCR_DBBL_2, 968
TIM_DCR_DBBL_3, 968
TIM_DCR_DBBL_4, 968
TIM_DIER_BIE, 968
TIM_DIER_CC1DE, 969
TIM_DIER_CC1IE, 969
TIM_DIER_CC2DE, 969
TIM_DIER_CC2IE, 969
TIM_DIER_CC3DE, 969
TIM_DIER_CC3IE, 969
TIM_DIER_CC4DE, 969
TIM_DIER_CC4IE, 970
TIM_DIER_COMDE, 970
TIM_DIER_COMIE, 970

TIM_DIER_TDE, 970
TIM_DIER_TIE, 970
TIM_DIER_UDE, 970
TIM_DIER_UIE, 970
TIM_DMAR_DMAB, 971
TIM_EGR_BG, 971
TIM_EGR_CC1G, 971
TIM_EGR_CC2G, 971
TIM_EGR_CC3G, 971
TIM_EGR_CC4G, 971
TIM_EGR_COMG, 971
TIM_EGR_TG, 972
TIM_EGR_UG, 972
TIM_OR_ITR1_RMP, 972
TIM_OR_ITR1_RMP_0, 972
TIM_OR_ITR1_RMP_1, 972
TIM_OR_TI4_RMP, 972
TIM_OR_TI4_RMP_0, 972
TIM_OR_TI4_RMP_1, 973
TIM_PSC_PSC, 973
TIM_RCR REP, 973
TIM_SMCR_ECE, 973
TIM_SMCR ETF, 973
TIM_SMCR ETF_0, 973
TIM_SMCR ETF_1, 973
TIM_SMCR ETF_2, 974
TIM_SMCR ETF_3, 974
TIM_SMCR ETP, 974
TIM_SMCR ETPLS, 974
TIM_SMCR ETPLS_0, 974
TIM_SMCR ETPLS_1, 974
TIM_SMCR MSM, 974
TIM_SMCR SMS, 974
TIM_SMCR SMS_0, 975
TIM_SMCR SMS_1, 975
TIM_SMCR SMS_2, 975
TIM_SMCR TS, 975
TIM_SMCR TS_0, 975
TIM_SMCR TS_1, 975
TIM_SMCR TS_2, 975
TIM_SR_BIF, 976
TIM_SR_CC1IF, 976
TIM_SR_CC1OF, 976
TIM_SR_CC2IF, 976
TIM_SR_CC2OF, 976
TIM_SR_CC3IF, 976
TIM_SR_CC3OF, 976
TIM_SR_CC4IF, 977
TIM_SR_CC4OF, 977
TIM_SR_COMIF, 977
TIM_SR_TIF, 977
TIM_SR_UIF, 977
USART_BRR_DIV_Fraction, 977
USART_BRR_DIV_Mantissa, 977
USART_CR1_IDLEIE, 978
USART_CR1_M, 978
USART_CR1_OVER8, 978
USART_CR1_PCE, 978
USART_CR1_PEIE, 978
USART_CR1_PS, 978
USART_CR1_RE, 978
USART_CR1_RWU, 979
USART_CR1_RXNEIE, 979
USART_CR1_SBK, 979
USART_CR1_TCIE, 979
USART_CR1_TE, 979
USART_CR1_TXEIE, 979
USART_CR1_UE, 979
USART_CR1_WAKE, 980
USART_CR2_ADD, 980
USART_CR2_CLKEN, 980
USART_CR2_CPHA, 980
USART_CR2_CPOL, 980
USART_CR2_LBCL, 980
USART_CR2_LBDIE, 980
USART_CR2_LBDL, 981
USART_CR2_LINEN, 981
USART_CR2_STOP, 981
USART_CR2_STOP_0, 981
USART_CR2_STOP_1, 981
USART_CR3_CTSE, 981
USART_CR3_CTSIE, 981
USART_CR3_DMAR, 982
USART_CR3_DMAT, 982
USART_CR3_EIE, 982
USART_CR3_HDSEL, 982
USART_CR3_IREN, 982
USART_CR3_IRLP, 982
USART_CR3_NACK, 982
USART_CR3_ONEBIT, 983
USART_CR3_RTSE, 983
USART_CR3_SCEN, 983
USART_DR_DR, 983
USART_GTPR_GT, 983
USART_GTPR_PSC, 983
USART_GTPR_PSC_0, 983
USART_GTPR_PSC_1, 984
USART_GTPR_PSC_2, 984
USART_GTPR_PSC_3, 984
USART_GTPR_PSC_4, 984
USART_GTPR_PSC_5, 984
USART_GTPR_PSC_6, 984
USART_GTPR_PSC_7, 984
USART_SR_CTS, 984
USART_SR_FE, 985
USART_SR_IDLE, 985
USART_SR_LBD, 985
USART_SR_NE, 985
USART_SR_ORE, 985
USART_SR_PE, 985
USART_SR_RXNE, 985
USART_SR_TC, 986
USART_SR_TXE, 986
WWDG_CFR_EWI, 986
WWDG_CFR_W, 986
WWDG_CFR_W_0, 986

WWDG_CFR_W_1, 986
 WWDG_CFR_W_2, 986
 WWDG_CFR_W_3, 987
 WWDG_CFR_W_4, 987
 WWDG_CFR_W_5, 987
 WWDG_CFR_W_6, 987
 WWDG_CFR_WDGTB, 987
 WWDG_CFR_WDGTB_0, 987
 WWDG_CFR_WDGTB_1, 987
 WWDG_CR_T, 987
 WWDG_CR_T_0, 988
 WWDG_CR_T_1, 988
 WWDG_CR_T_2, 988
 WWDG_CR_T_3, 988
 WWDG_CR_T_4, 988
 WWDG_CR_T_5, 988
 WWDG_CR_T_6, 988
 WWDG_CR_WDGA, 988
 WWDG_SR_EWIF, 989
 Peripheral_registers_structures, 530
 PFCR
 CMSIS, 498
 PFR
 ITM Functions, 140
 PID0
 ITM Functions, 140
 PID1
 ITM Functions, 140
 PID2
 ITM Functions, 141
 PID3
 ITM Functions, 141
 PID4
 ITM Functions, 141
 PID5
 ITM Functions, 141
 PID6
 ITM Functions, 141
 PID7
 ITM Functions, 141
 Platform Standard Types, 148
 boolean, 152
 ConstVoidPtr, 152
 CPU_BIT_ORDER, 150
 CPU_BYTE_ORDER, 150
 CPU_TYPE, 150
 CPU_TYPE_16, 150
 CPU_TYPE_32, 150
 CPU_TYPE_64, 150
 CPU_TYPE_8, 150
 FALSE, 151
 FLAG_RESET, 151
 FLAG_SET, 151
 float32, 152
 float64, 153
 HIGH_BYTE_FIRST, 151
 LOW_BYTE_FIRST, 151
 LSB_FIRST, 151
 MSB_FIRST, 151
 RESET, 152
 SET, 152
 sint16, 153
 sint16_least, 153
 sint32, 153
 sint32_least, 153
 sint64, 153
 sint8, 153
 sint8_least, 154
 TRUE, 152
 uint16, 154
 uint16_least, 154
 uint32, 154
 uint32_least, 154
 uint64, 154
 uint8, 154
 uint8_least, 154
 VoidPtr, 155
 vuInt16_t, 155
 vuInt32_t, 155
 vuInt8_t, 155
 PLLCFG
 CMSIS, 499
 RCC_TypeDef, 1036
 PLLI2SCFGR
 CMSIS, 499
 RCC_TypeDef, 1036
 PLLSAICFGR
 CMSIS, 499
 PMC
 CMSIS, 499
 SYSCFG_RegDef_t, 1055
 POL
 CRC_TypeDef, 1005
 PORT
 ITM Functions, 141
 POWER
 CMSIS, 499
 PR
 CMSIS, 499
 EXTI_TypeDef, 1013
 PRER
 CMSIS, 500
 PrimaryAddress
 S_I2C_Slave_address, 1041
 Print_Slots_LCD_Flag
 Global_Variables_Definitions, 371
 Priority Groups, 288
 Prototypes, 356
 fp_App_State_Handler, 356
 PSC
 CMSIS, 500
 TIM1_TypeDef, 1059
 PUPDR
 CMSIS, 500
 GPIO_TypeDef, 1018
 PVD_IRQn

- Configuration_section_for\CMSIS, 525
- PWR_CR_ADCDC1
 Peripheral_Registers_Bits_Definition, 868
- PWR_CR_CSBF
 Peripheral_Registers_Bits_Definition, 869
- PWR_CR_CWUF
 Peripheral_Registers_Bits_Definition, 869
- PWR_CR_DBP
 Peripheral_Registers_Bits_Definition, 869
- PWR_CR_FISSR
 Peripheral_Registers_Bits_Definition, 869
- PWR_CR_FMSSR
 Peripheral_Registers_Bits_Definition, 869
- PWR_CR_LPDS
 Peripheral_Registers_Bits_Definition, 869
- PWR_CR_LPLVDS
 Peripheral_Registers_Bits_Definition, 870
- PWR_CR_LPUDS
 Peripheral_Registers_Bits_Definition, 870
- PWR_CR_MRLVDS
 Peripheral_Registers_Bits_Definition, 870
- PWR_CR_MRUDS
 Peripheral_Registers_Bits_Definition, 870
- PWR_CR_ODEN
 Peripheral_Registers_Bits_Definition, 870
- PWR_CR_ODSWEN
 Peripheral_Registers_Bits_Definition, 870
- PWR_CR_PDDS
 Peripheral_Registers_Bits_Definition, 870
- PWR_CR_PLS
 Peripheral_Registers_Bits_Definition, 871
- PWR_CR_PLS_0
 Peripheral_Registers_Bits_Definition, 871
- PWR_CR_PLS_1
 Peripheral_Registers_Bits_Definition, 871
- PWR_CR_PLS_2
 Peripheral_Registers_Bits_Definition, 871
- PWR_CR_PLS_LEV0
 Peripheral_Registers_Bits_Definition, 871
- PWR_CR_PLS_LEV1
 Peripheral_Registers_Bits_Definition, 871
- PWR_CR_PLS_LEV2
 Peripheral_Registers_Bits_Definition, 871
- PWR_CR_PLS_LEV3
 Peripheral_Registers_Bits_Definition, 872
- PWR_CR_PLS_LEV4
 Peripheral_Registers_Bits_Definition, 872
- PWR_CR_PLS_LEV5
 Peripheral_Registers_Bits_Definition, 872
- PWR_CR_PLS_LEV6
 Peripheral_Registers_Bits_Definition, 872
- PWR_CR_PLS_LEV7
 Peripheral_Registers_Bits_Definition, 872
- PWR_CR_PVDE
 Peripheral_Registers_Bits_Definition, 872
- PWR_CR_UDEN
 Peripheral_Registers_Bits_Definition, 872
- PWR_CR_UDEN_0
 Peripheral_Registers_Bits_Definition, 873
- PWR_CR_UDEN_1
 Peripheral_Registers_Bits_Definition, 873
- PWR_CR_VOS
 Peripheral_Registers_Bits_Definition, 873
- PWR_CR_VOS_0
 Peripheral_Registers_Bits_Definition, 873
- PWR_CR_VOS_1
 Peripheral_Registers_Bits_Definition, 873
- PWR_CSR_BRE
 Peripheral_Registers_Bits_Definition, 873
- PWR_CSR_BRR
 Peripheral_Registers_Bits_Definition, 873
- PWR_CSR_EWUP
 Peripheral_Registers_Bits_Definition, 874
- PWR_CSR_ODRDY
 Peripheral_Registers_Bits_Definition, 874
- PWR_CSR_ODSWRDY
 Peripheral_Registers_Bits_Definition, 874
- PWR_CSR_PVDO
 Peripheral_Registers_Bits_Definition, 874
- PWR_CSR_SBF
 Peripheral_Registers_Bits_Definition, 874
- PWR_CSR_UDSWRDY
 Peripheral_Registers_Bits_Definition, 874
- PWR_CSR_VOSRDY
 Peripheral_Registers_Bits_Definition, 874
- PWR_CSR_WUF
 Peripheral_Registers_Bits_Definition, 875
- PWR_CSR_WUPP
 Peripheral_Registers_Bits_Definition, 875
- PWR_TypeDef, 1031
- Q
- ITM Functions, 141, 142
- RCC
- RCC Instance, 173
- RCC Bit Position Definitions, 206
- RCC Driver, 305
- RCC Instance, 172
- RCC, 173
- RCC Register Map, 163
- RCC_AHB1LPENR Bit Position Definitions, 229
- RCC_AHB1LPENR_CRCEN, 229
- RCC_AHB1LPENR_DMA1LPEN, 230
- RCC_AHB1LPENR_DMA2LPEN, 230
- RCC_AHB1LPENR_ETHMACLPEN, 230
- RCC_AHB1LPENR_ETHMACPTPLPEN, 230
- RCC_AHB1LPENR_ETHMACRXLPEN, 230
- RCC_AHB1LPENR_ETHMACTXLPEN, 230
- RCC_AHB1LPENR_GPIOALPEN, 230
- RCC_AHB1LPENR_GPIOBLPEN, 230
- RCC_AHB1LPENR_GPIOCLPEN, 231
- RCC_AHB1LPENR_GPIODLPEN, 231
- RCC_AHB1LPENR_GPIOELPEN, 231
- RCC_AHB1LPENR_GPIOFLPEN, 231

- RCC_AHB1LPENR_GPIOGLPEN, [231](#)
- RCC_AHB1LPENR_GPIOHLPEN, [231](#)
- RCC_AHB1LPENR_GPIOILPEN, [231](#)
- RCC_AHB1LPENR_OTGHSHULPI, [231](#)
- RCC_AHB1LPENR_OTGHSLPEN, [232](#)
- RCC_AHB1LPENR_CRCEN
 - RCC_AHB1LPENR Bit Position Definitions, [229](#)
- RCC_AHB1LPENR_DMA1LPEN
 - RCC_AHB1LPENR Bit Position Definitions, [230](#)
- RCC_AHB1LPENR_DMA2LPEN
 - RCC_AHB1LPENR Bit Position Definitions, [230](#)
- RCC_AHB1LPENR_ETHMACLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [230](#)
- RCC_AHB1LPENR_ETHMACPTPLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [230](#)
- RCC_AHB1LPENR_ETHMACRXLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [230](#)
- RCC_AHB1LPENR_ETHMACTXLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [230](#)
- RCC_AHB1LPENR_GPIOALPEN
 - RCC_AHB1LPENR Bit Position Definitions, [230](#)
- RCC_AHB1LPENR_GPIOBLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [230](#)
- RCC_AHB1LPENR_GPIOCLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [231](#)
- RCC_AHB1LPENR_GPIOFLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [231](#)
- RCC_AHB1LPENR_GPIOGLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [231](#)
- RCC_AHB1LPENR_GPIOHLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [231](#)
- RCC_AHB1LPENR_GPIOILPEN
 - RCC_AHB1LPENR Bit Position Definitions, [231](#)
- RCC_AHB1LPENR_OTGHSHULPI
 - RCC_AHB1LPENR Bit Position Definitions, [231](#)
- RCC_AHB1LPENR_OTGHSLPEN
 - RCC_AHB1LPENR Bit Position Definitions, [232](#)
- RCC_AHB1RSTR Bit Position Definitions, [217](#)
 - RCC_AHB1RSTR_CRC, [218](#)
 - RCC_AHB1RSTR_DMA1, [218](#)
 - RCC_AHB1RSTR_DMA2, [218](#)
 - RCC_AHB1RSTR_ETHMAC, [218](#)
 - RCC_AHB1RSTR_GPIOA, [219](#)
 - RCC_AHB1RSTR_GPIOB, [219](#)
 - RCC_AHB1RSTR_GPIOC, [219](#)
 - RCC_AHB1RSTR_GPIOD, [219](#)
 - RCC_AHB1RSTR_GPIOE, [219](#)
 - RCC_AHB1RSTR_GPIOF, [219](#)
 - RCC_AHB1RSTR_GPIOG, [219](#)
 - RCC_AHB1RSTR_GPIOH, [219](#)
 - RCC_AHB1RSTR_GPIOI, [220](#)
 - RCC_AHB1RSTR_OTGHS, [220](#)
 - RCC_AHB1RSTR_OTGHSULPI, [220](#)
- RCC_AHB1RSTR_CRC
- RCC_AHB1RSTR_BIT POSITION DEFINITIONS
 - RCC_AHB1RSTR Bit Position Definitions, [218](#)
- RCC_AHB1RSTR_DMA1
 - RCC_AHB1RSTR Bit Position Definitions, [218](#)
- RCC_AHB1RSTR_DMA2
 - RCC_AHB1RSTR Bit Position Definitions, [218](#)
- RCC_AHB1RSTR_ETHMAC
 - RCC_AHB1RSTR Bit Position Definitions, [218](#)
- RCC_AHB1RSTR_ETHMACPTPLPEN
 - RCC_AHB1RSTR Bit Position Definitions, [218](#)
- RCC_AHB1RSTR_GPIOA
 - RCC_AHB1RSTR Bit Position Definitions, [219](#)
- RCC_AHB1RSTR_GPIOB
 - RCC_AHB1RSTR Bit Position Definitions, [219](#)
- RCC_AHB1RSTR_GPIOC
 - RCC_AHB1RSTR Bit Position Definitions, [219](#)
- RCC_AHB1RSTR_GPIOD
 - RCC_AHB1RSTR Bit Position Definitions, [219](#)
- RCC_AHB1RSTR_GPIOE
 - RCC_AHB1RSTR Bit Position Definitions, [219](#)
- RCC_AHB1RSTR_GPIOF
 - RCC_AHB1RSTR Bit Position Definitions, [219](#)
- RCC_AHB1RSTR_GPIOG
 - RCC_AHB1RSTR Bit Position Definitions, [219](#)
- RCC_AHB1RSTR_GPIOH
 - RCC_AHB1RSTR Bit Position Definitions, [219](#)
- RCC_AHB1RSTR_OTGHS
 - RCC_AHB1RSTR Bit Position Definitions, [220](#)
- RCC_AHB1RSTR_OTGHSULPI
 - RCC_AHB1RSTR Bit Position Definitions, [220](#)
- RCC_AHB2LPENR Bit Position Definitions, [232](#)
 - RCC_AHB2LPENR_CRYPLPEN, [232](#)
 - RCC_AHB2LPENR_DCMILPEN, [232](#)
 - RCC_AHB2LPENR_HASHLPEN, [233](#)
 - RCC_AHB2LPENR_OTGFSLPEN, [233](#)
 - RCC_AHB2LPENR_RNGLPEN, [233](#)
- RCC_AHB2LPENR_CRYPLPEN
 - RCC_AHB2LPENR Bit Position Definitions, [232](#)
- RCC_AHB2LPENR_DCMILPEN
 - RCC_AHB2LPENR Bit Position Definitions, [232](#)
- RCC_AHB2LPENR_HASHLPEN
 - RCC_AHB2LPENR Bit Position Definitions, [233](#)
- RCC_AHB2LPENR_OTGFSLPEN
 - RCC_AHB2LPENR Bit Position Definitions, [233](#)
- RCC_AHB2LPENR_RNGLPEN
 - RCC_AHB2LPENR Bit Position Definitions, [233](#)
- RCC_AHB2RSTR Bit Position Definitions, [220](#)
 - RCC_AHB2RSTR_CRYP, [221](#)
 - RCC_AHB2RSTR_DCMI, [221](#)
 - RCC_AHB2RSTR_HASH, [221](#)
 - RCC_AHB2RSTR_OTGFS, [221](#)
 - RCC_AHB2RSTR RNG, [221](#)
- RCC_AHB2RSTR_CRYP
 - RCC_AHB2RSTR Bit Position Definitions, [221](#)
- RCC_AHB2RSTR_DCMI
 - RCC_AHB2RSTR Bit Position Definitions, [221](#)
- RCC_AHB2RSTR_HASH
 - RCC_AHB2RSTR Bit Position Definitions, [221](#)
- RCC_AHB2RSTR_OTGFS

- RCC_AHB2RSTR Bit Position Definitions, [221](#)
- RCC_AHB2RSTR_RNG
 - RCC_AHB2RSTR Bit Position Definitions, [221](#)
- RCC_AHB3LPENR Bit Position Definitions, [233](#)
 - RCC_AHB3LPENR_FSMCLPEN, [234](#)
- RCC_AHB3LPENR_FSMCLPEN
 - RCC_AHB3LPENR Bit Position Definitions, [234](#)
- RCC_AHB3RSTR Bit Position Definitions, [222](#)
 - RCC_AHB3RSTR_FSMC, [222](#)
- RCC_AHB3RSTR_FSMC
 - RCC_AHB3RSTR Bit Position Definitions, [222](#)
- RCC_APB1LPENR Bit Position Definitions, [234](#)
 - RCC_APB1LPENR_CAN1LPEN, [235](#)
 - RCC_APB1LPENR_CAN2LPEN, [235](#)
 - RCC_APB1LPENR_DACLPEN, [235](#)
 - RCC_APB1LPENR_I2C1LPEN, [235](#)
 - RCC_APB1LPENR_I2C2LPEN, [235](#)
 - RCC_APB1LPENR_I2C3LPEN, [235](#)
 - RCC_APB1LPENR_PWRLPEN, [235](#)
 - RCC_APB1LPENR_SPI2LPEN, [236](#)
 - RCC_APB1LPENR_SPI3LPEN, [236](#)
 - RCC_APB1LPENR_TIM12LPEN, [236](#)
 - RCC_APB1LPENR_TIM13LPEN, [236](#)
 - RCC_APB1LPENR_TIM14LPEN, [236](#)
 - RCC_APB1LPENR_TIM2LPEN, [236](#)
 - RCC_APB1LPENR_TIM3LPEN, [236](#)
 - RCC_APB1LPENR_TIM4LPEN, [236](#)
 - RCC_APB1LPENR_TIM5LPEN, [237](#)
 - RCC_APB1LPENR_TIM6LPEN, [237](#)
 - RCC_APB1LPENR_TIM7LPEN, [237](#)
 - RCC_APB1LPENR_UART4LPEN, [237](#)
 - RCC_APB1LPENR_UART5LPEN, [237](#)
 - RCC_APB1LPENR_UART7LPEN, [237](#)
 - RCC_APB1LPENR_UART8LPEN, [237](#)
 - RCC_APB1LPENR_USART2LPEN, [237](#)
 - RCC_APB1LPENR_USART3LPEN, [238](#)
 - RCC_APB1LPENR_WWDGLPEN, [238](#)
- RCC_APB1LPENR_CAN1LPEN
 - RCC_APB1LPENR Bit Position Definitions, [235](#)
- RCC_APB1LPENR_CAN2LPEN
 - RCC_APB1LPENR Bit Position Definitions, [235](#)
- RCC_APB1LPENR_DACLPEN
 - RCC_APB1LPENR Bit Position Definitions, [235](#)
- RCC_APB1LPENR_I2C1LPEN
 - RCC_APB1LPENR Bit Position Definitions, [235](#)
- RCC_APB1LPENR_I2C2LPEN
 - RCC_APB1LPENR Bit Position Definitions, [235](#)
- RCC_APB1LPENR_I2C3LPEN
 - RCC_APB1LPENR Bit Position Definitions, [235](#)
- RCC_APB1LPENR_PWRLPEN
 - RCC_APB1LPENR Bit Position Definitions, [235](#)
- RCC_APB1LPENR_SPI2LPEN
 - RCC_APB1LPENR Bit Position Definitions, [236](#)
- RCC_APB1LPENR_SPI3LPEN
 - RCC_APB1LPENR Bit Position Definitions, [236](#)
- RCC_APB1LPENR_TIM12LPEN
 - RCC_APB1LPENR Bit Position Definitions, [236](#)
- RCC_APB1RSTR Bit Position Definitions, [222](#)
 - RCC_APB1RSTR_CAN1, [223](#)
 - RCC_APB1RSTR_CAN2, [223](#)
 - RCC_APB1RSTR_DAC, [223](#)
 - RCC_APB1RSTR_I2C1, [224](#)
 - RCC_APB1RSTR_I2C2, [224](#)
 - RCC_APB1RSTR_I2C3, [224](#)
 - RCC_APB1RSTR_PWR, [224](#)
 - RCC_APB1RSTR_SPI2, [224](#)
 - RCC_APB1RSTR_SPI3, [224](#)
 - RCC_APB1RSTR_TIM12, [224](#)
 - RCC_APB1RSTR_TIM13, [224](#)
 - RCC_APB1RSTR_TIM14, [225](#)
 - RCC_APB1RSTR_TIM2, [225](#)
 - RCC_APB1RSTR_TIM3, [225](#)
 - RCC_APB1RSTR_TIM4, [225](#)
 - RCC_APB1RSTR_TIM5, [225](#)
 - RCC_APB1RSTR_TIM6, [225](#)
 - RCC_APB1RSTR_TIM7, [225](#)
 - RCC_APB1RSTR_UART4, [225](#)
 - RCC_APB1RSTR_UART5, [226](#)
 - RCC_APB1RSTR_UART7, [226](#)
 - RCC_APB1RSTR_UART8, [226](#)
 - RCC_APB1RSTR_USART2, [226](#)
 - RCC_APB1RSTR_USART3, [226](#)
 - RCC_APB1RSTR_WWDG, [226](#)
- RCC_APB1RSTR_CAN1
 - RCC_APB1RSTR Bit Position Definitions, [223](#)
- RCC_APB1RSTR_CAN2
 - RCC_APB1RSTR Bit Position Definitions, [223](#)

RCC_APB1RSTR Bit Position Definitions, [223](#)
RCC_APB1RSTR_DAC
 RCC_APB1RSTR Bit Position Definitions, [223](#)
RCC_APB1RSTR_I2C1
 RCC_APB1RSTR Bit Position Definitions, [224](#)
RCC_APB1RSTR_I2C2
 RCC_APB1RSTR Bit Position Definitions, [224](#)
RCC_APB1RSTR_I2C3
 RCC_APB1RSTR Bit Position Definitions, [224](#)
RCC_APB1RSTR_PWR
 RCC_APB1RSTR Bit Position Definitions, [224](#)
RCC_APB1RSTR_SPI2
 RCC_APB1RSTR Bit Position Definitions, [224](#)
RCC_APB1RSTR_SPI3
 RCC_APB1RSTR Bit Position Definitions, [224](#)
RCC_APB1RSTR_TIM12
 RCC_APB1RSTR Bit Position Definitions, [224](#)
RCC_APB1RSTR_TIM13
 RCC_APB1RSTR Bit Position Definitions, [224](#)
RCC_APB1RSTR_TIM14
 RCC_APB1RSTR Bit Position Definitions, [225](#)
RCC_APB1RSTR_TIM2
 RCC_APB1RSTR Bit Position Definitions, [225](#)
RCC_APB1RSTR_TIM3
 RCC_APB1RSTR Bit Position Definitions, [225](#)
RCC_APB1RSTR_TIM4
 RCC_APB1RSTR Bit Position Definitions, [225](#)
RCC_APB1RSTR_TIM5
 RCC_APB1RSTR Bit Position Definitions, [225](#)
RCC_APB1RSTR_TIM6
 RCC_APB1RSTR Bit Position Definitions, [225](#)
RCC_APB1RSTR_TIM7
 RCC_APB1RSTR Bit Position Definitions, [225](#)
RCC_APB1RSTR_UART4
 RCC_APB1RSTR Bit Position Definitions, [225](#)
RCC_APB1RSTR_UART5
 RCC_APB1RSTR Bit Position Definitions, [226](#)
RCC_APB1RSTR_UART7
 RCC_APB1RSTR Bit Position Definitions, [226](#)
RCC_APB1RSTR_UART8
 RCC_APB1RSTR Bit Position Definitions, [226](#)
RCC_APB1RSTR_USART2
 RCC_APB1RSTR Bit Position Definitions, [226](#)
RCC_APB1RSTR_USART3
 RCC_APB1RSTR Bit Position Definitions, [226](#)
RCC_APB1RSTR_WWDG
 RCC_APB1RSTR Bit Position Definitions, [226](#)
RCC_APB2LPENR Bit Position Definitions, [238](#)
 RCC_APB2LPENR_ADCLPEN, [239](#)
 RCC_APB2LPENR_SDOLPEN, [239](#)
 RCC_APB2LPENR_SPI1LPEN, [239](#)
 RCC_APB2LPENR_SYSCFGLPEN, [239](#)
 RCC_APB2LPENR_TIM10LPEN, [239](#)
 RCC_APB2LPENR_TIM11LPEN, [239](#)
 RCC_APB2LPENR_TIM1LPEN, [239](#)
 RCC_APB2LPENR_TIM8LPEN, [240](#)
 RCC_APB2LPENR_TIM9LPEN, [240](#)
 RCC_APB2LPENR_USART1LPEN, [240](#)
 RCC_APB2LPENR_USART6LPEN
 RCC_APB2LPENR Bit Position Definitions, [240](#)
RCC_APB2LPENR_ADCLPEN
 RCC_APB2LPENR Bit Position Definitions, [239](#)
RCC_APB2LPENR_SDOLPEN
 RCC_APB2LPENR Bit Position Definitions, [239](#)
RCC_APB2LPENR_SPI1LPEN
 RCC_APB2LPENR Bit Position Definitions, [239](#)
RCC_APB2LPENR_SYSCFGLPEN
 RCC_APB2LPENR Bit Position Definitions, [239](#)
RCC_APB2LPENR_TIM10LPEN
 RCC_APB2LPENR Bit Position Definitions, [239](#)
RCC_APB2LPENR_TIM11LPEN
 RCC_APB2LPENR Bit Position Definitions, [239](#)
RCC_APB2LPENR_TIM1LPEN
 RCC_APB2LPENR Bit Position Definitions, [239](#)
RCC_APB2LPENR_TIM8LPEN
 RCC_APB2LPENR Bit Position Definitions, [240](#)
RCC_APB2LPENR_TIM9LPEN
 RCC_APB2LPENR Bit Position Definitions, [240](#)
RCC_APB2LPENR_USART1LPEN
 RCC_APB2LPENR Bit Position Definitions, [240](#)
RCC_APB2LPENR_USART6LPEN
 RCC_APB2LPENR Bit Position Definitions, [240](#)
RCC_APB2RSTR Bit Position Definitions, [227](#)
 RCC_APB2RSTR_ADC, [227](#)
 RCC_APB2RSTR_SDIO, [227](#)
 RCC_APB2RSTR_SPI1, [227](#)
 RCC_APB2RSTR_SYSCFG, [228](#)
 RCC_APB2RSTR_TIM1, [228](#)
 RCC_APB2RSTR_TIM10, [228](#)
 RCC_APB2RSTR_TIM11, [228](#)
 RCC_APB2RSTR_TIM8, [228](#)
 RCC_APB2RSTR_TIM9, [228](#)
 RCC_APB2RSTR_USART1, [228](#)
 RCC_APB2RSTR_USART6, [228](#)
RCC_APB2RSTR_ADC
 RCC_APB2RSTR Bit Position Definitions, [227](#)
RCC_APB2RSTR_SDIO
 RCC_APB2RSTR Bit Position Definitions, [227](#)
RCC_APB2RSTR_SPI1
 RCC_APB2RSTR Bit Position Definitions, [227](#)
RCC_APB2RSTR_SYSCFG
 RCC_APB2RSTR Bit Position Definitions, [228](#)
RCC_APB2RSTR_TIM1
 RCC_APB2RSTR Bit Position Definitions, [228](#)
RCC_APB2RSTR_TIM10
 RCC_APB2RSTR Bit Position Definitions, [228](#)
RCC_APB2RSTR_TIM11
 RCC_APB2RSTR Bit Position Definitions, [228](#)
RCC_APB2RSTR_TIM8
 RCC_APB2RSTR Bit Position Definitions, [228](#)
RCC_APB2RSTR_TIM9
 RCC_APB2RSTR Bit Position Definitions, [228](#)
RCC_APB2RSTR_USART1
 RCC_APB2RSTR Bit Position Definitions, [228](#)
RCC_APB2RSTR_USART6
 RCC_APB2RSTR Bit Position Definitions, [228](#)
RCC_BDCR Bit Position Definitions, [240](#)

RCC_BDCR_BDRST, [241](#)
RCC_BDCR_LSEBYP, [241](#)
RCC_BDCR_LSEON, [241](#)
RCC_BDCR_LSERDY, [241](#)
RCC_BDCR_RTCEN, [241](#)
RCC_BDCR_RTCSEL, [241](#)
RCC_BDCR_BDRST
 RCC_BDCR Bit Position Definitions, [241](#)
RCC_BDCR_LSEBYP
 RCC_BDCR Bit Position Definitions, [241](#)
RCC_BDCR_LSEON
 RCC_BDCR Bit Position Definitions, [241](#)
RCC_BDCR_LSERDY
 RCC_BDCR Bit Position Definitions, [241](#)
RCC_BDCR_RTCEN
 RCC_BDCR Bit Position Definitions, [241](#)
RCC_BDCR_RTCSEL
 RCC_BDCR Bit Position Definitions, [241](#)
RCC_CFGR Bit Position Definitions, [211](#)
 RCC_CFGR_HPRE, [212](#)
 RCC_CFGR_I2SSRC, [212](#)
 RCC_CFGR_MCO1, [212](#)
 RCC_CFGR_MCO1PRE, [212](#)
 RCC_CFGR_MCO2, [212](#)
 RCC_CFGR_MCO2PRE, [213](#)
 RCC_CFGR_PPREG1, [213](#)
 RCC_CFGR_PPREG2, [213](#)
 RCC_CFGR_RTCPRE, [213](#)
 RCC_CFGR_SW, [213](#)
 RCC_CFGR_SWS, [213](#)
RCC_CFGR_HPRE
 Peripheral Registers Bits Definition, [875](#)
 RCC_CFGR Bit Position Definitions, [212](#)
RCC_CFGR_HPRE_0
 Peripheral Registers Bits Definition, [875](#)
RCC_CFGR_HPRE_1
 Peripheral Registers Bits Definition, [875](#)
RCC_CFGR_HPRE_2
 Peripheral Registers Bits Definition, [875](#)
RCC_CFGR_HPRE_3
 Peripheral Registers Bits Definition, [875](#)
RCC_CFGR_HPRE_DIV1
 Peripheral Registers Bits Definition, [876](#)
RCC_CFGR_HPRE_DIV128
 Peripheral Registers Bits Definition, [876](#)
RCC_CFGR_HPRE_DIV16
 Peripheral Registers Bits Definition, [876](#)
RCC_CFGR_HPRE_DIV2
 Peripheral Registers Bits Definition, [876](#)
RCC_CFGR_HPRE_DIV256
 Peripheral Registers Bits Definition, [876](#)
RCC_CFGR_HPRE_DIV4
 Peripheral Registers Bits Definition, [876](#)
RCC_CFGR_HPRE_DIV512
 Peripheral Registers Bits Definition, [876](#)
RCC_CFGR_HPRE_DIV64
 Peripheral Registers Bits Definition, [876](#)
RCC_CFGR_HPRE_DIV8
 Peripheral Registers Bits Definition, [877](#)
RCC_CFGR_I2SSRC
 RCC_CFGR Bit Position Definitions, [212](#)
RCC_CFGR_MCO1
 RCC_CFGR Bit Position Definitions, [212](#)
RCC_CFGR_MCO1PRE
 RCC_CFGR Bit Position Definitions, [212](#)
RCC_CFGR_MCO2
 RCC_CFGR Bit Position Definitions, [212](#)
RCC_CFGR_MCO2PRE
 RCC_CFGR Bit Position Definitions, [213](#)
RCC_CFGR_PPREG1
 Peripheral Registers Bits Definition, [877](#)
 RCC_CFGR Bit Position Definitions, [213](#)
RCC_CFGR_PPREG1_0
 Peripheral Registers Bits Definition, [877](#)
RCC_CFGR_PPREG1_1
 Peripheral Registers Bits Definition, [877](#)
RCC_CFGR_PPREG1_2
 Peripheral Registers Bits Definition, [877](#)
RCC_CFGR_PPREG1_DIV1
 Peripheral Registers Bits Definition, [877](#)
RCC_CFGR_PPREG1_DIV16
 Peripheral Registers Bits Definition, [877](#)
RCC_CFGR_PPREG1_DIV2
 Peripheral Registers Bits Definition, [877](#)
RCC_CFGR_PPREG1_DIV4
 Peripheral Registers Bits Definition, [878](#)
RCC_CFGR_PPREG1_DIV8
 Peripheral Registers Bits Definition, [878](#)
RCC_CFGR_PPREG2
 Peripheral Registers Bits Definition, [878](#)
 RCC_CFGR Bit Position Definitions, [213](#)
RCC_CFGR_PPREG2_0
 Peripheral Registers Bits Definition, [878](#)
RCC_CFGR_PPREG2_1
 Peripheral Registers Bits Definition, [878](#)
RCC_CFGR_PPREG2_2
 Peripheral Registers Bits Definition, [878](#)
RCC_CFGR_PPREG2_DIV1
 Peripheral Registers Bits Definition, [878](#)
RCC_CFGR_PPREG2_DIV16
 Peripheral Registers Bits Definition, [878](#)
RCC_CFGR_PPREG2_DIV2
 Peripheral Registers Bits Definition, [879](#)
RCC_CFGR_PPREG2_DIV4
 Peripheral Registers Bits Definition, [879](#)
RCC_CFGR_PPREG2_DIV8
 Peripheral Registers Bits Definition, [879](#)
RCC_CFGR_RTCPRE
 RCC_CFGR Bit Position Definitions, [213](#)
RCC_CFGR_RTCPRE_4
 Peripheral Registers Bits Definition, [879](#)
RCC_CFGR_SW
 Peripheral Registers Bits Definition, [879](#)
 RCC_CFGR Bit Position Definitions, [213](#)
RCC_CFGR_SW_0
 Peripheral Registers Bits Definition, [879](#)

RCC_CFGR_SW_1
 Peripheral_Registers_Bits_Definition, [879](#)
RCC_CFGR_SW_HSE
 Peripheral_Registers_Bits_Definition, [879](#)
RCC_CFGR_SW_HSI
 Peripheral_Registers_Bits_Definition, [880](#)
RCC_CFGR_SW_PLL
 Peripheral_Registers_Bits_Definition, [880](#)
RCC_CFGR_SWS
 Peripheral_Registers_Bits_Definition, [880](#)
 RCC_CFGR Bit Position Definitions, [213](#)
RCC_CFGR_SWS_0
 Peripheral_Registers_Bits_Definition, [880](#)
RCC_CFGR_SWS_1
 Peripheral_Registers_Bits_Definition, [880](#)
RCC_CFGR_SWS_HSE
 Peripheral_Registers_Bits_Definition, [880](#)
RCC_CFGR_SWS_HSI
 Peripheral_Registers_Bits_Definition, [880](#)
RCC_CFGR_SWS_PLL
 Peripheral_Registers_Bits_Definition, [880](#)
RCC_CIR Bit Position Definitions, [214](#)
 RCC_CIR_CSSF, [214](#)
 RCC_CIR_HSERDYC, [215](#)
 RCC_CIR_HSERDYF, [215](#)
 RCC_CIR_HSERDYIE, [215](#)
 RCC_CIR_HSIRDYC, [215](#)
 RCC_CIR_HSIRDYF, [215](#)
 RCC_CIR_HSIRDYIE, [215](#)
 RCC_CIR_LSERDYC, [215](#)
 RCC_CIR_LSERDYF, [215](#)
 RCC_CIR_LSERDYIE, [216](#)
 RCC_CIR_LSIRDYC, [216](#)
 RCC_CIR_LSIRDYF, [216](#)
 RCC_CIR_LSIRDYIE, [216](#)
 RCC_CIR_PLLI2SRDYC, [216](#)
 RCC_CIR_PLLI2SRDYF, [216](#)
 RCC_CIR_PLLI2SRDYIE, [216](#)
 RCC_CIR_PLLRDYC, [216](#)
 RCC_CIR_PLLRDYF, [217](#)
 RCC_CIR_PLLRDYIE, [217](#)
 RCC_CIR_PLLSAIRDYC, [217](#)
 RCC_CIR_PLLSAIRDYF, [217](#)
 RCC_CIR_PLLSAIRDYIE, [217](#)
RCC_CIR_CSSF
 RCC_CIR Bit Position Definitions, [214](#)
RCC_CIR_HSERDYC
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_HSERDYF
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_HSERDYIE
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_HSIRDYC
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_HSIRDYF
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_HSIRDYIE
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_HSIRDYC
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_HSIRDYF
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_HSIRDYIE
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_LSERDYC
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_LSERDYF
 RCC_CIR Bit Position Definitions, [215](#)
RCC_CIR_LSERDYIE
 RCC_CIR Bit Position Definitions, [216](#)
RCC_CIR_LSIRDYC
 RCC_CIR Bit Position Definitions, [216](#)
RCC_CIR_LSIRDYF
 RCC_CIR Bit Position Definitions, [216](#)
RCC_CIR_LSIRDYIE
 RCC_CIR Bit Position Definitions, [216](#)
RCC_CIR_PLLI2SRDYC
 RCC_CIR Bit Position Definitions, [216](#)
RCC_CIR_PLLI2SRDYF
 RCC_CIR Bit Position Definitions, [216](#)
RCC_CIR_PLLI2SRDYIE
 RCC_CIR Bit Position Definitions, [216](#)
RCC_CIR_PLLRDYC
 RCC_CIR Bit Position Definitions, [216](#)
RCC_CIR_PLLRDYF
 RCC_CIR Bit Position Definitions, [217](#)
RCC_CIR_PLLRDYIE
 RCC_CIR Bit Position Definitions, [217](#)
RCC_CIR_PLLSAIRDYC
 RCC_CIR Bit Position Definitions, [217](#)
RCC_CIR_PLLSAIRDYF
 RCC_CIR Bit Position Definitions, [217](#)
RCC_CIR_PLLSAIRDYIE
 RCC_CIR Bit Position Definitions, [217](#)
RCC_CR Bit Position Definitions, [207](#)
 RCC_CR_CSSON, [208](#)
 RCC_CR_HSEBYP, [208](#)
 RCC_CR_HSEON, [208](#)
 RCC_CR_HSERDY, [208](#)
 RCC_CR_HSICAL, [208](#)
 RCC_CR_HSION, [209](#)
 RCC_CR_HSIRDY, [209](#)
 RCC_CR_HSITRIM, [209](#)
 RCC_CR_PLLI2SON, [209](#)
 RCC_CR_PLLI2SRDY, [209](#)
 RCC_CR_PLLON, [209](#)
 RCC_CR_PLLRDY, [209](#)
 RCC_CR_PLLSAION, [209](#)
 RCC_CR_PLLSAIRDY, [210](#)
RCC_CR_CSSON
 RCC_CR Bit Position Definitions, [208](#)
RCC_CR_HSEBYP
 RCC_CR Bit Position Definitions, [208](#)
RCC_CR_HSEON
 RCC_CR Bit Position Definitions, [208](#)
RCC_CR_HSERDY
 RCC_CR Bit Position Definitions, [208](#)
RCC_CR_HSICAL
 RCC_CR Bit Position Definitions, [208](#)
RCC_CR_HSICAL_0
 Peripheral_Registers_Bits_Definition, [881](#)
RCC_CR_HSICAL_1

Peripheral_Registers_Bits_Definition, 881
RCC_CR_HSICAL_2
 Peripheral_Registers_Bits_Definition, 881
RCC_CR_HSICAL_3
 Peripheral_Registers_Bits_Definition, 881
RCC_CR_HSICAL_4
 Peripheral_Registers_Bits_Definition, 881
RCC_CR_HSICAL_5
 Peripheral_Registers_Bits_Definition, 881
RCC_CR_HSICAL_6
 Peripheral_Registers_Bits_Definition, 881
RCC_CR_HSICAL_7
 Peripheral_Registers_Bits_Definition, 882
RCC_CR_HSION
 RCC_CR Bit Position Definitions, 209
RCC_CR_HSIRDY
 RCC_CR Bit Position Definitions, 209
RCC_CR_HSITRIM
 RCC_CR Bit Position Definitions, 209
RCC_CR_HSITRIM_0
 Peripheral_Registers_Bits_Definition, 882
RCC_CR_HSITRIM_1
 Peripheral_Registers_Bits_Definition, 882
RCC_CR_HSITRIM_2
 Peripheral_Registers_Bits_Definition, 882
RCC_CR_HSITRIM_3
 Peripheral_Registers_Bits_Definition, 882
RCC_CR_HSITRIM_4
 Peripheral_Registers_Bits_Definition, 882
RCC_CR_PLLI2SON
 RCC_CR Bit Position Definitions, 209
RCC_CR_PLLI2SRDY
 RCC_CR Bit Position Definitions, 209
RCC_CR_PLLON
 RCC_CR Bit Position Definitions, 209
RCC_CR_PLLRDY
 RCC_CR Bit Position Definitions, 209
RCC_CR_PLLSAION
 RCC_CR Bit Position Definitions, 209
RCC_CR_PLLSAIRDY
 RCC_CR Bit Position Definitions, 210
RCC_CRC
 Peripheral Selection, 308
RCC_CSR Bit Position Definitions, 242
 RCC_CSR_IWDGRSTF, 242
 RCC_CSR_LPWRSTF, 242
 RCC_CSR_LSION, 243
 RCC_CSR_LSIRDY, 243
 RCC_CSR_OBLRSTF, 243
 RCC_CSR_PINRSTF, 243
 RCC_CSR_PORRSTF, 243
 RCC_CSR_RMVF, 243
 RCC_CSR_SFTRSTF, 243
 RCC_CSR_WWDGRSTF, 243
RCC_CSR_IWDGRSTF
 RCC_CSR Bit Position Definitions, 242
RCC_CSR_LPWRSTF
 RCC_CSR Bit Position Definitions, 242
RCC_CSR_LSION
 RCC_CSR Bit Position Definitions, 243
RCC_CSR_LSIRDY
 RCC_CSR Bit Position Definitions, 243
RCC_CSR_OBLRSTF
 RCC_CSR Bit Position Definitions, 243
RCC_CSR_PINRSTF
 RCC_CSR Bit Position Definitions, 243
RCC_CSR_PORRSTF
 RCC_CSR Bit Position Definitions, 243
RCC_CSR_RMVF
 RCC_CSR Bit Position Definitions, 243
RCC_CSR_SFTRSTF
 RCC_CSR Bit Position Definitions, 243
RCC_CSR_WWDGRSTF
 RCC_CSR Bit Position Definitions, 243
RCC_GetPLLOutputClock
 API Function Prototypes, 314
RCC_GPIOA
 Peripheral Selection, 308
RCC_GPIOB
 Peripheral Selection, 308
RCC_GPIOC
 Peripheral Selection, 309
RCC_GPIOD
 Peripheral Selection, 309
RCC_GPIOE
 Peripheral Selection, 309
RCC_I2C1
 Peripheral Selection, 309
RCC_I2C2
 Peripheral Selection, 309
RCC_IRQn
 Configuration_section_for_CMSIS, 525
RCC_PLLCFG Bit Position Definitions, 210
 RCC_PLLCFG_PLLM, 210
 RCC_PLLCFG_PLLN, 211
 RCC_PLLCFG_PLLP, 211
 RCC_PLLCFG_PLLQ, 211
 RCC_PLLCFG_PLLSRC, 211
RCC_PLLCFG_PLLM
 RCC_PLLCFG Bit Position Definitions, 210
RCC_PLLCFG_PLLN
 RCC_PLLCFG Bit Position Definitions, 211
RCC_PLLCFG_PLLP
 RCC_PLLCFG Bit Position Definitions, 211
RCC_PLLCFG_PLLQ
 RCC_PLLCFG Bit Position Definitions, 211
RCC_PLLCFG_PLLSRC
 RCC_PLLCFG Bit Position Definitions, 211
RCC_PLLCFG_PLLSRC_HSE
 Clock Source Selection, 307
RCC_PLLI2SCFGR Bit Position Definitions, 245
 RCC_PLLI2SCFGR_PLLI2SN, 245
 RCC_PLLI2SCFGR_PLLI2SR, 245
RCC_PLLI2SCFGR_PLLI2SN
 RCC_PLLI2SCFGR Bit Position Definitions, 245
RCC_PLLI2SCFGR_PLLI2SR

RCC_PLLI2SCFGR Bit Position Definitions, [245](#)
RCC_SELECT_HSE
 Clock Source Selection, [307](#)
RCC_SELECT_HSI
 Clock Source Selection, [307](#)
RCC_SELECT_PLL
 Clock Source Selection, [307](#)
RCC_SPI1
 Peripheral Selection, [309](#)
RCC_SPI2
 Peripheral Selection, [309](#)
RCC_SSCGR Bit Position Definitions, [244](#)
 RCC_SSCGR_INCSTEP, [244](#)
 RCC_SSCGR_MODPER, [244](#)
 RCC_SSCGR_SPREADSEL, [244](#)
 RCC_SSCGR_SSCGEN, [245](#)
RCC_SSCGR_INCSTEP
 RCC_SSCGR Bit Position Definitions, [244](#)
RCC_SSCGR_MODPER
 RCC_SSCGR Bit Position Definitions, [244](#)
RCC_SSCGR_SPREADSEL
 RCC_SSCGR Bit Position Definitions, [244](#)
RCC_SSCGR_SSCGEN
 RCC_SSCGR Bit Position Definitions, [245](#)
RCC_SYSCFG
 Peripheral Selection, [309](#)
RCC_TIM2
 Peripheral Selection, [310](#)
RCC_TypeDef, [1032](#)
 AHB1ENR, [1033](#)
 AHB1LPENR, [1033](#)
 AHB1RSTR, [1034](#)
 AHB2ENR, [1034](#)
 AHB2LPENR, [1034](#)
 AHB2RSTR, [1034](#)
 APB1ENR, [1034](#)
 APB1LPENR, [1034](#)
 APB1RSTR, [1034](#)
 APB2ENR, [1034](#)
 APB2LPENR, [1035](#)
 APB2RSTR, [1035](#)
 BDCR, [1035](#)
 CFGR, [1035](#)
 CIR, [1035](#)
 CR, [1035](#)
 CSR, [1035](#)
 DCKCFGR, [1035](#)
 PLLCFGR, [1036](#)
 PLLI2SCFGR, [1036](#)
 RESERVED0, [1036](#)
 RESERVED1, [1036](#)
 RESERVED2, [1036](#)
 RESERVED3, [1036](#)
 RESERVED4, [1036](#)
 RESERVED5, [1036](#)
 RESERVED6, [1037](#)
 RESERVED7, [1037](#)
 SSCGR, [1037](#)

 RCC_USART1
 Peripheral Selection, [310](#)
RCC_USART2
 Peripheral Selection, [310](#)
RCC_USART6
 Peripheral Selection, [310](#)
RCR
 CMSIS, [500](#)
 TIM1_TypeDef, [1059](#)
RDHR
 CMSIS, [500](#)
RDLR
 CMSIS, [500](#)
RDTR
 CMSIS, [500](#)
reload_value
 STK_config_t, [1053](#)
Repeated
 I2C Driver, [270](#)
Repeated_Start
 I2C Driver, [270](#)
RESERVED
 CMSIS, [501](#)
 CRC_TypeDef, [1005](#)
 SCB_TypeDef, [1048](#)
Reserved
 S_IRQ_SRC, [1042](#)
RESERVED0
 CMSIS, [501, 502](#)
 NVIC_TypeDef, [1030](#)
 RCC_TypeDef, [1036](#)
RESERVED1
 CMSIS, [502–504](#)
 NVIC_TypeDef, [1030](#)
 RCC_TypeDef, [1036](#)
 SYSCFG_RegDef_t, [1055](#)
RESERVED10
 CMSIS, [504](#)
RESERVED11
 CMSIS, [504](#)
RESERVED12
 CMSIS, [504](#)
RESERVED13
 CMSIS, [504](#)
RESERVED14
 CMSIS, [504](#)
RESERVED2
 CMSIS, [505](#)
 NVIC_TypeDef, [1031](#)
 RCC_TypeDef, [1036](#)
 SYSCFG_RegDef_t, [1055](#)
RESERVED3
 CMSIS, [506](#)
 NVIC_TypeDef, [1031](#)
 RCC_TypeDef, [1036](#)
RESERVED4
 CMSIS, [507](#)
 NVIC_TypeDef, [1031](#)

RCC_TypeDef, 1036
RESERVED5
 CMSIS, 507, 508
 NVIC_TypeDef, 1031
 RCC_TypeDef, 1036
RESERVED6
 CMSIS, 508, 509
 RCC_TypeDef, 1037
RESERVED7
 CMSIS, 509
 RCC_TypeDef, 1037
RESERVED8
 CMSIS, 510
RESERVED9
 CMSIS, 510
RESET
 Platform Standard Types, 152
Reset
 Exported_types, 530
 I2C Driver, 269
RESP1
 CMSIS, 510
RESP2
 CMSIS, 510
RESP3
 CMSIS, 511
RESP4
 CMSIS, 511
RESPCMD
 CMSIS, 511
RF0R
 CMSIS, 511
RF1R
 CMSIS, 511
RIR
 CMSIS, 511
RISR
 CMSIS, 511, 512
RLR
 CMSIS, 512
RNG_TypeDef, 1037
Rows
 LCD_t, 1026
RS_PIN
 LCD_t, 1026
RTC_Alarm_IRQHandler
 Configuration_section_for\CMSIS, 526
RTC_TypeDef, 1038
RTC_WKUP_IRQHandler
 Configuration_section_for\CMSIS, 525
RTSR
 CMSIS, 512
 EXTI_TypeDef, 1013
running_mode
 STK_config_t, 1053
RXCRCR
 CMSIS, 512
 SPI_TypeDef, 1051
RXE
 S_IRQ_SRC, 1042
s32
 Exported_types, 528
S_I2C_Config_t, 1039
 General_Call_Address_Detection, 1039
 I2C_Ack_Conrtol, 1040
 I2C_Mode, 1040
 I2C_Slave_address, 1040
 I2C_Speed, 1040
 I2C_stretchmode, 1040
 P_Slave_Event_CallBack, 1040
S_I2C_Slave_address, 1041
 Enable_Dual_Address, 1041
 I2C_Addressing_Mode, 1041
 PrimaryAddress, 1041
 SecondaryAddress, 1041
S_IRQ_SRC, 1042
 ERRI, 1042
 Reserved, 1042
 RXE, 1042
 TXE, 1042
S_SPI_Config_t, 1043
 Clock_Phase, 1043
 Clock_Polarity, 1043
 Communication_Mode, 1044
 Device_Mode, 1044
 Frame_Format, 1044
 IRQ_Enable, 1044
 NSS, 1044
 P_IRQ_CallBack, 1044
 Payload_Length, 1044
 SPI_Prescaler, 1044
SAI_Block_TypeDef, 1045
SAI_GCR_SYNCIN
 Peripheral_Registers_Bits_Definition, 882
SAI_GCR_SYNCIN_0
 Peripheral_Registers_Bits_Definition, 882
SAI_GCR_SYNCIN_1
 Peripheral_Registers_Bits_Definition, 883
SAI_GCR_SYNCOUT
 Peripheral_Registers_Bits_Definition, 883
SAI_GCR_SYNCOUT_0
 Peripheral_Registers_Bits_Definition, 883
SAI_GCR_SYNCOUT_1
 Peripheral_Registers_Bits_Definition, 883
SAI_TypeDef, 1045
SAI_xCLRFR_CAFSDET
 Peripheral_Registers_Bits_Definition, 883
SAI_xCLRFR_CCNRDY
 Peripheral_Registers_Bits_Definition, 883
SAI_xCLRFR_CFREQ
 Peripheral_Registers_Bits_Definition, 883
SAI_xCLRFR_CLFSDET
 Peripheral_Registers_Bits_Definition, 884
SAI_xCLRFR_CMUTEDET
 Peripheral_Registers_Bits_Definition, 884
SAI_xCLRFR_COVRUDR

Peripheral_Registers_Bits_Definition, 884
SAI_xCLRFR_CWCKCFG
 Peripheral_Registers_Bits_Definition, 884
SAI_xCR1_CKSTR
 Peripheral_Registers_Bits_Definition, 884
SAI_xCR1_DMAEN
 Peripheral_Registers_Bits_Definition, 884
SAI_xCR1_DS
 Peripheral_Registers_Bits_Definition, 884
SAI_xCR1_DS_0
 Peripheral_Registers_Bits_Definition, 885
SAI_xCR1_DS_1
 Peripheral_Registers_Bits_Definition, 885
SAI_xCR1_DS_2
 Peripheral_Registers_Bits_Definition, 885
SAI_xCR1_LSBFIRST
 Peripheral_Registers_Bits_Definition, 885
SAI_xCR1_MCKDIV
 Peripheral_Registers_Bits_Definition, 885
SAI_xCR1_MCKDIV_0
 Peripheral_Registers_Bits_Definition, 885
SAI_xCR1_MCKDIV_1
 Peripheral_Registers_Bits_Definition, 885
SAI_xCR1_MCKDIV_2
 Peripheral_Registers_Bits_Definition, 886
SAI_xCR1_MCKDIV_3
 Peripheral_Registers_Bits_Definition, 886
SAI_xCR1_MODE
 Peripheral_Registers_Bits_Definition, 886
SAI_xCR1_MODE_0
 Peripheral_Registers_Bits_Definition, 886
SAI_xCR1_MODE_1
 Peripheral_Registers_Bits_Definition, 886
SAI_xCR1_MONO
 Peripheral_Registers_Bits_Definition, 886
SAI_xCR1_NODIV
 Peripheral_Registers_Bits_Definition, 886
SAI_xCR1_OUTDRV
 Peripheral_Registers_Bits_Definition, 887
SAI_xCR1_PRTCFG
 Peripheral_Registers_Bits_Definition, 887
SAI_xCR1_PRTCFG_0
 Peripheral_Registers_Bits_Definition, 887
SAI_xCR1_PRTCFG_1
 Peripheral_Registers_Bits_Definition, 887
SAI_xCR1_SAIEN
 Peripheral_Registers_Bits_Definition, 887
SAI_xCR1_SYNCEN
 Peripheral_Registers_Bits_Definition, 887
SAI_xCR1_SYNCEN_0
 Peripheral_Registers_Bits_Definition, 887
SAI_xCR1_SYNCEN_1
 Peripheral_Registers_Bits_Definition, 888
SAI_xCR2_COMP
 Peripheral_Registers_Bits_Definition, 888
SAI_xCR2_COMP_0
 Peripheral_Registers_Bits_Definition, 888
SAI_xCR2_COMP_1
 Peripheral_Registers_Bits_Definition, 888
SAI_xCR2_CPL
 Peripheral_Registers_Bits_Definition, 888
SAI_xCR2_FFLUSH
 Peripheral_Registers_Bits_Definition, 888
SAI_xCR2_FTH
 Peripheral_Registers_Bits_Definition, 888
SAI_xCR2_FTH_0
 Peripheral_Registers_Bits_Definition, 889
SAI_xCR2_FTH_1
 Peripheral_Registers_Bits_Definition, 889
SAI_xCR2_MUTE
 Peripheral_Registers_Bits_Definition, 889
SAI_xCR2_MUTECNT
 Peripheral_Registers_Bits_Definition, 889
SAI_xCR2_MUTECNT_0
 Peripheral_Registers_Bits_Definition, 889
SAI_xCR2_MUTECNT_1
 Peripheral_Registers_Bits_Definition, 889
SAI_xCR2_MUTECNT_2
 Peripheral_Registers_Bits_Definition, 889
SAI_xCR2_MUTECNT_3
 Peripheral_Registers_Bits_Definition, 890
SAI_xCR2_MUTECNT_4
 Peripheral_Registers_Bits_Definition, 890
SAI_xCR2_MUTECNT_5
 Peripheral_Registers_Bits_Definition, 890
SAI_xCR2_MUTEVAL
 Peripheral_Registers_Bits_Definition, 890
SAI_xCR2_TRIS
 Peripheral_Registers_Bits_Definition, 890
SAI_xFRCR_FRL
 Peripheral_Registers_Bits_Definition, 890
SAI_xFRCR_FRL_0
 Peripheral_Registers_Bits_Definition, 890
SAI_xFRCR_FRL_1
 Peripheral_Registers_Bits_Definition, 891
SAI_xFRCR_FRL_2
 Peripheral_Registers_Bits_Definition, 891
SAI_xFRCR_FRL_3
 Peripheral_Registers_Bits_Definition, 891
SAI_xFRCR_FRL_4
 Peripheral_Registers_Bits_Definition, 891
SAI_xFRCR_FRL_5
 Peripheral_Registers_Bits_Definition, 891
SAI_xFRCR_FRL_6
 Peripheral_Registers_Bits_Definition, 891
SAI_xFRCR_FRL_7
 Peripheral_Registers_Bits_Definition, 891
SAI_xFRCR_FSALL
 Peripheral_Registers_Bits_Definition, 891
SAI_xFRCR_FSALL_0
 Peripheral_Registers_Bits_Definition, 892
SAI_xFRCR_FSALL_1
 Peripheral_Registers_Bits_Definition, 892
SAI_xFRCR_FSALL_2
 Peripheral_Registers_Bits_Definition, 892
SAI_xFRCR_FSALL_3

Peripheral_Registers_Bits_Definition, 892
SAI_xFRCR_FSALL_4
Peripheral_Registers_Bits_Definition, 892
SAI_xFRCR_FSALL_5
Peripheral_Registers_Bits_Definition, 892
SAI_xFRCR_FSALL_6
Peripheral_Registers_Bits_Definition, 892
SAI_xFRCR_FSDEF
Peripheral_Registers_Bits_Definition, 893
SAI_xFRCR_FSOFF
Peripheral_Registers_Bits_Definition, 893
SAI_xFRCR_FSPOL
Peripheral_Registers_Bits_Definition, 893
SAI_xIMR_AFSDETIE
Peripheral_Registers_Bits_Definition, 893
SAI_xIMR_CNRDYIE
Peripheral_Registers_Bits_Definition, 893
SAI_xIMR_FREQIE
Peripheral_Registers_Bits_Definition, 893
SAI_xIMR_LFSDETIE
Peripheral_Registers_Bits_Definition, 893
SAI_xIMR_MUTEDETIE
Peripheral_Registers_Bits_Definition, 894
SAI_xIMR_OVRUDRIE
Peripheral_Registers_Bits_Definition, 894
SAI_xIMR_WCKCFGIE
Peripheral_Registers_Bits_Definition, 894
SAI_xSLOTR_FBOFF
Peripheral_Registers_Bits_Definition, 894
SAI_xSLOTR_FBOFF_0
Peripheral_Registers_Bits_Definition, 894
SAI_xSLOTR_FBOFF_1
Peripheral_Registers_Bits_Definition, 894
SAI_xSLOTR_FBOFF_2
Peripheral_Registers_Bits_Definition, 894
SAI_xSLOTR_FBOFF_3
Peripheral_Registers_Bits_Definition, 895
SAI_xSLOTR_FBOFF_4
Peripheral_Registers_Bits_Definition, 895
SAI_xSLOTR_NBSLOT
Peripheral_Registers_Bits_Definition, 895
SAI_xSLOTR_NBSLOT_0
Peripheral_Registers_Bits_Definition, 895
SAI_xSLOTR_NBSLOT_1
Peripheral_Registers_Bits_Definition, 895
SAI_xSLOTR_NBSLOT_2
Peripheral_Registers_Bits_Definition, 895
SAI_xSLOTR_NBSLOT_3
Peripheral_Registers_Bits_Definition, 895
SAI_xSLOTR_SLOTEN
Peripheral_Registers_Bits_Definition, 896
SAI_xSLOTR_SLOTSZ
Peripheral_Registers_Bits_Definition, 896
SAI_xSLOTR_SLOT SZ_0
Peripheral_Registers_Bits_Definition, 896
SAI_xSLOTR_SLOT SZ_1
Peripheral_Registers_Bits_Definition, 896
SAI_xSR_AFSDET
Peripheral_Registers_Bits_Definition, 896
SAI_xSR_CNRDY
Peripheral_Registers_Bits_Definition, 896
SAI_xSR_FLVL
Peripheral_Registers_Bits_Definition, 896
SAI_xSR_FLVL_0
Peripheral_Registers_Bits_Definition, 897
SAI_xSR_FLVL_1
Peripheral_Registers_Bits_Definition, 897
SAI_xSR_FLVL_2
Peripheral_Registers_Bits_Definition, 897
SAI_xSR_FREQ
Peripheral_Registers_Bits_Definition, 897
SAI_xSR_LFSDET
Peripheral_Registers_Bits_Definition, 897
SAI_xSR_MUTEDET
Peripheral_Registers_Bits_Definition, 897
SAI_xSR_OVRUDR
Peripheral_Registers_Bits_Definition, 897
SAI_xSR_WCKCFG
Peripheral_Registers_Bits_Definition, 898
sc16
Exported_types, 528
sc32
Exported_types, 528
sc8
Exported_types, 528
SCB
CMSIS, 423
Core Definitions, 110
SCB Instance, 170
SCB Instance, 169
SCB, 170
SCB Register Map, 162
SCB_AIRCR_ENDIANESS_Msk
CMSIS, 423
System Control Block (SCB), 53
SCB_AIRCR_ENDIANESS_Pos
CMSIS, 423
System Control Block (SCB), 53
SCB_AIRCR_PRIGROUP_Msk
CMSIS, 423
System Control Block (SCB), 53
SCB_AIRCR_PRIGROUP_Pos
CMSIS, 423
System Control Block (SCB), 53
SCB_AIRCR_SYSRESETREQ_Msk
CMSIS, 423
System Control Block (SCB), 53
SCB_AIRCR_SYSRESETREQ_Pos
CMSIS, 423
System Control Block (SCB), 54
SCB_AIRCR_VECTCLRACTIVE_Msk
CMSIS, 424
System Control Block (SCB), 54
SCB_AIRCR_VECTCLRACTIVE_Pos
CMSIS, 424
System Control Block (SCB), 54

SCB_AIRCR_VECTKEY_Msk
CMSIS, 424
System Control Block (SCB), 54

SCB_AIRCR_VECTKEY_Pos
CMSIS, 424
System Control Block (SCB), 54

SCB_AIRCR_VECTKEYSTAT_Msk
CMSIS, 424
System Control Block (SCB), 54

SCB_AIRCR_VECTKEYSTAT_Pos
CMSIS, 424
System Control Block (SCB), 54

SCB_AIRCR_VECTRESET_Msk
CMSIS, 424
System Control Block (SCB), 54

SCB_AIRCR_VECTRESET_Pos
CMSIS, 424
System Control Block (SCB), 55

SCB_BASE
CMSIS, 425
Core Definitions, 110

SCB_CCR_BFHFNIGN_Msk
CMSIS, 425
System Control Block (SCB), 55

SCB_CCR_BFHFNIGN_Pos
CMSIS, 425
System Control Block (SCB), 55

SCB_CCR_DIV_0_TRP_Msk
CMSIS, 425
System Control Block (SCB), 55

SCB_CCR_DIV_0_TRP_Pos
CMSIS, 425
System Control Block (SCB), 55

SCB_CCR_NONBASETHRDENA_Msk
CMSIS, 425
System Control Block (SCB), 55

SCB_CCR_NONBASETHRDENA_Pos
CMSIS, 425
System Control Block (SCB), 55

SCB_CCR_STKALIGN_Msk
CMSIS, 426
System Control Block (SCB), 55

SCB_CCR_STKALIGN_Pos
CMSIS, 426
System Control Block (SCB), 56

SCB_CCR_UNALIGN_TRP_Msk
CMSIS, 426
System Control Block (SCB), 56

SCB_CCR_UNALIGN_TRP_Pos
CMSIS, 426
System Control Block (SCB), 56

SCB_CCR_USERSETMPEND_Msk
CMSIS, 426
System Control Block (SCB), 56

SCB_CCR_USERSETMPEND_Pos
CMSIS, 426
System Control Block (SCB), 56

SCB_CFSR_BUSFAULTSR_Msk
CMSIS, 426
System Control Block (SCB), 56

SCB_CFSR_BUSFAULTSR_Pos
CMSIS, 426
System Control Block (SCB), 56

SCB_CFSR_MEMFAULTSR_Msk
CMSIS, 427
System Control Block (SCB), 56

SCB_CFSR_MEMFAULTSR_Pos
CMSIS, 427
System Control Block (SCB), 57

SCB_CFSR_USGFAULTSR_Msk
CMSIS, 427
System Control Block (SCB), 57

SCB_CFSR_USGFAULTSR_Pos
CMSIS, 427
System Control Block (SCB), 57

SCB_CPUID_ARCHITECTURE_Msk
CMSIS, 427
System Control Block (SCB), 57

SCB_CPUID_ARCHITECTURE_Pos
CMSIS, 427
System Control Block (SCB), 57

SCB_CPUID_IMPLEMENTER_Msk
CMSIS, 427
System Control Block (SCB), 57

SCB_CPUID_IMPLEMENTER_Pos
CMSIS, 427
System Control Block (SCB), 57

SCB_CPUID_PARTNO_Msk
CMSIS, 428
System Control Block (SCB), 57

SCB_CPUID_PARTNO_Pos
CMSIS, 428
System Control Block (SCB), 58

SCB_CPUID_REVISION_Msk
CMSIS, 428
System Control Block (SCB), 58

SCB_CPUID_REVISION_Pos
CMSIS, 428
System Control Block (SCB), 58

SCB_CPUID_VARIANT_Msk
CMSIS, 428
System Control Block (SCB), 58

SCB_CPUID_VARIANT_Pos
CMSIS, 428
System Control Block (SCB), 58

SCB_DFSR_BKPT_Msk
CMSIS, 428
System Control Block (SCB), 58

SCB_DFSR_BKPT_Pos
CMSIS, 428
System Control Block (SCB), 58

SCB_DFSR_DWTTRAP_Msk
CMSIS, 429
System Control Block (SCB), 58

SCB_DFSR_DWTTRAP_Pos
CMSIS, 429
System Control Block (SCB), 58

System Control Block (SCB), 59
SCB_DFSR_EXTERNAL_Msk
CMSIS, 429
System Control Block (SCB), 59
SCB_DFSR_EXTERNAL_Pos
CMSIS, 429
System Control Block (SCB), 59
SCB_DFSR_HALTED_Msk
CMSIS, 429
System Control Block (SCB), 59
SCB_DFSR_HALTED_Pos
CMSIS, 429
System Control Block (SCB), 59
SCB_DFSR_VCATCH_Msk
CMSIS, 429
System Control Block (SCB), 59
SCB_DFSR_VCATCH_Pos
CMSIS, 429
System Control Block (SCB), 59
SCB_HFSR_DEBUGEVT_Msk
CMSIS, 430
System Control Block (SCB), 59
SCB_HFSR_DEBUGEVT_Pos
CMSIS, 430
System Control Block (SCB), 60
SCB_HFSR_FORCED_Msk
CMSIS, 430
System Control Block (SCB), 60
SCB_HFSR_FORCED_Pos
CMSIS, 430
System Control Block (SCB), 60
SCB_HFSR_VECTTBL_Msk
CMSIS, 430
System Control Block (SCB), 60
SCB_HFSR_VECTTBL_Pos
CMSIS, 430
System Control Block (SCB), 60
SCB_ICSR_ISRPENDING_Msk
CMSIS, 430
System Control Block (SCB), 60
SCB_ICSR_ISRPENDING_Pos
CMSIS, 430
System Control Block (SCB), 60
SCB_ICSR_ISRPREEMPT_Msk
CMSIS, 431
System Control Block (SCB), 60
SCB_ICSR_ISRPREEMPT_Pos
CMSIS, 431
System Control Block (SCB), 61
SCB_ICSR_NMIPENDSET_Msk
CMSIS, 431
System Control Block (SCB), 61
SCB_ICSR_NMIPENDSET_Pos
CMSIS, 431
System Control Block (SCB), 61
SCB_ICSR_PENDSTCLR_Msk
CMSIS, 431
System Control Block (SCB), 61
SCB_ICSR_PENDSTCLR_Pos
CMSIS, 431
System Control Block (SCB), 61
SCB_ICSR_PENDSTSET_Msk
CMSIS, 431
System Control Block (SCB), 61
SCB_ICSR_PENDSTSET_Pos
CMSIS, 431
System Control Block (SCB), 61
SCB_ICSR_PENDSVCLR_Msk
CMSIS, 432
System Control Block (SCB), 61
SCB_ICSR_PENDSVCLR_Pos
CMSIS, 432
System Control Block (SCB), 62
SCB_ICSR_PENDSVSET_Msk
CMSIS, 432
System Control Block (SCB), 62
SCB_ICSR_PENDSVSET_Pos
CMSIS, 432
System Control Block (SCB), 62
SCB_ICSR_RETTOBASE_Msk
CMSIS, 432
System Control Block (SCB), 62
SCB_ICSR_RETTOBASE_Pos
CMSIS, 432
System Control Block (SCB), 62
SCB_ICSR_VECTACTIVE_Msk
CMSIS, 432
System Control Block (SCB), 62
SCB_ICSR_VECTACTIVE_Pos
CMSIS, 432
System Control Block (SCB), 62
SCB_ICSR_VECTPENDING_Msk
CMSIS, 433
System Control Block (SCB), 62
SCB_ICSR_VECTPENDING_Pos
CMSIS, 433
System Control Block (SCB), 63
SCB_SCR_SEVONPEND_Msk
CMSIS, 433
System Control Block (SCB), 63
SCB_SCR_SEVONPEND_Pos
CMSIS, 433
System Control Block (SCB), 63
SCB_SCR_SLEEPDEEP_Msk
CMSIS, 433
System Control Block (SCB), 63
SCB_SCR_SLEEPDEEP_Pos
CMSIS, 433
System Control Block (SCB), 63
SCB_SCR_SLEEPONEXIT_Msk
CMSIS, 433
System Control Block (SCB), 63
SCB_SCR_SLEEPONEXIT_Pos
CMSIS, 433
System Control Block (SCB), 63
SCB_SHCSR_BUSFAULTACT_Msk

CMSIS, 434
 System Control Block (SCB), 63
SCB_SHCSR_BUSFAULTACT_Pos
 CMSIS, 434
 System Control Block (SCB), 64
SCB_SHCSR_BUSFAULTENA_Msk
 CMSIS, 434
 System Control Block (SCB), 64
SCB_SHCSR_BUSFAULTENA_Pos
 CMSIS, 434
 System Control Block (SCB), 64
SCB_SHCSR_BUSFAULTPENDED_Msk
 CMSIS, 434
 System Control Block (SCB), 64
SCB_SHCSR_BUSFAULTPENDED_Pos
 CMSIS, 434
 System Control Block (SCB), 64
SCB_SHCSR_MEMFAULTACT_Msk
 CMSIS, 434
 System Control Block (SCB), 64
SCB_SHCSR_MEMFAULTACT_Pos
 CMSIS, 434
 System Control Block (SCB), 64
SCB_SHCSR_MEMFAULTENA_Msk
 CMSIS, 435
 System Control Block (SCB), 64
SCB_SHCSR_MEMFAULTENA_Pos
 CMSIS, 435
 System Control Block (SCB), 64
SCB_SHCSR_MEMFAULTPENDED_Msk
 CMSIS, 435
 System Control Block (SCB), 64
SCB_SHCSR_MEMFAULTPENDED_Pos
 CMSIS, 435
 System Control Block (SCB), 64
SCB_SHCSR_MONITORACT_Msk
 CMSIS, 435
 System Control Block (SCB), 64
SCB_SHCSR_MONITORACT_Pos
 CMSIS, 435
 System Control Block (SCB), 64
SCB_SHCSR_PENDSVACT_Msk
 CMSIS, 435
 System Control Block (SCB), 64
SCB_SHCSR_PENDSVACT_Pos
 CMSIS, 435
 System Control Block (SCB), 64
SCB_SHCSR_SVCALLACT_Msk
 CMSIS, 436
 System Control Block (SCB), 64
SCB_SHCSR_SVCALLACT_Pos
 CMSIS, 436
 System Control Block (SCB), 64
SCB_SHCSR_SVCALLPENDED_Msk
 CMSIS, 436
 System Control Block (SCB), 64
SCB_SHCSR_SVCALLPENDED_Pos
 CMSIS, 436
System Control Block (SCB), 66
SCB_SHCSR_SYSTICKACT_Msk
 CMSIS, 436
 System Control Block (SCB), 66
SCB_SHCSR_SYSTICKACT_Pos
 CMSIS, 436
 System Control Block (SCB), 66
SCB_SHCSR_USGFAULTACT_Msk
 CMSIS, 436
 System Control Block (SCB), 66
SCB_SHCSR_USGFAULTACT_Pos
 CMSIS, 436
 System Control Block (SCB), 66
SCB_SHCSR_USGFAULTENA_Msk
 CMSIS, 437
 System Control Block (SCB), 66
SCB_SHCSR_USGFAULTENA_Pos
 CMSIS, 437
 System Control Block (SCB), 67
SCB_SHCSR_USGFAULTPENDED_Msk
 CMSIS, 437
 System Control Block (SCB), 67
SCB_SHCSR_USGFAULTPENDED_Pos
 CMSIS, 437
 System Control Block (SCB), 67
SCB_Type, 1046
SCB_TypeDef, 1046
 AIRCR, 1047
 BFAR, 1047
 CCR, 1047
 CFSR, 1047
 CPUID, 1047
 HFSR, 1047
 ICSR, 1047
 MMAR, 1047
 RESERVED, 1048
 SCR, 1048
 SHCSR, 1048
 SHP, 1048
 VTOR, 1048
SCB_VTOR_TBLOFF_Msk
 CMSIS, 437
 System Control Block (SCB), 67
SCB_VTOR_TBLOFF_Pos
 CMSIS, 437
 System Control Block (SCB), 67
SCnSCB
 CMSIS, 437
 Core Definitions, 110
SCnSCB_ACTLR_DISDEFWBUF_Msk
 CMSIS, 437
 System Controls not in SCB (SCnSCB), 68
SCnSCB_ACTLR_DISDEFWBUF_Pos
 CMSIS, 438
 System Controls not in SCB (SCnSCB), 68
SCnSCB_ACTLR_DISFOLD_Msk
 CMSIS, 438
 System Controls not in SCB (SCnSCB), 68

SCnSCB_ACTLR_DISFOLD_Pos
 CMSIS, [438](#)
 System Controls not in SCB (SCnSCB), [68](#)

SCnSCB_ACTLR_DISFPCA_Msk
 CMSIS, [438](#)
 System Controls not in SCB (SCnSCB), [69](#)

SCnSCB_ACTLR_DISFPCA_Pos
 CMSIS, [438](#)
 System Controls not in SCB (SCnSCB), [69](#)

SCnSCB_ACTLR_DISMCYCINT_Msk
 CMSIS, [438](#)
 System Controls not in SCB (SCnSCB), [69](#)

SCnSCB_ACTLR_DISMCYCINT_Pos
 CMSIS, [438](#)
 System Controls not in SCB (SCnSCB), [69](#)

SCnSCB_ACTLR_DISOOFP_Msk
 CMSIS, [438](#)
 System Controls not in SCB (SCnSCB), [69](#)

SCnSCB_ACTLR_DISOOFP_Pos
 CMSIS, [439](#)
 System Controls not in SCB (SCnSCB), [69](#)

SCnSCB_ICTR_INTLINESNUM_Msk
 CMSIS, [439](#)
 System Controls not in SCB (SCnSCB), [69](#)

SCnSCB_ICTR_INTLINESNUM_Pos
 CMSIS, [439](#)
 System Controls not in SCB (SCnSCB), [69](#)

SCnSCB_Type, [1048](#)

SCR
 ITM Functions, [142](#)
 SCB_TypeDef, [1048](#)

SCS_BASE
 CMSIS, [439](#)
 Core Definitions, [110](#)

SDIO_ARG_CMDARG
 Peripheral_Registers_Bits_Definition, [898](#)

SDIO_CLKCR_BYPASS
 Peripheral_Registers_Bits_Definition, [898](#)

SDIO_CLKCR_CLKDIV
 Peripheral_Registers_Bits_Definition, [898](#)

SDIO_CLKCR_CLKEN
 Peripheral_Registers_Bits_Definition, [898](#)

SDIO_CLKCR_HWFC_EN
 Peripheral_Registers_Bits_Definition, [898](#)

SDIO_CLKCR_NEGEDGE
 Peripheral_Registers_Bits_Definition, [898](#)

SDIO_CLKCR_PWRSAV
 Peripheral_Registers_Bits_Definition, [899](#)

SDIO_CLKCR_WIDBUS
 Peripheral_Registers_Bits_Definition, [899](#)

SDIO_CLKCR_WIDBUS_0
 Peripheral_Registers_Bits_Definition, [899](#)

SDIO_CLKCR_WIDBUS_1
 Peripheral_Registers_Bits_Definition, [899](#)

SDIO_CMD_CEATACMD
 Peripheral_Registers_Bits_Definition, [899](#)

SDIO_CMD_CMDINDEX
 Peripheral_Registers_Bits_Definition, [899](#)

SDIO_CMD_CPSMEN
 Peripheral_Registers_Bits_Definition, [899](#)

SDIO_CMD_ENCMDCOMPL
 Peripheral_Registers_Bits_Definition, [900](#)

SDIO_CMD_NIEN
 Peripheral_Registers_Bits_Definition, [900](#)

SDIO_CMD_SDIOSUSPEND
 Peripheral_Registers_Bits_Definition, [900](#)

SDIO_CMD_WAITINT
 Peripheral_Registers_Bits_Definition, [900](#)

SDIO_CMD_WAITPEND
 Peripheral_Registers_Bits_Definition, [900](#)

SDIO_CMD_WAITRESP
 Peripheral_Registers_Bits_Definition, [900](#)

SDIO_CMD_WAITRESP_0
 Peripheral_Registers_Bits_Definition, [900](#)

SDIO_CMD_WAITRESP_1
 Peripheral_Registers_Bits_Definition, [901](#)

SDIO_DCOUNT_DATACOUNT
 Peripheral_Registers_Bits_Definition, [901](#)

SDIO_DCTRL_DBLOCKSIZE
 Peripheral_Registers_Bits_Definition, [901](#)

SDIO_DCTRL_DBLOCKSIZE_0
 Peripheral_Registers_Bits_Definition, [901](#)

SDIO_DCTRL_DBLOCKSIZE_1
 Peripheral_Registers_Bits_Definition, [901](#)

SDIO_DCTRL_DBLOCKSIZE_2
 Peripheral_Registers_Bits_Definition, [901](#)

SDIO_DCTRL_DBLOCKSIZE_3
 Peripheral_Registers_Bits_Definition, [901](#)

SDIO_DCTRL_DMAEN
 Peripheral_Registers_Bits_Definition, [901](#)

SDIO_DCTRL_DTDIR
 Peripheral_Registers_Bits_Definition, [902](#)

SDIO_DCTRL_DTEN
 Peripheral_Registers_Bits_Definition, [902](#)

SDIO_DCTRL_DTMODE
 Peripheral_Registers_Bits_Definition, [902](#)

SDIO_DCTRL_RWMOD
 Peripheral_Registers_Bits_Definition, [902](#)

SDIO_DCTRL_RWSTART
 Peripheral_Registers_Bits_Definition, [902](#)

SDIO_DCTRL_RWSTOP
 Peripheral_Registers_Bits_Definition, [902](#)

SDIO_DCTRL_SDIOEN
 Peripheral_Registers_Bits_Definition, [902](#)

SDIO_DLLEN_DATALENGTH
 Peripheral_Registers_Bits_Definition, [903](#)

SDIO_DTIMER_DATETIME
 Peripheral_Registers_Bits_Definition, [903](#)

SDIO_FIFO_FIFODATA
 Peripheral_Registers_Bits_Definition, [903](#)

SDIO_FIFOCNT_FIFOCOUNT
 Peripheral_Registers_Bits_Definition, [903](#)

SDIO_ICR_CCRCFAILC
 Peripheral_Registers_Bits_Definition, [903](#)

SDIO_ICR_CEATAENDC
 Peripheral_Registers_Bits_Definition, [903](#)

SDIO_ICR_CMDRENDC
 Peripheral_Registers_Bits_Definition, 903
SDIO_ICR_CMDSENTC
 Peripheral_Registers_Bits_Definition, 904
SDIO_ICR_CTIMEOUTC
 Peripheral_Registers_Bits_Definition, 904
SDIO_ICR_DATAENDC
 Peripheral_Registers_Bits_Definition, 904
SDIO_ICR_DBCKENDC
 Peripheral_Registers_Bits_Definition, 904
SDIO_ICR_DCRCFAILC
 Peripheral_Registers_Bits_Definition, 904
SDIO_ICR_DTIMEOUTC
 Peripheral_Registers_Bits_Definition, 904
SDIO_ICR_RXOVERRC
 Peripheral_Registers_Bits_Definition, 904
SDIO_ICR_SDIOITC
 Peripheral_Registers_Bits_Definition, 905
SDIO_ICR_STBITERRC
 Peripheral_Registers_Bits_Definition, 905
SDIO_ICR_TXUNDERRC
 Peripheral_Registers_Bits_Definition, 905
SDIO IRQn
 Configuration_section_for\CMSIS, 526
SDIO_MASK_CCRCFAILIE
 Peripheral_Registers_Bits_Definition, 905
SDIO_MASK_CEATAENDIE
 Peripheral_Registers_Bits_Definition, 905
SDIO_MASK_CMDACTIE
 Peripheral_Registers_Bits_Definition, 905
SDIO_MASK_CMDRENIE
 Peripheral_Registers_Bits_Definition, 905
SDIO_MASK_CMDSENTIE
 Peripheral_Registers_Bits_Definition, 906
SDIO_MASK_CTIMEOUTIE
 Peripheral_Registers_Bits_Definition, 906
SDIO_MASK_DATAENDIE
 Peripheral_Registers_Bits_Definition, 906
SDIO_MASK_DBCKENDIE
 Peripheral_Registers_Bits_Definition, 906
SDIO_MASK_DCRCFAILIE
 Peripheral_Registers_Bits_Definition, 906
SDIO_MASK_DTIMEOUTIE
 Peripheral_Registers_Bits_Definition, 906
SDIO_MASK_RXACTIE
 Peripheral_Registers_Bits_Definition, 906
SDIO_MASK_RXDAVLIE
 Peripheral_Registers_Bits_Definition, 907
SDIO_MASK_RXFIFOEIE
 Peripheral_Registers_Bits_Definition, 907
SDIO_MASK_RXFIFOFIE
 Peripheral_Registers_Bits_Definition, 907
SDIO_MASK_RXFIFOHIE
 Peripheral_Registers_Bits_Definition, 907
SDIO_MASK_RXOVERRIE
 Peripheral_Registers_Bits_Definition, 907
SDIO_MASK_SDIOITIE
 Peripheral_Registers_Bits_Definition, 907
SDIO_MASK_STBITERRIE
 Peripheral_Registers_Bits_Definition, 907
SDIO_MASK_TXACTIE
 Peripheral_Registers_Bits_Definition, 908
SDIO_MASK_TXDAVLIE
 Peripheral_Registers_Bits_Definition, 908
SDIO_MASK_TXFIFOEIE
 Peripheral_Registers_Bits_Definition, 908
SDIO_MASK_TXFIFOFIE
 Peripheral_Registers_Bits_Definition, 908
SDIO_MASK_TXFIFOHEIE
 Peripheral_Registers_Bits_Definition, 908
SDIO_MASK_TXUNDERRIE
 Peripheral_Registers_Bits_Definition, 908
SDIO_POWER_PWRCTRL
 Peripheral_Registers_Bits_Definition, 908
SDIO_POWER_PWRCTRL_0
 Peripheral_Registers_Bits_Definition, 909
SDIO_POWER_PWRCTRL_1
 Peripheral_Registers_Bits_Definition, 909
SDIO_RESP0_CARDSTATUS0
 Peripheral_Registers_Bits_Definition, 909
SDIO_RESP1_CARDSTATUS1
 Peripheral_Registers_Bits_Definition, 909
SDIO_RESP2_CARDSTATUS2
 Peripheral_Registers_Bits_Definition, 909
SDIO_RESP3_CARDSTATUS3
 Peripheral_Registers_Bits_Definition, 909
SDIO_RESP4_CARDSTATUS4
 Peripheral_Registers_Bits_Definition, 909
SDIO_RESPCMD_RESPCMD
 Peripheral_Registers_Bits_Definition, 909
SDIO_STA_CCRCFAIL
 Peripheral_Registers_Bits_Definition, 910
SDIO_STA_CEATAEND
 Peripheral_Registers_Bits_Definition, 910
SDIO_STA_CMDACT
 Peripheral_Registers_Bits_Definition, 910
SDIO_STA_CMDREN
 Peripheral_Registers_Bits_Definition, 910
SDIO_STA_CMDSENT
 Peripheral_Registers_Bits_Definition, 910
SDIO_STA_CTIMEOUT
 Peripheral_Registers_Bits_Definition, 910
SDIO_STA_DATAEND
 Peripheral_Registers_Bits_Definition, 910
SDIO_STA_DBCKEND
 Peripheral_Registers_Bits_Definition, 911
SDIO_STA_DCRCFAIL
 Peripheral_Registers_Bits_Definition, 911
SDIO_STA_DTIMEOUT
 Peripheral_Registers_Bits_Definition, 911
SDIO_STA_RXACT
 Peripheral_Registers_Bits_Definition, 911
SDIO_STA_RXDAVL
 Peripheral_Registers_Bits_Definition, 911
SDIO_STA_RXFIFOE
 Peripheral_Registers_Bits_Definition, 911

SDIO_STA_RXFIFOF
 Peripheral_Registers_Bits_Definition, 911

SDIO_STA_RXFIFOHF
 Peripheral_Registers_Bits_Definition, 912

SDIO_STA_RXOVERR
 Peripheral_Registers_Bits_Definition, 912

SDIO_STA_SDIOIT
 Peripheral_Registers_Bits_Definition, 912

SDIO_STA_STBITERR
 Peripheral_Registers_Bits_Definition, 912

SDIO_STA_TXACT
 Peripheral_Registers_Bits_Definition, 912

SDIO_STA_TXDAVL
 Peripheral_Registers_Bits_Definition, 912

SDIO_STA_TXFIFOE
 Peripheral_Registers_Bits_Definition, 912

SDIO_STA_TXFIFOFOF
 Peripheral_Registers_Bits_Definition, 913

SDIO_STA_TXFIFOHE
 Peripheral_Registers_Bits_Definition, 913

SDIO_STA_TXUNDERR
 Peripheral_Registers_Bits_Definition, 913

SDIO_TypeDef, 1049

SecondaryAddress
 S_I2C_Slave_address, 1041

SERVO MOTOR Driver, 37

Servo1_Entry_Gate
 API Function Prototypes, 39

Servo1_Entry_Gate_Init
 API Function Prototypes, 39

Servo2_Exit_Gate
 API Function Prototypes, 40

Servo2_Exit_Gate_Init
 API Function Prototypes, 41

SET
 Platform Standard Types, 152

Set
 Exported_types, 530
 I2C Driver, 269

sFIFOMailBox
 CMSIS, 512

sFilterRegister
 CMSIS, 512

SHCSR
 ITM Functions, 142
 SCB_TypeDef, 1048

SHIFTR
 CMSIS, 512

SHP
 ITM Functions, 142
 SCB_TypeDef, 1048

sint16
 Platform Standard Types, 153

sint16_least
 Platform Standard Types, 153

sint32
 Platform Standard Types, 153

sint32_least

Platform Standard Types, 153

sint64
 Platform Standard Types, 153

sint8
 Platform Standard Types, 153

sint8_least
 Platform Standard Types, 154

SIZE1
 Global_Variables_Definitions, 370

Slave_State
 I2C Driver, 270

SLEEPCNT
 ITM Functions, 142

SLOTR
 CMSIS, 513

SMCR
 CMSIS, 513
 TIM1_TypeDef, 1059

SMPR1
 CMSIS, 513

SMPR2
 CMSIS, 513

SPI Clock Phase Definitions, 321
 SPI_Clock_Phase_Leading, 321
 SPI_Clock_Phase_Trailing, 321

SPI Clock Polarity Definitions, 320
 SPI_Clock_Polarity_High_Idle, 320
 SPI_Clock_Polarity_Low_Idle, 320

SPI Communication Mode Definitions, 317
 SPI_Direction_1line_Receive_only, 318
 SPI_Direction_1line_Transmit_only, 318
 SPI_Direction_2lines, 318
 SPI_Direction_2lines_RX_Only, 318

SPI Driver, 315
 Disabled, 316
 Enabled, 316
 SPI_PollingMechanism, 316

SPI Frame Format Definitions, 319
 SPI_Frame_Format_LSB, 319
 SPI_Frame_Format_MSB, 320

SPI Instances, 176
 SPI1, 177
 SPI2, 177

SPI IRQ Enable Definitions, 325
 SPI_IRQ_Enable_Err, 325
 SPI_IRQ_Enable_None, 325
 SPI_IRQ_Enable_Rx_Only, 325
 SPI_IRQ_Enable_Tx_Only, 325

SPI Mode Definitions, 317
 SPI_Mode_Master, 317
 SPI_Mode_Slave, 317

SPI NSS Definitions, 322
 SPI_NSS_Hard_Master_SS_No_output, 322
 SPI_NSS_Hard_Master_SS_Output_Enable, 322
 SPI_NSS_Hard_Slave, 322
 SPI_NSS_Internal_Soft_Reset, 322
 SPI_NSS_Internal_Soft_Set, 323

SPI Payload Length Definitions, 318

SPI_Payload_Length_16bit, 319
 SPI_Payload_Length_8bit, 319
SPI Prescaler Definitions, 323
 SPI_Prescaler_By128, 323
 SPI_Prescaler_By16, 323
 SPI_Prescaler_By2, 324
 SPI_Prescaler_By256, 324
 SPI_Prescaler_By32, 324
 SPI_Prescaler_By4, 324
 SPI_Prescaler_By64, 324
 SPI_Prescaler_By8, 324
SPI Register Map, 165
SPI1
 SPI Instances, 177
SPI1_Index
 stm32f401xx_spi_driver.c, 1222
SPI1_IRQHandler
 stm32f401xx_spi_driver.c, 1222
SPI1 IRQn
 Configuration_section_for\CMSIS, 526
SPI2
 SPI Instances, 177
SPI2_Index
 stm32f401xx_spi_driver.c, 1222
SPI2_IRQHandler
 stm32f401xx_spi_driver.c, 1223
SPI2 IRQn
 Configuration_section_for\CMSIS, 526
SPI3_IRQHandler
 Configuration_section_for\CMSIS, 526
SPI4_IRQHandler
 Configuration_section_for\CMSIS, 526
SPI_Clock_Phase_Leading
 SPI Clock Phase Definitions, 321
SPI_Clock_Phase_Trailing
 SPI Clock Phase Definitions, 321
SPI_Clock_Polarity_High_Idle
 SPI Clock Polarity Definitions, 320
SPI_Clock_Polarity_Low_Idle
 SPI Clock Polarity Definitions, 320
SPI_CR1_BIDIMODE
 Peripheral_Registers_Bits_Definition, 913
SPI_CR1_BIDIOE
 Peripheral_Registers_Bits_Definition, 913
SPI_CR1_BR
 Peripheral_Registers_Bits_Definition, 913
SPI_CR1_BR_0
 Peripheral_Registers_Bits_Definition, 913
SPI_CR1_BR_1
 Peripheral_Registers_Bits_Definition, 914
SPI_CR1_BR_2
 Peripheral_Registers_Bits_Definition, 914
SPI_CR1_CPHA
 Peripheral_Registers_Bits_Definition, 914
SPI_CR1_CPOL
 Peripheral_Registers_Bits_Definition, 914
SPI_CR1_CRCEN
 Peripheral_Registers_Bits_Definition, 914
SPI_CR1_CRCNEXT
 Peripheral_Registers_Bits_Definition, 914
SPI_CR1_DFF
 Peripheral_Registers_Bits_Definition, 914
SPI_CR1_LSBFIRST
 Peripheral_Registers_Bits_Definition, 915
SPI_CR1_MSTR
 Peripheral_Registers_Bits_Definition, 915
SPI_CR1_RXONLY
 Peripheral_Registers_Bits_Definition, 915
SPI_CR1_SPE
 Peripheral_Registers_Bits_Definition, 915
SPI_CR1_SSI
 Peripheral_Registers_Bits_Definition, 915
SPI_CR1_SSM
 Peripheral_Registers_Bits_Definition, 915
SPI_CR2_ERRIE
 Peripheral_Registers_Bits_Definition, 915
SPI_CR2_RXDMAEN
 Peripheral_Registers_Bits_Definition, 916
SPI_CR2_RXNEIE
 Peripheral_Registers_Bits_Definition, 916
SPI_CR2_SSOE
 Peripheral_Registers_Bits_Definition, 916
SPI_CR2_TXDMAEN
 Peripheral_Registers_Bits_Definition, 916
SPI_CR2_TXEIE
 Peripheral_Registers_Bits_Definition, 916
SPI_CRCPR_CRCPOLY
 Peripheral_Registers_Bits_Definition, 916
SPI_Direction_1line_Receive_only
 SPI Communication Mode Definitions, 318
SPI_Direction_1line_Transmit_only
 SPI Communication Mode Definitions, 318
SPI_Direction_2lines
 SPI Communication Mode Definitions, 318
SPI_Direction_2lines_RX_Only
 SPI Communication Mode Definitions, 318
SPI_DR_DR
 Peripheral_Registers_Bits_Definition, 916
SPI_Frame_Format_LSB
 SPI Frame Format Definitions, 319
SPI_Frame_Format_MSB
 SPI Frame Format Definitions, 320
SPI_I2SCFGR_CHLEN
 Peripheral_Registers_Bits_Definition, 917
SPI_I2SCFGR_CKPOL
 Peripheral_Registers_Bits_Definition, 917
SPI_I2SCFGR_DATLEN
 Peripheral_Registers_Bits_Definition, 917
SPI_I2SCFGR_DATLEN_0
 Peripheral_Registers_Bits_Definition, 917
SPI_I2SCFGR_DATLEN_1
 Peripheral_Registers_Bits_Definition, 917
SPI_I2SCFGR_I2SCFG
 Peripheral_Registers_Bits_Definition, 917
SPI_I2SCFGR_I2SCFG_0
 Peripheral_Registers_Bits_Definition, 917

SPI_I2SCFGR_I2SCFG_1
 Peripheral_Registers_Bits_Definition, 918

SPI_I2SCFGR_I2SE
 Peripheral_Registers_Bits_Definition, 918

SPI_I2SCFGR_I2SMOD
 Peripheral_Registers_Bits_Definition, 918

SPI_I2SCFGR_I2SSTD
 Peripheral_Registers_Bits_Definition, 918

SPI_I2SCFGR_I2SSTD_0
 Peripheral_Registers_Bits_Definition, 918

SPI_I2SCFGR_I2SSTD_1
 Peripheral_Registers_Bits_Definition, 918

SPI_I2SCFGR_PCMSYNC
 Peripheral_Registers_Bits_Definition, 918

SPI_I2SPR_I2SDIV
 Peripheral_Registers_Bits_Definition, 919

SPI_I2SPR_MCKOE
 Peripheral_Registers_Bits_Definition, 919

SPI_I2SPR_ODD
 Peripheral_Registers_Bits_Definition, 919

SPI_IRQ_Enable_Err
 SPI IRQ Enable Definitions, 325

SPI_IRQ_Enable_None
 SPI IRQ Enable Definitions, 325

SPI_IRQ_Enable_Rx_Only
 SPI IRQ Enable Definitions, 325

SPI_IRQ_Enable_Tx_Only
 SPI IRQ Enable Definitions, 325

SPI_Mode_Master
 SPI Mode Definitions, 317

SPI_Mode_Slave
 SPI Mode Definitions, 317

SPI_NSS_Hard_Master_SS_No_output
 SPI NSS Definitions, 322

SPI_NSS_Hard_Master_SS_Output_Enable
 SPI NSS Definitions, 322

SPI_NSS_Hard_Slave
 SPI NSS Definitions, 322

SPI_NSS_Internal_Soft_Reset
 SPI NSS Definitions, 322

SPI_NSS_Internal_Soft_Set
 SPI NSS Definitions, 323

SPI_Payload_Length_16bit
 SPI Payload Length Definitions, 319

SPI_Payload_Length_8bit
 SPI Payload Length Definitions, 319

SPI_PollingMechanism
 SPI Driver, 316

SPI_Prescaler
 S_SPI_Config_t, 1044

SPI_Prescaler_By128
 SPI Prescaler Definitions, 323

SPI_Prescaler_By16
 SPI Prescaler Definitions, 323

SPI_Prescaler_By2
 SPI Prescaler Definitions, 324

SPI_Prescaler_By256
 SPI Prescaler Definitions, 324

SPI_Prescaler_By32
 SPI Prescaler Definitions, 324

SPI_Prescaler_By4
 SPI Prescaler Definitions, 324

SPI_Prescaler_By64
 SPI Prescaler Definitions, 324

SPI_Prescaler_By8
 SPI Prescaler Definitions, 324

SPI_RXCRCR_RXCRC
 Peripheral_Registers_Bits_Definition, 919

SPI_SR_BSY
 Peripheral_Registers_Bits_Definition, 919

SPI_SR_CHSIDE
 Peripheral_Registers_Bits_Definition, 919

SPI_SR_CRCERR
 Peripheral_Registers_Bits_Definition, 919

SPI_SR_MODF
 Peripheral_Registers_Bits_Definition, 920

SPI_SR_OVR
 Peripheral_Registers_Bits_Definition, 920

SPI_SR_RXNE
 Peripheral_Registers_Bits_Definition, 920
 stm32f401xx_spi_driver.c, 1222

SPI_SR_TXE
 Peripheral_Registers_Bits_Definition, 920
 stm32f401xx_spi_driver.c, 1222

SPI_SR_UDR
 Peripheral_Registers_Bits_Definition, 920

SPI_TXCRCR_TXCRC
 Peripheral_Registers_Bits_Definition, 920

SPI_TypeDef, 1050

 CR1, 1050

 CR2, 1050

 CRCPR, 1051

 DR, 1051

 I2SCFGR, 1051

 I2SPR, 1051

 RXCRCR, 1051

 SR, 1051

 TXCRCR, 1051

SPPR
 ITM Functions, 143

SPSEL
 ITM Functions, 143

SQR1
 CMSIS, 513

SQR2
 CMSIS, 513

SQR3
 CMSIS, 513

SR
 CMSIS, 513–515

 CRC_TypeDef, 1005

 SPI_TypeDef, 1051

 TIM1_TypeDef, 1059

 USART_TypeDef, 1063

SR1
 CMSIS, 515

I2C_TypeDef, 1020
SR2
 CMSIS, 515
 I2C_TypeDef, 1021
SRAM1_BASE
 Peripheral_memory_map, 535
SRAM1_BB_BASE
 Peripheral_memory_map, 536
SRAM_BB_BASE
 Peripheral_memory_map, 536
Src/app_states.c, 1228
Src/ecu.c, 1229
Src/sysmem.c, 1231
Src/system_stm32f4xx.c, 1233
SRCR
 CMSIS, 515
SSCGR
 CMSIS, 516
 RCC_TypeDef, 1037
SSCR
 CMSIS, 516
SSPSR
 ITM Functions, 143
SSR
 CMSIS, 516
STA
 CMSIS, 516
Start
 I2C Driver, 270
State Handlers, 357
 STATE_API, 358
State Macros, 355
 STATE_API, 355
 STATE_NAME, 355
STATE_API
 Application States, 352, 353
 State Handlers, 358
 State Macros, 355
STATE_NAME
 State Macros, 355
STATES
 Application States, 352
Status
 I2C Driver, 271
Status and Control Registers, 43
 APSR_C_Msk, 44
 APSR_C_Pos, 44
 APSR_GE_Msk, 44
 APSR_GE_Pos, 44
 APSR_N_Msk, 45
 APSR_N_Pos, 45
 APSR_Q_Msk, 45
 APSR_Q_Pos, 45
 APSR_V_Msk, 45
 APSR_V_Pos, 45
 APSR_Z_Msk, 45
 APSR_Z_Pos, 45
 CONTROL_FPCA_Msk, 46
 CONTROL_FPCA_Pos, 46
 CONTROL_nPRIV_Msk, 46
 CONTROL_nPRIV_Pos, 46
 CONTROL_SPSEL_Msk, 46
 CONTROL_SPSEL_Pos, 46
 IPSR_ISR_Msk, 46
 IPSR_ISR_Pos, 46
 xPSR_C_Msk, 47
 xPSR_C_Pos, 47
 xPSR_GE_Msk, 47
 xPSR_GE_Pos, 47
 xPSR_ISR_Msk, 47
 xPSR_ISR_Pos, 47
 xPSR_IT_Msk, 47
 xPSR_IT_Pos, 47
 xPSR_N_Msk, 48
 xPSR_N_Pos, 48
 xPSR_Q_Msk, 48
 xPSR_Q_Pos, 48
 xPSR_T_Msk, 48
 xPSR_T_Pos, 48
 xPSR_V_Msk, 48
 xPSR_V_Pos, 48
 xPSR_Z_Msk, 49
 xPSR_Z_Pos, 49
STIR
 ITM Functions, 143
 NVIC_TypeDef, 1031
STK
 SysTick Timer Instance, 170
STK_CLK_AHB
 SysTick Clock Configuration, 333
STK_CLK_AHB_8
 SysTick Clock Configuration, 333
STK_config_t, 1052
 Callback_Function, 1052
 clock_config, 1052
 interrupt_config, 1052
 reload_value, 1053
 running_mode, 1053
STK_FCPU
 SYSTICK Driver, 332
STK_INTERRUPT_DISABLED
 SysTick Interrupt Configuration, 332
STK_INTERRUPT_ENABLED
 SysTick Interrupt Configuration, 332
STK_ONE_SHOT_MODE
 SysTick Running Mode Configuration, 334
STK_PERIODIC_MODE
 SysTick Running Mode Configuration, 334
STK_TypeDef, 1053
 CALIB, 1053
 CTRL, 1053
 LOAD, 1053
 VAL, 1054
STM32F401xx MCU Header File, 156
stm32f401xx_i2c_driver.c
 Global_I2C_Config, 1220

I2C1_Index, 1220
I2C2_Index, 1220
stm32f401xx_spi_driver.c
 Global_SPI_Config, 1223
 SPI1_Index, 1222
 SPI1_IRQHandler, 1222
 SPI2_Index, 1222
 SPI2_IRQHandler, 1223
 SPI_SR_RXNE, 1222
 SPI_SR_TXE, 1222
Stm32f4xx, 519
Stm32f4xx_system, 990
STM32F4xx_System_Exported_Constants, 992
STM32F4xx_System_Exported_Functions, 992
 SystemCoreClockUpdate, 993
 SystemInit, 994
STM32F4xx_System_Exported_Macros, 992
STM32F4xx_System_Exported_types, 991
 SystemCoreClock, 992
STM32F4xx_System_INCLUDES, 991
STM32F4xx_System_Private_Defines, 995
 VECT_TAB_OFFSET, 995
STM32F4xx_System_Private_FunctionPrototypes, 996
STM32F4xx_System_Private_Functions, 996
 SystemCoreClockUpdate, 997
 SystemInit, 998
STM32F4xx_System_Private_INCLUDES, 994
STM32F4xx_System_Private_MACROS, 995
STM32F4xx_System_Private_TYPESDefinitions, 994
STM32F4xx_System_Private_Variables, 996
StopCondition
 I2C_Driver, 271
STR
 CMSIS, 516
Structure for configuring an LED., 33
sTxMailBox
 CMSIS, 516
SVCALL IRQn
 Configuration_section_for_CMSIS, 524
SWIER
 CMSIS, 516
 EXTI_TypeDef, 1014
SWTRIGR
 CMSIS, 517
SYSCFG
 SYSCFG_Instance, 174
SYSCFG_Instance, 174
 SYSCFG, 174
SYSCFG_Register_Map, 164
SYSCFG_CMPPCR_CMP_PD
 Peripheral_Registers_Bits_Definition, 920
SYSCFG_CMPPCR_READY
 Peripheral_Registers_Bits_Definition, 921
SYSCFG_EXTICR1_EXTI0
 Peripheral_Registers_Bits_Definition, 921
SYSCFG_EXTICR1_EXTI0_PA
 Peripheral_Registers_Bits_Definition, 921
SYSCFG_EXTICR1_EXTI0_PB
 Peripheral_Registers_Bits_Definition, 921
SYSCFG_EXTICR1_EXTI0_PC
 Peripheral_Registers_Bits_Definition, 921
SYSCFG_EXTICR1_EXTI0_PD
 Peripheral_Registers_Bits_Definition, 921
SYSCFG_EXTICR1_EXTI0_PE
 Peripheral_Registers_Bits_Definition, 921
SYSCFG_EXTICR1_EXTI0_PF
 Peripheral_Registers_Bits_Definition, 922
SYSCFG_EXTICR1_EXTI0_PG
 Peripheral_Registers_Bits_Definition, 922
SYSCFG_EXTICR1_EXTI0_PH
 Peripheral_Registers_Bits_Definition, 922
SYSCFG_EXTICR1_EXTI0_PI
 Peripheral_Registers_Bits_Definition, 922
SYSCFG_EXTICR1_EXTI0_PJ
 Peripheral_Registers_Bits_Definition, 922
SYSCFG_EXTICR1_EXTI0_PK
 Peripheral_Registers_Bits_Definition, 922
SYSCFG_EXTICR1_EXTI1
 Peripheral_Registers_Bits_Definition, 922
SYSCFG_EXTICR1_EXTI1_PA
 Peripheral_Registers_Bits_Definition, 922
SYSCFG_EXTICR1_EXTI1_PB
 Peripheral_Registers_Bits_Definition, 923
SYSCFG_EXTICR1_EXTI1_PC
 Peripheral_Registers_Bits_Definition, 923
SYSCFG_EXTICR1_EXTI1_PD
 Peripheral_Registers_Bits_Definition, 923
SYSCFG_EXTICR1_EXTI1_PE
 Peripheral_Registers_Bits_Definition, 923
SYSCFG_EXTICR1_EXTI1_PF
 Peripheral_Registers_Bits_Definition, 923
SYSCFG_EXTICR1_EXTI1_PG
 Peripheral_Registers_Bits_Definition, 923
SYSCFG_EXTICR1_EXTI1_PH
 Peripheral_Registers_Bits_Definition, 923
SYSCFG_EXTICR1_EXTI1_PI
 Peripheral_Registers_Bits_Definition, 924
SYSCFG_EXTICR1_EXTI1_PJ
 Peripheral_Registers_Bits_Definition, 924
SYSCFG_EXTICR1_EXTI1_PK
 Peripheral_Registers_Bits_Definition, 924
SYSCFG_EXTICR1_EXTI2
 Peripheral_Registers_Bits_Definition, 924
SYSCFG_EXTICR1_EXTI2_PA
 Peripheral_Registers_Bits_Definition, 924
SYSCFG_EXTICR1_EXTI2_PB
 Peripheral_Registers_Bits_Definition, 924
SYSCFG_EXTICR1_EXTI2_PC
 Peripheral_Registers_Bits_Definition, 924
SYSCFG_EXTICR1_EXTI2_PD
 Peripheral_Registers_Bits_Definition, 925
SYSCFG_EXTICR1_EXTI2_PE
 Peripheral_Registers_Bits_Definition, 925
SYSCFG_EXTICR1_EXTI2_PF
 Peripheral_Registers_Bits_Definition, 925

- Peripheral_Registers_Bits_Definition, [941](#)
- SYSCFG_EXTICR4_EXTI12_PH**
 - Peripheral_Registers_Bits_Definition, [941](#)
- SYSCFG_EXTICR4_EXTI12_PI**
 - Peripheral_Registers_Bits_Definition, [941](#)
- SYSCFG_EXTICR4_EXTI12_PJ**
 - Peripheral_Registers_Bits_Definition, [941](#)
- SYSCFG_EXTICR4_EXTI13**
 - Peripheral_Registers_Bits_Definition, [941](#)
- SYSCFG_EXTICR4_EXTI13_PA**
 - Peripheral_Registers_Bits_Definition, [942](#)
- SYSCFG_EXTICR4_EXTI13_PB**
 - Peripheral_Registers_Bits_Definition, [942](#)
- SYSCFG_EXTICR4_EXTI13_PC**
 - Peripheral_Registers_Bits_Definition, [942](#)
- SYSCFG_EXTICR4_EXTI13_PD**
 - Peripheral_Registers_Bits_Definition, [942](#)
- SYSCFG_EXTICR4_EXTI13_PE**
 - Peripheral_Registers_Bits_Definition, [942](#)
- SYSCFG_EXTICR4_EXTI13_PF**
 - Peripheral_Registers_Bits_Definition, [942](#)
- SYSCFG_EXTICR4_EXTI13_PH**
 - Peripheral_Registers_Bits_Definition, [943](#)
- SYSCFG_EXTICR4_EXTI13_PI**
 - Peripheral_Registers_Bits_Definition, [943](#)
- SYSCFG_EXTICR4_EXTI13_PJ**
 - Peripheral_Registers_Bits_Definition, [943](#)
- SYSCFG_EXTICR4_EXTI14**
 - Peripheral_Registers_Bits_Definition, [943](#)
- SYSCFG_EXTICR4_EXTI14_PA**
 - Peripheral_Registers_Bits_Definition, [943](#)
- SYSCFG_EXTICR4_EXTI14_PB**
 - Peripheral_Registers_Bits_Definition, [943](#)
- SYSCFG_EXTICR4_EXTI14_PC**
 - Peripheral_Registers_Bits_Definition, [943](#)
- SYSCFG_EXTICR4_EXTI14_PD**
 - Peripheral_Registers_Bits_Definition, [944](#)
- SYSCFG_EXTICR4_EXTI14_PE**
 - Peripheral_Registers_Bits_Definition, [944](#)
- SYSCFG_EXTICR4_EXTI14_PF**
 - Peripheral_Registers_Bits_Definition, [944](#)
- SYSCFG_EXTICR4_EXTI14_PG**
 - Peripheral_Registers_Bits_Definition, [944](#)
- SYSCFG_EXTICR4_EXTI14_PH**
 - Peripheral_Registers_Bits_Definition, [944](#)
- SYSCFG_EXTICR4_EXTI14_PI**
 - Peripheral_Registers_Bits_Definition, [944](#)
- SYSCFG_EXTICR4_EXTI14_PJ**
 - Peripheral_Registers_Bits_Definition, [944](#)
- SYSCFG_EXTICR4_EXTI15**
 - Peripheral_Registers_Bits_Definition, [944](#)
- SYSCFG_EXTICR4_EXTI15_PA**
 - Peripheral_Registers_Bits_Definition, [945](#)
- SYSCFG_EXTICR4_EXTI15_PB**
 - Peripheral_Registers_Bits_Definition, [945](#)
- SYSCFG_EXTICR4_EXTI15_PC**
 - Peripheral_Registers_Bits_Definition, [945](#)
- SYSCFG_EXTICR4_EXTI15_PD**
 - Peripheral_Registers_Bits_Definition, [945](#)
- SYSCFG_EXTICR4_EXTI15_PE**
 - Peripheral_Registers_Bits_Definition, [945](#)
- SYSCFG_EXTICR4_EXTI15_PF**
 - Peripheral_Registers_Bits_Definition, [945](#)
- SYSCFG_EXTICR4_EXTI15_PG**
 - Peripheral_Registers_Bits_Definition, [945](#)
- SYSCFG_EXTICR4_EXTI15_PH**
 - Peripheral_Registers_Bits_Definition, [946](#)
- SYSCFG_EXTICR4_EXTI15_PI**
 - Peripheral_Registers_Bits_Definition, [946](#)
- SYSCFG_EXTICR4_EXTI15_PJ**
 - Peripheral_Registers_Bits_Definition, [946](#)
- SYSCFG_MEMRMP_FB_MODE**
 - Peripheral_Registers_Bits_Definition, [946](#)
- SYSCFG_MEMRMP_MEM_MODE**
 - Peripheral_Registers_Bits_Definition, [946](#)
- SYSCFG_MEMRMP_MEM_MODE_0**
 - Peripheral_Registers_Bits_Definition, [946](#)
- SYSCFG_MEMRMP_MEM_MODE_1**
 - Peripheral_Registers_Bits_Definition, [946](#)
- SYSCFG_MEMRMP_MEM_MODE_2**
 - Peripheral_Registers_Bits_Definition, [946](#)
- SYSCFG_MEMRMP_SWP_FMC**
 - Peripheral_Registers_Bits_Definition, [947](#)
- SYSCFG_MEMRMP_SWP_FMC_0**
 - Peripheral_Registers_Bits_Definition, [947](#)
- SYSCFG_MEMRMP_SWP_FMC_1**
 - Peripheral_Registers_Bits_Definition, [947](#)
- SYSCFG_PMC_ADC1DC2**
 - Peripheral_Registers_Bits_Definition, [947](#)
- SYSCFG_PMC_ADC2DC2**
 - Peripheral_Registers_Bits_Definition, [947](#)
- SYSCFG_PMC_ADC3DC2**
 - Peripheral_Registers_Bits_Definition, [947](#)
- SYSCFG_PMC_ADCxCDC2**
 - Peripheral_Registers_Bits_Definition, [947](#)
- SYSCFG_PMC_MII_RMII_SEL**
 - Peripheral_Registers_Bits_Definition, [948](#)
- SYSCFG_RegDef_t**, [1054](#)
 - CFGR, [1054](#)
 - CMPCR, [1054](#)
 - EXTICR, [1054](#)
 - MEMRMP, [1055](#)
 - PMC, [1055](#)
 - RESERVED1, [1055](#)
 - RESERVED2, [1055](#)
- SYSCFG_TypeDef**, [1055](#)
- sysmem.c**
 - _sbrk, [1232](#)
- System Control Block (SCB)**, [50](#)
 - SCB_AIRCR_ENDIANESS_Msk, [53](#)
 - SCB_AIRCR_ENDIANESS_Pos, [53](#)
 - SCB_AIRCR_PRIGROUP_Msk, [53](#)
 - SCB_AIRCR_PRIGROUP_Pos, [53](#)
 - SCB_AIRCR_SYSRESETREQ_Msk, [53](#)

SCB_AIRCR_SYSRESETREQ_Pos, 54
SCB_AIRCR_VECTCLRACTIVE_Msk, 54
SCB_AIRCR_VECTCLRACTIVE_Pos, 54
SCB_AIRCR_VECTKEY_Msk, 54
SCB_AIRCR_VECTKEY_Pos, 54
SCB_AIRCR_VECTKEYSTAT_Msk, 54
SCB_AIRCR_VECTKEYSTAT_Pos, 54
SCB_AIRCR_VECTRESET_Msk, 54
SCB_AIRCR_VECTRESET_Pos, 55
SCB_CCR_BFHFNMIGN_Msk, 55
SCB_CCR_BFHFNMIGN_Pos, 55
SCB_CCR_DIV_0_TRP_Msk, 55
SCB_CCR_DIV_0_TRP_Pos, 55
SCB_CCR_NONBASETHRDENA_Msk, 55
SCB_CCR_NONBASETHRDENA_Pos, 55
SCB_CCR_STKALIGN_Msk, 55
SCB_CCR_STKALIGN_Pos, 56
SCB_CCR_UNALIGN_TRP_Msk, 56
SCB_CCR_UNALIGN_TRP_Pos, 56
SCB_CCR_USERSETMPEND_Msk, 56
SCB_CCR_USERSETMPEND_Pos, 56
SCB_CFSR_BUSFAULTSR_Msk, 56
SCB_CFSR_BUSFAULTSR_Pos, 56
SCB_CFSR_MEMFAULTSR_Msk, 56
SCB_CFSR_MEMFAULTSR_Pos, 57
SCB_CFSR_USGFAULTSR_Msk, 57
SCB_CFSR_USGFAULTSR_Pos, 57
SCB_CPUID_ARCHITECTURE_Msk, 57
SCB_CPUID_ARCHITECTURE_Pos, 57
SCB_CPUID_IMPLEMENTER_Msk, 57
SCB_CPUID_IMPLEMENTER_Pos, 57
SCB_CPUID_PARTNO_Msk, 57
SCB_CPUID_PARTNO_Pos, 58
SCB_CPUID_REVISION_Msk, 58
SCB_CPUID_REVISION_Pos, 58
SCB_CPUID_VARIANT_Msk, 58
SCB_CPUID_VARIANT_Pos, 58
SCB_DFSR_BKPT_Msk, 58
SCB_DFSR_BKPT_Pos, 58
SCB_DFSR_DWTTRAP_Msk, 58
SCB_DFSR_DWTTRAP_Pos, 59
SCB_DFSR_EXTERNAL_Msk, 59
SCB_DFSR_EXTERNAL_Pos, 59
SCB_DFSR_HALTED_Msk, 59
SCB_DFSR_HALTED_Pos, 59
SCB_DFSR_VCATCH_Msk, 59
SCB_DFSR_VCATCH_Pos, 59
SCB_HFSR_DEBUGEVT_Msk, 59
SCB_HFSR_DEBUGEVT_Pos, 60
SCB_HFSR_FORCED_Msk, 60
SCB_HFSR_FORCED_Pos, 60
SCB_HFSR_VECTTBL_Msk, 60
SCB_HFSR_VECTTBL_Pos, 60
SCB_ICSR_ISRPENDING_Msk, 60
SCB_ICSR_ISRPENDING_Pos, 60
SCB_ICSR_ISRPREEMPT_Msk, 60
SCB_ICSR_ISRPREEMPT_Pos, 61
SCB_ICSR_NMIPENDSET_Msk, 61
SCB_ICSR_NMIPENDSET_Pos, 61
SCB_ICSR_PENDSTCLR_Msk, 61
SCB_ICSR_PENDSTCLR_Pos, 61
SCB_ICSR_PENDSTSET_Msk, 61
SCB_ICSR_PENDSTSET_Pos, 61
SCB_ICSR_PENDSVCLR_Msk, 61
SCB_ICSR_PENDSVCLR_Pos, 62
SCB_ICSR_PENDSVSET_Msk, 62
SCB_ICSR_PENDSVSET_Pos, 62
SCB_ICSR_RETTOBASE_Msk, 62
SCB_ICSR_RETTOBASE_Pos, 62
SCB_ICSR_VECTACTIVE_Msk, 62
SCB_ICSR_VECTACTIVE_Pos, 62
SCB_ICSR_VECTPENDING_Msk, 62
SCB_ICSR_VECTPENDING_Pos, 63
SCB_SCR_SEVONPEND_Msk, 63
SCB_SCR_SEVONPEND_Pos, 63
SCB_SCR_SLEEPDEEP_Msk, 63
SCB_SCR_SLEEPDEEP_Pos, 63
SCB_SCR_SLEEPONEEXIT_Msk, 63
SCB_SCR_SLEEPONEEXIT_Pos, 63
SCB_SHCSR_BUSFAULTACT_Msk, 63
SCB_SHCSR_BUSFAULTACT_Pos, 64
SCB_SHCSR_BUSFAULTENA_Msk, 64
SCB_SHCSR_BUSFAULTENA_Pos, 64
SCB_SHCSR_BUSFAULTPENDED_Msk, 64
SCB_SHCSR_BUSFAULTPENDED_Pos, 64
SCB_SHCSR_MEMFAULTACT_Msk, 64
SCB_SHCSR_MEMFAULTACT_Pos, 64
SCB_SHCSR_MEMFAULTENA_Msk, 64
SCB_SHCSR_MEMFAULTENA_Pos, 65
SCB_SHCSR_MEMFAULTPENDED_Msk, 65
SCB_SHCSR_MEMFAULTPENDED_Pos, 65
SCB_SHCSR_MONITORACT_Msk, 65
SCB_SHCSR_MONITORACT_Pos, 65
SCB_SHCSR_PENDSVACT_Msk, 65
SCB_SHCSR_PENDSVACT_Pos, 65
SCB_SHCSR_SVCALLACT_Msk, 65
SCB_SHCSR_SVCALLACT_Pos, 66
SCB_SHCSR_SVCALLPENDED_Msk, 66
SCB_SHCSR_SVCALLPENDED_Pos, 66
SCB_SHCSR_SYSTICKACT_Msk, 66
SCB_SHCSR_SYSTICKACT_Pos, 66
SCB_SHCSR_USGFAULTACT_Msk, 66
SCB_SHCSR_USGFAULTACT_Pos, 66
SCB_SHCSR_USGFAULTENA_Msk, 66
SCB_SHCSR_USGFAULTENA_Pos, 67
SCB_SHCSR_USGFAULTPENDED_Msk, 67
SCB_SHCSR_USGFAULTPENDED_Pos, 67
SCB_VTOR_TBLOFF_Msk, 67
SCB_VTOR_TBLOFF_Pos, 67
System Controls not in SCB (SCnSCB), 67
SCnSCB_ACTLR_DISDEFWBUF_Msk, 68
SCnSCB_ACTLR_DISDEFWBUF_Pos, 68
SCnSCB_ACTLR_DISFOLD_Msk, 68
SCnSCB_ACTLR_DISFOLD_Pos, 68
SCnSCB_ACTLR_DISFPCA_Msk, 69
SCnSCB_ACTLR_DISFPCA_Pos, 69

SCnSCB_ACTLR_DISMCYCINT_Msk, 69
 SCnSCB_ACTLR_DISMCYCINT_Pos, 69
 SCnSCB_ACTLR_DISOOFP_Msk, 69
 SCnSCB_ACTLR_DISOOFP_Pos, 69
 SCnSCB_ICTR_INTLINESNUM_Msk, 69
 SCnSCB_ICTR_INTLINESNUM_Pos, 69

System Tick Timer (SysTick), 70
 SysTick_CALIB_NOREF_Msk, 71
 SysTick_CALIB_NOREF_Pos, 71
 SysTick_CALIB_SKEW_Msk, 71
 SysTick_CALIB_SKEW_Pos, 71
 SysTick_CALIB_TENMS_Msk, 71
 SysTick_CALIB_TENMS_Pos, 71
 SysTick_CTRL_CLKSOURCE_Msk, 71
 SysTick_CTRL_CLKSOURCE_Pos, 72
 SysTick_CTRL_COUNTFLAG_Msk, 72
 SysTick_CTRL_COUNTFLAG_Pos, 72
 SysTick_CTRL_ENABLE_Msk, 72
 SysTick_CTRL_ENABLE_Pos, 72
 SysTick_CTRL_TICKINT_Msk, 72
 SysTick_CTRL_TICKINT_Pos, 72
 SysTick_LOAD_RELOAD_Msk, 72
 SysTick_LOAD_RELOAD_Pos, 73
 SysTick_VAL_CURRENT_Msk, 73
 SysTick_VAL_CURRENT_Pos, 73

SystemCoreClock
 STM32F4xx_System_Exported_types, 992

SystemCoreClockUpdate
 STM32F4xx_System_Exported_Functions, 993
 STM32F4xx_System_Private_Functions, 997

SystemInit
 STM32F4xx_System_Exported_Functions, 994
 STM32F4xx_System_Private_Functions, 998

SysTick
 CMSIS, 439
 Core Definitions, 110

SysTick Clock Configuration, 333
 STK_CLK_AHB, 333
 STK_CLK_AHB_8, 333

SYSTICK Driver, 331
 STK_FCPU, 332

SysTick Functions, 118
 SysTick_Config, 118

SysTick Interrupt Configuration, 332
 STK_INTERRUPT_DISABLED, 332
 STK_INTERRUPT_ENABLED, 332

SysTick Running Mode Configuration, 334
 STK_ONE_SHOT_MODE, 334
 STK_PERIODIC_MODE, 334

SysTick Timer Instance, 170
 STK, 170

SysTick Timer Register Map, 162

SysTick_BASE
 CMSIS, 439
 Core Definitions, 111

SysTick_CALIB_NOREF_Msk
 CMSIS, 439
 System Tick Timer (SysTick), 71

SysTick_CALIB_NOREF_Pos
 CMSIS, 440
 System Tick Timer (SysTick), 71

SysTick_CALIB_SKEW_Msk
 CMSIS, 440
 System Tick Timer (SysTick), 71

SysTick_CALIB_SKEW_Pos
 CMSIS, 440
 System Tick Timer (SysTick), 71

SysTick_CALIB_TENMS_Msk
 CMSIS, 440
 System Tick Timer (SysTick), 71

SysTick_CALIB_TENMS_Pos
 CMSIS, 440
 System Tick Timer (SysTick), 71

SysTick_Config
 CMSIS, 460
 SysTick Functions, 118

SysTick_CTRL_CLKSOURCE_Msk
 CMSIS, 440
 System Tick Timer (SysTick), 71

SysTick_CTRL_CLKSOURCE_Pos
 CMSIS, 440
 System Tick Timer (SysTick), 72

SysTick_CTRL_COUNTFLAG_Msk
 CMSIS, 440
 System Tick Timer (SysTick), 72

SysTick_CTRL_COUNTFLAG_Pos
 CMSIS, 441
 System Tick Timer (SysTick), 72

SysTick_CTRL_ENABLE_Msk
 CMSIS, 441
 System Tick Timer (SysTick), 72

SysTick_CTRL_ENABLE_Pos
 CMSIS, 441
 System Tick Timer (SysTick), 72

SysTick_CTRL_TICKINT_Msk
 CMSIS, 441
 System Tick Timer (SysTick), 72

SysTick_CTRL_TICKINT_Pos
 CMSIS, 441
 System Tick Timer (SysTick), 72

SysTick_Handler
 CMSIS, 460

SysTick_IRQn
 Configuration_section_for_CMSIS, 524

SysTick_LOAD_RELOAD_Msk
 CMSIS, 441
 System Tick Timer (SysTick), 72

SysTick_LOAD_RELOAD_Pos
 CMSIS, 441
 System Tick Timer (SysTick), 73

SysTick_Type, 1056

SysTick_VAL_CURRENT_Msk
 CMSIS, 441
 System Tick Timer (SysTick), 73

SysTick_VAL_CURRENT_Pos
 CMSIS, 442

System Tick Timer (SysTick), 73

T

- ITM Functions, 143
- TAFCR
 - CMSIS, 517
- TAMP_STAMP_IRQn
 - Configuration_section_for\CMSIS, 525
- TCR
 - ITM Functions, 144
- TDHR
 - CMSIS, 517
- TDLR
 - CMSIS, 517
- TDTR
 - CMSIS, 517
- TER
 - ITM Functions, 144
- TIM1
 - Timer Instances, 175
- TIM1 Register Map, 167
- TIM1_BRK_TIM9_IRQn
 - Configuration_section_for\CMSIS, 525
- TIM1_CC_IRQn
 - Configuration_section_for\CMSIS, 525
- TIM1_TRG_COM_TIM11_IRQn
 - Configuration_section_for\CMSIS, 525
- TIM1_TypeDef, 1056
 - ARR, 1057
 - BDTR, 1057
 - CCER, 1057
 - CCMR1, 1057
 - CCMR2, 1057
 - CCR1, 1057
 - CCR2, 1057
 - CCR3, 1057
 - CCR4, 1058
 - CNT, 1058
 - CR1, 1058
 - CR2, 1058
 - DCR, 1058
 - DIER, 1058
 - DMAR, 1058
 - EGR, 1058
 - OR, 1059
 - PSC, 1059
 - RCR, 1059
 - SMCR, 1059
 - SR, 1059
- TIM1_UP_TIM10_IRQn
 - Configuration_section_for\CMSIS, 525
- TIM2
 - Timer Instances, 175
- TIM2 Register group, 340
 - TIM2_TIMER_BASE, 341
- TIM2_IRQn
 - Configuration_section_for\CMSIS, 525
- TIM2_TIMER_BASE
 - TIM2 Register group, 341

TIM_CCMR2_IC4F
 Peripheral_Registers_Bits_Definition, 959

TIM_CCMR2_IC4F_0
 Peripheral_Registers_Bits_Definition, 959

TIM_CCMR2_IC4F_1
 Peripheral_Registers_Bits_Definition, 960

TIM_CCMR2_IC4F_2
 Peripheral_Registers_Bits_Definition, 960

TIM_CCMR2_IC4F_3
 Peripheral_Registers_Bits_Definition, 960

TIM_CCMR2_IC4PSC
 Peripheral_Registers_Bits_Definition, 960

TIM_CCMR2_IC4PSC_0
 Peripheral_Registers_Bits_Definition, 960

TIM_CCMR2_IC4PSC_1
 Peripheral_Registers_Bits_Definition, 960

TIM_CCMR2_OC3CE
 Peripheral_Registers_Bits_Definition, 960

TIM_CCMR2_OC3FE
 Peripheral_Registers_Bits_Definition, 960

TIM_CCMR2_OC3M
 Peripheral_Registers_Bits_Definition, 961

TIM_CCMR2_OC3M_1
 Peripheral_Registers_Bits_Definition, 961

TIM_CCMR2_OC3M_0
 Peripheral_Registers_Bits_Definition, 961

TIM_CCMR2_OC3M_2
 Peripheral_Registers_Bits_Definition, 961

TIM_CCMR2_OC3PE
 Peripheral_Registers_Bits_Definition, 961

TIM_CCMR2_OC4CE
 Peripheral_Registers_Bits_Definition, 961

TIM_CCMR2_OC4FE
 Peripheral_Registers_Bits_Definition, 961

TIM_CCMR2_OC4M
 Peripheral_Registers_Bits_Definition, 962

TIM_CCMR2_OC4M_0
 Peripheral_Registers_Bits_Definition, 962

TIM_CCMR2_OC4M_1
 Peripheral_Registers_Bits_Definition, 962

TIM_CCMR2_OC4M_2
 Peripheral_Registers_Bits_Definition, 962

TIM_CCMR2_OC4PE
 Peripheral_Registers_Bits_Definition, 962

TIM_CCR1_CCR1
 Peripheral_Registers_Bits_Definition, 962

TIM_CCR2_CCR2
 Peripheral_Registers_Bits_Definition, 962

TIM_CCR3_CCR3
 Peripheral_Registers_Bits_Definition, 963

TIM_CCR4_CCR4
 Peripheral_Registers_Bits_Definition, 963

TIM_CNT_CNT
 Peripheral_Registers_Bits_Definition, 963

TIM_CR1_ARPE
 Peripheral_Registers_Bits_Definition, 963

TIM_CR1_CEN
 Peripheral_Registers_Bits_Definition, 963

TIM_CR1_CKD
 Peripheral_Registers_Bits_Definition, 963

TIM_CR1_CKD_0
 Peripheral_Registers_Bits_Definition, 963

TIM_CR1_CKD_1
 Peripheral_Registers_Bits_Definition, 964

TIM_CR1_CMS
 Peripheral_Registers_Bits_Definition, 964

TIM_CR1_CMS_0
 Peripheral_Registers_Bits_Definition, 964

TIM_CR1_CMS_1
 Peripheral_Registers_Bits_Definition, 964

TIM_CR1_DIR
 Peripheral_Registers_Bits_Definition, 964

TIM_CR1_OPM
 Peripheral_Registers_Bits_Definition, 964

TIM_CR1_UDIS
 Peripheral_Registers_Bits_Definition, 964

TIM_CR1_URS
 Peripheral_Registers_Bits_Definition, 965

TIM_CR2_CCDS
 Peripheral_Registers_Bits_Definition, 965

TIM_CR2_CCPC
 Peripheral_Registers_Bits_Definition, 965

TIM_CR2_CCUS
 Peripheral_Registers_Bits_Definition, 965

TIM_CR2_MMS
 Peripheral_Registers_Bits_Definition, 965

TIM_CR2_MMS_0
 Peripheral_Registers_Bits_Definition, 965

TIM_CR2_MMS_1
 Peripheral_Registers_Bits_Definition, 965

TIM_CR2_MMS_2
 Peripheral_Registers_Bits_Definition, 966

TIM_CR2_OIS1
 Peripheral_Registers_Bits_Definition, 966

TIM_CR2_OIS1N
 Peripheral_Registers_Bits_Definition, 966

TIM_CR2_OIS2
 Peripheral_Registers_Bits_Definition, 966

TIM_CR2_OIS2N
 Peripheral_Registers_Bits_Definition, 966

TIM_CR2_OIS3
 Peripheral_Registers_Bits_Definition, 966

TIM_CR2_OIS3N
 Peripheral_Registers_Bits_Definition, 966

TIM_CR2_OIS4
 Peripheral_Registers_Bits_Definition, 967

TIM_CR2_TI1S
 Peripheral_Registers_Bits_Definition, 967

TIM_DCR_DBAA
 Peripheral_Registers_Bits_Definition, 967

TIM_DCR_DBAA_0
 Peripheral_Registers_Bits_Definition, 967

TIM_DCR_DBAA_1
 Peripheral_Registers_Bits_Definition, 967

TIM_DCR_DBAA_2
 Peripheral_Registers_Bits_Definition, 967

TIM_DCR_DB_A_3
 Peripheral_Registers_Bits_Definition, 967

TIM_DCR_DB_A_4
 Peripheral_Registers_Bits_Definition, 968

TIM_DCR_DB_L
 Peripheral_Registers_Bits_Definition, 968

TIM_DCR_DB_L_0
 Peripheral_Registers_Bits_Definition, 968

TIM_DCR_DB_L_1
 Peripheral_Registers_Bits_Definition, 968

TIM_DCR_DB_L_2
 Peripheral_Registers_Bits_Definition, 968

TIM_DCR_DB_L_3
 Peripheral_Registers_Bits_Definition, 968

TIM_DCR_DB_L_4
 Peripheral_Registers_Bits_Definition, 968

TIM_DIER_BIE
 Peripheral_Registers_Bits_Definition, 968

TIM_DIER_CC1DE
 Peripheral_Registers_Bits_Definition, 969

TIM_DIER_CC1IE
 Peripheral_Registers_Bits_Definition, 969

TIM_DIER_CC2DE
 Peripheral_Registers_Bits_Definition, 969

TIM_DIER_CC2IE
 Peripheral_Registers_Bits_Definition, 969

TIM_DIER_CC3DE
 Peripheral_Registers_Bits_Definition, 969

TIM_DIER_CC3IE
 Peripheral_Registers_Bits_Definition, 969

TIM_DIER_CC4DE
 Peripheral_Registers_Bits_Definition, 969

TIM_DIER_CC4IE
 Peripheral_Registers_Bits_Definition, 970

TIM_DIER_COMDE
 Peripheral_Registers_Bits_Definition, 970

TIM_DIER_COMIE
 Peripheral_Registers_Bits_Definition, 970

TIM_DIER_TDE
 Peripheral_Registers_Bits_Definition, 970

TIM_DIER_TIE
 Peripheral_Registers_Bits_Definition, 970

TIM_DIER_UDE
 Peripheral_Registers_Bits_Definition, 970

TIM_DIER_UIE
 Peripheral_Registers_Bits_Definition, 970

TIM_DMAR_DMAB
 Peripheral_Registers_Bits_Definition, 971

TIM_EGR_BG
 Peripheral_Registers_Bits_Definition, 971

TIM_EGR_CC1G
 Peripheral_Registers_Bits_Definition, 971

TIM_EGR_CC2G
 Peripheral_Registers_Bits_Definition, 971

TIM_EGR_CC3G
 Peripheral_Registers_Bits_Definition, 971

TIM_EGR_CC4G
 Peripheral_Registers_Bits_Definition, 971

TIM_EGR_COMG
 Peripheral_Registers_Bits_Definition, 971

TIM_EGR_TG
 Peripheral_Registers_Bits_Definition, 972

TIM_EGR_UG
 Peripheral_Registers_Bits_Definition, 972

TIM_OR_ITR1_RMP
 Peripheral_Registers_Bits_Definition, 972

TIM_OR_ITR1_RMP_0
 Peripheral_Registers_Bits_Definition, 972

TIM_OR_ITR1_RMP_1
 Peripheral_Registers_Bits_Definition, 972

TIM_OR_TI4_RMP
 Peripheral_Registers_Bits_Definition, 972

TIM_OR_TI4_RMP_0
 Peripheral_Registers_Bits_Definition, 972

TIM_OR_TI4_RMP_1
 Peripheral_Registers_Bits_Definition, 973

TIM_PSC_PSC
 Peripheral_Registers_Bits_Definition, 973

TIM_RCR REP
 Peripheral_Registers_Bits_Definition, 973

TIM_SMCR_ECE
 Peripheral_Registers_Bits_Definition, 973

TIM_SMCR ETF
 Peripheral_Registers_Bits_Definition, 973

TIM_SMCR ETF_0
 Peripheral_Registers_Bits_Definition, 973

TIM_SMCR ETF_1
 Peripheral_Registers_Bits_Definition, 973

TIM_SMCR ETF_2
 Peripheral_Registers_Bits_Definition, 974

TIM_SMCR ETF_3
 Peripheral_Registers_Bits_Definition, 974

TIM_SMCR ETP
 Peripheral_Registers_Bits_Definition, 974

TIM_SMCR ETSPS
 Peripheral_Registers_Bits_Definition, 974

TIM_SMCR ETPS_0
 Peripheral_Registers_Bits_Definition, 974

TIM_SMCR ETPS_1
 Peripheral_Registers_Bits_Definition, 974

TIM_SMCR MSM
 Peripheral_Registers_Bits_Definition, 974

TIM_SMCR SMS
 Peripheral_Registers_Bits_Definition, 974

TIM_SMCR SMS_0
 Peripheral_Registers_Bits_Definition, 975

TIM_SMCR SMS_1
 Peripheral_Registers_Bits_Definition, 975

TIM_SMCR SMS_2
 Peripheral_Registers_Bits_Definition, 975

TIM_SMCR TS
 Peripheral_Registers_Bits_Definition, 975

TIM_SMCR TS_0
 Peripheral_Registers_Bits_Definition, 975

TIM_SMCR TS_1
 Peripheral_Registers_Bits_Definition, 975

TIM_SMCR_TS_2
Peripheral_Registers_Bits_Definition, 975

TIM_SR_BIF
Peripheral_Registers_Bits_Definition, 976

TIM_SR_CC1IF
Peripheral_Registers_Bits_Definition, 976

TIM_SR_CC1OF
Peripheral_Registers_Bits_Definition, 976

TIM_SR_CC2IF
Peripheral_Registers_Bits_Definition, 976

TIM_SR_CC2OF
Peripheral_Registers_Bits_Definition, 976

TIM_SR_CC3IF
Peripheral_Registers_Bits_Definition, 976

TIM_SR_CC3OF
Peripheral_Registers_Bits_Definition, 976

TIM_SR_CC4IF
Peripheral_Registers_Bits_Definition, 977

TIM_SR_CC4OF
Peripheral_Registers_Bits_Definition, 977

TIM_SR_COMIF
Peripheral_Registers_Bits_Definition, 977

TIM_SR_TIF
Peripheral_Registers_Bits_Definition, 977

TIM_SR_UIF
Peripheral_Registers_Bits_Definition, 977

TIM_TypeDef, 1059

TIMER Driver, 339

Timer Instances, 174

- TIM1, 175
- TIM2, 175

Timer2_init
API Function Prototypes, 343

TIR
CMSIS, 517

TPI
CMSIS, 442

Core Definitions, 111

TPI_ACPR_PRESCALER_Msk
CMSIS, 442

Trace Port Interface (TPI), 91

TPI_ACPR_PRESCALER_Pos
CMSIS, 442

Trace Port Interface (TPI), 91

TPI_BASE
CMSIS, 442

Core Definitions, 111

TPI_DEVID_AsynClkIn_Msk
CMSIS, 442

Trace Port Interface (TPI), 91

TPI_DEVID_AsynClkIn_Pos
CMSIS, 442

Trace Port Interface (TPI), 91

TPI_DEVID_MANCVALID_Msk
CMSIS, 443

Trace Port Interface (TPI), 91

TPI_DEVID_MANCVALID_Pos
CMSIS, 443

Trace Port Interface (TPI), 91

TPI_DEVID_MinBufSz_Msk
CMSIS, 443

Trace Port Interface (TPI), 91

TPI_DEVID_MinBufSz_Pos
CMSIS, 443

Trace Port Interface (TPI), 92

TPI_DEVID_NrTraceInput_Msk
CMSIS, 443

Trace Port Interface (TPI), 92

TPI_DEVID_NrTraceInput_Pos
CMSIS, 443

Trace Port Interface (TPI), 92

TPI_DEVID_NRZVALID_Msk
CMSIS, 443

Trace Port Interface (TPI), 92

TPI_DEVID_NRZVALID_Pos
CMSIS, 443

Trace Port Interface (TPI), 92

TPI_DEVID_PTINVALID_Msk
CMSIS, 444

Trace Port Interface (TPI), 92

TPI_DEVID_PTINVALID_Pos
CMSIS, 444

Trace Port Interface (TPI), 92

TPI_DEVTYPE_MajorType_Msk
CMSIS, 444

Trace Port Interface (TPI), 92

TPI_DEVTYPE_MajorType_Pos
CMSIS, 444

Trace Port Interface (TPI), 93

TPI_DEVTYPE_SubType_Msk
CMSIS, 444

Trace Port Interface (TPI), 93

TPI_DEVTYPE_SubType_Pos
CMSIS, 444

Trace Port Interface (TPI), 93

TPI_FFCR_EnFCont_Msk
CMSIS, 444

Trace Port Interface (TPI), 93

TPI_FFCR_EnFCont_Pos
CMSIS, 444

Trace Port Interface (TPI), 93

TPI_FFCR_TrigIn_Msk
CMSIS, 445

Trace Port Interface (TPI), 93

TPI_FFCR_TrigIn_Pos
CMSIS, 445

Trace Port Interface (TPI), 93

TPI_FFSR_FInProg_Msk
CMSIS, 445

Trace Port Interface (TPI), 93

TPI_FFSR_FInProg_Pos
CMSIS, 445

Trace Port Interface (TPI), 94

TPI_FFSR_FtNonStop_Msk
CMSIS, 445

Trace Port Interface (TPI), 94

TPI_FFSR_FtNonStop_Pos
 CMSIS, 445
 Trace Port Interface (TPI), 94

TPI_FFSR_FtStopped_Msk
 CMSIS, 445
 Trace Port Interface (TPI), 94

TPI_FFSR_FtStopped_Pos
 CMSIS, 445
 Trace Port Interface (TPI), 94

TPI_FFSR_TCPresent_Msk
 CMSIS, 446
 Trace Port Interface (TPI), 94

TPI_FFSR_TCPresent_Pos
 CMSIS, 446
 Trace Port Interface (TPI), 94

TPI_FIFO0_ETM0_Msk
 CMSIS, 446
 Trace Port Interface (TPI), 94

TPI_FIFO0_ETM0_Pos
 CMSIS, 446
 Trace Port Interface (TPI), 95

TPI_FIFO0_ETM1_Msk
 CMSIS, 446
 Trace Port Interface (TPI), 95

TPI_FIFO0_ETM1_Pos
 CMSIS, 446
 Trace Port Interface (TPI), 95

TPI_FIFO0_ETM2_Msk
 CMSIS, 446
 Trace Port Interface (TPI), 95

TPI_FIFO0_ETM2_Pos
 CMSIS, 446
 Trace Port Interface (TPI), 95

TPI_FIFO0_ETM_ATVALID_Msk
 CMSIS, 447
 Trace Port Interface (TPI), 95

TPI_FIFO0_ETM_ATVALID_Pos
 CMSIS, 447
 Trace Port Interface (TPI), 95

TPI_FIFO0_ETM_bytecount_Msk
 CMSIS, 447
 Trace Port Interface (TPI), 95

TPI_FIFO0_ETM_bytecount_Pos
 CMSIS, 447
 Trace Port Interface (TPI), 96

TPI_FIFO0_ITM_ATVALID_Msk
 CMSIS, 447
 Trace Port Interface (TPI), 96

TPI_FIFO0_ITM_ATVALID_Pos
 CMSIS, 447
 Trace Port Interface (TPI), 96

TPI_FIFO0_ITM_bytecount_Msk
 CMSIS, 447
 Trace Port Interface (TPI), 96

TPI_FIFO0_ITM_bytecount_Pos
 CMSIS, 447
 Trace Port Interface (TPI), 96

TPI_FIFO1_ETM_ATVALID_Msk
 CMSIS, 448
 Trace Port Interface (TPI), 96

TPI_FIFO1_ETM_ATVALID_Pos
 CMSIS, 448
 Trace Port Interface (TPI), 96

TPI_FIFO1_ETM_bytecount_Msk
 CMSIS, 448
 Trace Port Interface (TPI), 96

TPI_FIFO1_ETM_bytecount_Pos
 CMSIS, 448
 Trace Port Interface (TPI), 97

TPI_FIFO1_ITM0_Msk
 CMSIS, 448
 Trace Port Interface (TPI), 97

TPI_FIFO1_ITM0_Pos
 CMSIS, 448
 Trace Port Interface (TPI), 97

TPI_FIFO1_ITM1_Msk
 CMSIS, 448
 Trace Port Interface (TPI), 97

TPI_FIFO1_ITM1_Pos
 CMSIS, 448
 Trace Port Interface (TPI), 97

TPI_FIFO1_ITM2_Msk
 CMSIS, 449
 Trace Port Interface (TPI), 97

TPI_FIFO1_ITM2_Pos
 CMSIS, 449
 Trace Port Interface (TPI), 97

TPI_FIFO1_ITM_ATVALID_Msk
 CMSIS, 449
 Trace Port Interface (TPI), 97

TPI_FIFO1_ITM_ATVALID_Pos
 CMSIS, 449
 Trace Port Interface (TPI), 98

TPI_FIFO1_ITM_bytecount_Msk
 CMSIS, 449
 Trace Port Interface (TPI), 98

TPI_FIFO1_ITM_bytecount_Pos
 CMSIS, 449
 Trace Port Interface (TPI), 98

TPI_ITATBCTR0_ATREADY_Msk
 CMSIS, 449
 Trace Port Interface (TPI), 98

TPI_ITATBCTR0_ATREADY_Pos
 CMSIS, 449
 Trace Port Interface (TPI), 98

TPI_ITATBCTR2_ATREADY_Msk
 CMSIS, 450
 Trace Port Interface (TPI), 98

TPI_ITATBCTR2_ATREADY_Pos
 CMSIS, 450
 Trace Port Interface (TPI), 98

TPI_ITCTRL_Mode_Msk
 CMSIS, 450
 Trace Port Interface (TPI), 98

TPI_ITCTRL_Mode_Pos
 CMSIS, 450

Trace Port Interface (TPI), 99
TPI_SPPR_TXMODE_Msk
 CMSIS, 450
 Trace Port Interface (TPI), 99
TPI_SPPR_TXMODE_Pos
 CMSIS, 450
 Trace Port Interface (TPI), 99
TPI_TRIGGER_TRIGGER_Msk
 CMSIS, 450
 Trace Port Interface (TPI), 99
TPI_TRIGGER_TRIGGER_Pos
 CMSIS, 450
 Trace Port Interface (TPI), 99
TPI_Type, 1060
TPR
 ITM Functions, 144
TR
 CMSIS, 517
Trace Port Interface (TPI), 89
 TPI_ACPR_PRESCALER_Msk, 91
 TPI_ACPR_PRESCALER_Pos, 91
 TPI_DEVID_AsynClkIn_Msk, 91
 TPI_DEVID_AsynClkIn_Pos, 91
 TPI_DEVID_MANCVALID_Msk, 91
 TPI_DEVID_MANCVALID_Pos, 91
 TPI_DEVID_MinBufSz_Msk, 91
 TPI_DEVID_MinBufSz_Pos, 92
 TPI_DEVID_NrTraceInput_Msk, 92
 TPI_DEVID_NrTraceInput_Pos, 92
 TPI_DEVID_NRZVALID_Msk, 92
 TPI_DEVID_NRZVALID_Pos, 92
 TPI_DEVID_PTINVALIDID_Msk, 92
 TPI_DEVID_PTINVALIDID_Pos, 92
 TPI_DEVTYPE_MajorType_Msk, 92
 TPI_DEVTYPE_MajorType_Pos, 93
 TPI_DEVTYPE_SubType_Msk, 93
 TPI_DEVTYPE_SubType_Pos, 93
 TPI_FFCR_EnFCont_Msk, 93
 TPI_FFCR_EnFCont_Pos, 93
 TPI_FFCR_TrigIn_Msk, 93
 TPI_FFCR_TrigIn_Pos, 93
 TPI_FFSR_FInProg_Msk, 93
 TPI_FFSR_FInProg_Pos, 94
 TPI_FFSR_FtNonStop_Msk, 94
 TPI_FFSR_FtNonStop_Pos, 94
 TPI_FFSR_FtStopped_Msk, 94
 TPI_FFSR_FtStopped_Pos, 94
 TPI_FFSR_TCPresent_Msk, 94
 TPI_FFSR_TCPresent_Pos, 94
 TPI_FIFO0_ETM0_Msk, 94
 TPI_FIFO0_ETM0_Pos, 95
 TPI_FIFO0_ETM1_Msk, 95
 TPI_FIFO0_ETM1_Pos, 95
 TPI_FIFO0_ETM2_Msk, 95
 TPI_FIFO0_ETM2_Pos, 95
 TPI_FIFO0_ETM_ATVALID_Msk, 95
 TPI_FIFO0_ETM_ATVALID_Pos, 95
 TPI_FIFO0_ETM_bytecount_Msk, 95
 TPI_FIFO0_ETM_bytecount_Pos, 96
 TPI_FIFO0_ITM_ATVALID_Msk, 96
 TPI_FIFO0_ITM_ATVALID_Pos, 96
 TPI_FIFO0_ITM_bytecount_Msk, 96
 TPI_FIFO0_ITM_bytecount_Pos, 96
 TPI_FIFO1_ETM_ATVALID_Msk, 96
 TPI_FIFO1_ETM_ATVALID_Pos, 96
 TPI_FIFO1_ETM_bytecount_Msk, 96
 TPI_FIFO1_ETM_bytecount_Pos, 97
 TPI_FIFO1_ITM0_Msk, 97
 TPI_FIFO1_ITM0_Pos, 97
 TPI_FIFO1_ITM1_Msk, 97
 TPI_FIFO1_ITM1_Pos, 97
 TPI_FIFO1_ITM2_Msk, 97
 TPI_FIFO1_ITM2_Pos, 97
 TPI_FIFO1_ITM_ATVALID_Msk, 97
 TPI_FIFO1_ITM_ATVALID_Pos, 98
 TPI_FIFO1_ITM_bytecount_Msk, 98
 TPI_FIFO1_ITM_bytecount_Pos, 98
 TPI_ITATBCTR0_ATREADY_Msk, 98
 TPI_ITATBCTR0_ATREADY_Pos, 98
 TPI_ITATBCTR2_ATREADY_Msk, 98
 TPI_ITATBCTR2_ATREADY_Pos, 98
 TPI_ITCTRL_Mode_Msk, 98
 TPI_ITCTRL_Mode_Pos, 99
 TPI_SPPR_TXMODE_Msk, 99
 TPI_SPPR_TXMODE_Pos, 99
 TPI_TRIGGER_TRIGGER_Msk, 99
 TPI_TRIGGER_TRIGGER_Pos, 99
TRIGGER
 ITM Functions, 144
Trigger_Alarm
 API_Definitions, 377
 ECU APIs, 367
TRISE
 CMSIS, 517
 I2C_TypeDef, 1021
TRUE
 Platform Standard Types, 152
TSDR
 CMSIS, 518
TSR
 CMSIS, 518
TSSR
 CMSIS, 518
TSTR
 CMSIS, 518
TWCR
 CMSIS, 518
TXCRCR
 CMSIS, 518
 SPI_TypeDef, 1051
TXE
 S_IRQ_SRC, 1042
u16
 ITM Functions, 144
u32
 ITM Functions, 144, 145

u8
 ITM Functions, 145
UART8_BASE
 Peripheral_memory_map, 536
uc16
 Exported_types, 528
uc32
 Exported_types, 528
uc8
 Exported_types, 528
uint16
 Platform Standard Types, 154
uint16_least
 Platform Standard Types, 154
uint32
 Platform Standard Types, 154
uint32_least
 Platform Standard Types, 154
uint64
 Platform Standard Types, 154
uint8
 Platform Standard Types, 154
uint8_least
 Platform Standard Types, 154
UsageFault_IRQHandler
 Configuration_section_for\CMSIS, 524
USART Driver, 344
USART Instances, 175
 USART1, 176
 USART2, 176
 USART6, 176
USART Register Map, 165
USART1
 USART Instances, 176
USART1_IRQHandler
 Configuration_section_for\CMSIS, 526
USART2
 USART Instances, 176
USART2_IRQHandler
 Configuration_section_for\CMSIS, 526
USART6
 USART Instances, 176
USART6_IRQHandler
 Configuration_section_for\CMSIS, 526
USART_BRR_DIV_Fraction
 PeripheralRegisters_Bits_Definition, 977
USART_BRR_DIV_Mantissa
 PeripheralRegisters_Bits_Definition, 977
USART_cfg_t, 1061
USART_CR1_IDLEIE
 PeripheralRegisters_Bits_Definition, 978
USART_CR1_M
 PeripheralRegisters_Bits_Definition, 978
USART_CR1_OVER8
 PeripheralRegisters_Bits_Definition, 978
USART_CR1_PCE
 PeripheralRegisters_Bits_Definition, 978
USART_CR1_PEIE
 PeripheralRegisters_Bits_Definition, 978
PeripheralRegisters_Bits_Definition, 978
USART_CR1_PS
 PeripheralRegisters_Bits_Definition, 978
USART_CR1_RE
 PeripheralRegisters_Bits_Definition, 978
USART_CR1_RWU
 PeripheralRegisters_Bits_Definition, 979
USART_CR1_RXNEIE
 PeripheralRegisters_Bits_Definition, 979
USART_CR1_SBK
 PeripheralRegisters_Bits_Definition, 979
USART_CR1_TCIE
 PeripheralRegisters_Bits_Definition, 979
USART_CR1_TE
 PeripheralRegisters_Bits_Definition, 979
USART_CR1_TXEIE
 PeripheralRegisters_Bits_Definition, 979
USART_CR1_UE
 PeripheralRegisters_Bits_Definition, 979
USART_CR1_WAKE
 PeripheralRegisters_Bits_Definition, 980
USART_CR2_ADD
 PeripheralRegisters_Bits_Definition, 980
USART_CR2_CLKEN
 PeripheralRegisters_Bits_Definition, 980
USART_CR2_CPHA
 PeripheralRegisters_Bits_Definition, 980
USART_CR2_CPOL
 PeripheralRegisters_Bits_Definition, 980
USART_CR2_LBCL
 PeripheralRegisters_Bits_Definition, 980
USART_CR2_LBDIE
 PeripheralRegisters_Bits_Definition, 980
USART_CR2_LBDL
 PeripheralRegisters_Bits_Definition, 981
USART_CR2_LINE_N
 PeripheralRegisters_Bits_Definition, 981
USART_CR2_STOP
 PeripheralRegisters_Bits_Definition, 981
USART_CR2_STOP_0
 PeripheralRegisters_Bits_Definition, 981
USART_CR2_STOP_1
 PeripheralRegisters_Bits_Definition, 981
USART_CR3_CTSE
 PeripheralRegisters_Bits_Definition, 981
USART_CR3_CTSIE
 PeripheralRegisters_Bits_Definition, 981
USART_CR3_DMAR
 PeripheralRegisters_Bits_Definition, 982
USART_CR3_DMAT
 PeripheralRegisters_Bits_Definition, 982
USART_CR3_EIE
 PeripheralRegisters_Bits_Definition, 982
USART_CR3_HDSEL
 PeripheralRegisters_Bits_Definition, 982
USART_CR3_IREN
 PeripheralRegisters_Bits_Definition, 982
USART_CR3_IRLP

Peripheral_Registers_Bits_Definition, 982
USART_CR3_NACK
Peripheral_Registers_Bits_Definition, 982
USART_CR3_ONEBIT
Peripheral_Registers_Bits_Definition, 983
USART_CR3_RTSE
Peripheral_Registers_Bits_Definition, 983
USART_CR3_SCEN
Peripheral_Registers_Bits_Definition, 983
USART_DR_DR
Peripheral_Registers_Bits_Definition, 983
USART_GTPR_GT
Peripheral_Registers_Bits_Definition, 983
USART_GTPR_PSC
Peripheral_Registers_Bits_Definition, 983
USART_GTPR_PSC_0
Peripheral_Registers_Bits_Definition, 983
USART_GTPR_PSC_1
Peripheral_Registers_Bits_Definition, 984
USART_GTPR_PSC_2
Peripheral_Registers_Bits_Definition, 984
USART_GTPR_PSC_3
Peripheral_Registers_Bits_Definition, 984
USART_GTPR_PSC_4
Peripheral_Registers_Bits_Definition, 984
USART_GTPR_PSC_5
Peripheral_Registers_Bits_Definition, 984
USART_GTPR_PSC_6
Peripheral_Registers_Bits_Definition, 984
USART_GTPR_PSC_7
Peripheral_Registers_Bits_Definition, 984
USART_SR_CTS
Peripheral_Registers_Bits_Definition, 984
USART_SR_FE
Peripheral_Registers_Bits_Definition, 985
USART_SR_IDLE
Peripheral_Registers_Bits_Definition, 985
USART_SR_LBD
Peripheral_Registers_Bits_Definition, 985
USART_SR_NE
Peripheral_Registers_Bits_Definition, 985
USART_SR_ORE
Peripheral_Registers_Bits_Definition, 985
USART_SR_PE
Peripheral_Registers_Bits_Definition, 985
USART_SR_RXNE
Peripheral_Registers_Bits_Definition, 985
USART_SR_TC
Peripheral_Registers_Bits_Definition, 986
USART_SR_TXE
Peripheral_Registers_Bits_Definition, 986
USART_TypeDef, 1062
BRR, 1062
CR1, 1063
CR2, 1063
CR3, 1063
DR, 1063
GTPR, 1063
SR, 1063
UserLCD_PrintFreeSlots
API_Definitions, 377
ECU APIs, 368

V
ITM Functions, 145

VAL
ITM Functions, 146
STK_TypeDef, 1054

VECT_TAB_OFFSET
STM32F4xx_System_Private_Defines, 995

VoidPtr
Platform Standard Types, 155

vsc16
Exported_types, 529

vsc32
Exported_types, 529

vsc8
Exported_types, 529

VTOR
ITM Functions, 146
SCB_TypeDef, 1048

vuc16
Exported_types, 529

vuc32
Exported_types, 529

vuc8
Exported_types, 529

vuint16_t
Platform Standard Types, 155

vuint32_t
Platform Standard Types, 155

vuint8_t
Platform Standard Types, 155

W
ITM Functions, 146

WHPCR
CMSIS, 518

WithoutStop
I2C Driver, 271

WithStop
I2C Driver, 271

WPR
CMSIS, 519

Wrong_RFID
API_Definitions, 378
ECU APIs, 369

WUTR
CMSIS, 519

WVPCR
CMSIS, 519

WWDG_CFR_EWI
Peripheral_Registers_Bits_Definition, 986

WWDG_CFR_W
Peripheral_Registers_Bits_Definition, 986

WWDG_CFR_W_0
Peripheral_Registers_Bits_Definition, 986

WWDG_CFR_W_1
 Peripheral_Registers_Bits_Definition, [986](#)
WWDG_CFR_W_2
 Peripheral_Registers_Bits_Definition, [986](#)
WWDG_CFR_W_3
 Peripheral_Registers_Bits_Definition, [987](#)
WWDG_CFR_W_4
 Peripheral_Registers_Bits_Definition, [987](#)
WWDG_CFR_W_5
 Peripheral_Registers_Bits_Definition, [987](#)
WWDG_CFR_W_6
 Peripheral_Registers_Bits_Definition, [987](#)
WWDG_CFR_WDGTB
 Peripheral_Registers_Bits_Definition, [987](#)
WWDG_CFR_WDGTB_0
 Peripheral_Registers_Bits_Definition, [987](#)
WWDG_CFR_WDGTB_1
 Peripheral_Registers_Bits_Definition, [987](#)
WWDG_CR_T
 Peripheral_Registers_Bits_Definition, [987](#)
WWDG_CR_T_0
 Peripheral_Registers_Bits_Definition, [988](#)
WWDG_CR_T_1
 Peripheral_Registers_Bits_Definition, [988](#)
WWDG_CR_T_2
 Peripheral_Registers_Bits_Definition, [988](#)
WWDG_CR_T_4
 Peripheral_Registers_Bits_Definition, [988](#)
WWDG_CR_T_5
 Peripheral_Registers_Bits_Definition, [988](#)
WWDG_CR_T_6
 Peripheral_Registers_Bits_Definition, [988](#)
WWDG_CR_WDGA
 Peripheral_Registers_Bits_Definition, [988](#)
WWDG_IRQn
 Configuration_section_for_CMSIS, [524](#)
WWDG_SR_EWIF
 Peripheral_Registers_Bits_Definition, [989](#)
WWDG_TypeDef, [1064](#)

xPSR_C_Msk
 CMSIS, [451](#)
 Status and Control Registers, [47](#)
xPSR_C_Pos
 CMSIS, [451](#)
 Status and Control Registers, [47](#)
xPSR_GE_Msk
 CMSIS, [451](#)
 Status and Control Registers, [47](#)
xPSR_GE_Pos
 CMSIS, [451](#)
 Status and Control Registers, [47](#)
xPSR_ISR_Msk
 CMSIS, [451](#)
 Status and Control Registers, [47](#)
xPSR_ISR_Pos
 CMSIS, [451](#)

xPSR_IT_Msk
 CMSIS, [451](#)
 Status and Control Registers, [47](#)
xPSR_IT_Pos
 CMSIS, [451](#)
 Status and Control Registers, [47](#)
xPSR_N_Msk
 CMSIS, [452](#)
 Status and Control Registers, [48](#)
xPSR_N_Pos
 CMSIS, [452](#)
 Status and Control Registers, [48](#)
xPSR_Q_Msk
 CMSIS, [452](#)
 Status and Control Registers, [48](#)
xPSR_Q_Pos
 CMSIS, [452](#)
 Status and Control Registers, [48](#)
xPSR_T_Msk
 CMSIS, [452](#)
 Status and Control Registers, [48](#)
xPSR_T_Pos
 CMSIS, [452](#)
 Status and Control Registers, [48](#)
xPSR_Type, [1064](#)
xPSR_V_Msk
 CMSIS, [452](#)
 Status and Control Registers, [48](#)
xPSR_V_Pos
 CMSIS, [452](#)
 Status and Control Registers, [48](#)
xPSR_Z_Msk
 CMSIS, [453](#)
 Status and Control Registers, [49](#)
xPSR_Z_Pos
 CMSIS, [453](#)
 Status and Control Registers, [49](#)

Z
 ITM Functions, [146](#), [147](#)