

## ADAS Graduation Project

Generated on Mon Nov 28 2022 19:04:42 for ADAS Graduation Project by Doxygen 1.9.5

Mon Nov 28 2022 19:04:42



<b>1 ADAS Graudation Project</b>	<b>1</b>
1.1 Description	1
1.1.1 Adaptive Cruise Control	1
1.1.2 Adaptive Light Control	1
1.1.3 Driver Health Care	1
1.2 Hardware specifications	1
<b>2 Module Index</b>	<b>3</b>
2.1 Modules	3
<b>3 Data Structure Index</b>	<b>5</b>
3.1 Data Structures	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Module Documentation</b>	<b>9</b>
5.1 Bit Manipulation Math Macros	9
5.1.1 Detailed Description	9
5.1.2 Macro Definition Documentation	9
5.1.2.1 SET_BIT	9
5.1.2.2 CLR_BIT	10
5.1.2.3 TOGGLE_BIT	10
5.1.2.4 GET_BIT	10
5.2 Bit Group Manipulation Math Macros	10
5.2.1 Detailed Description	11
5.2.2 Macro Definition Documentation	11
5.2.2.1 SET_BITS	11
5.2.2.2 CLR_BITS	11
5.2.2.3 TOGGLE_BITS	11
5.2.2.4 GET_BITS	12
5.3 Compiler standard macros	12
5.3.1 Detailed Description	12
5.3.2 Macro Definition Documentation	12
5.3.2.1 VAR	12
5.3.2.2 FUNC	13
5.3.2.3 P2VAR	13
5.3.2.4 P2CONST	13
5.3.2.5 CONSTP2VAR	13
5.3.2.6 CONSTP2CONST	14
5.3.2.7 P2FUNC	14
5.3.2.8 CONST	14
5.3.2.9 STATIC	14
5.4 Utilities macros	15

5.4.1 Detailed Description	15
5.4.2 Macro Definition Documentation	15
5.4.2.1 MY_MS_DELAY	15
5.5 Standard types	15
5.5.1 Detailed Description	15
5.5.2 Typedef Documentation	15
5.5.2.1 bool_t	16
5.5.2.2 u8_t	16
5.5.2.3 u16_t	16
5.5.2.4 u32_t	16
5.5.2.5 s8_t	17
5.5.2.6 s16_t	17
5.5.2.7 s32_t	17
5.5.2.8 fl32_t	17
5.5.2.9 fl64_t	18
5.6 Standard values	18
5.6.1 Detailed Description	18
5.6.2 Macro Definition Documentation	18
5.6.2.1 TRUE	18
5.6.2.2 FALSE	18
5.6.2.3 NULL	19
5.6.2.4 INITIAL_ZERO	19
5.6.2.5 FLAG_SET	19
5.6.2.6 FLAG_CLEARED	19
5.7 GPIO Modes	20
5.7.1 Detailed Description	20
5.7.2 Macro Definition Documentation	20
5.7.2.1 GPIOx_MODE_INPUT	20
5.7.2.2 GPIOx_MODE_OUTPUT	20
5.7.2.3 GPIOx_MODE_AF	20
5.7.2.4 GPIOx_MODE_ANALOG	21
5.8 GPIO Ports	21
5.8.1 Detailed Description	21
5.8.2 Macro Definition Documentation	21
5.8.2.1 GPIO_PORTA	21
5.8.2.2 GPIO_PORTB	21
5.8.2.3 GPIO_PORTC	22
5.9 GPIO Output Types	22
5.9.1 Detailed Description	22
5.9.2 Macro Definition Documentation	22
5.9.2.1 GPIOx_OPENDRAIN	22
5.9.2.2 GPIOx_PUSHPULL	22

5.10 GPIO PIN Speed . . . . .	23
5.10.1 Detailed Description . . . . .	23
5.10.2 Macro Definition Documentation . . . . .	23
5.10.2.1 GPIOx_LowSpeed . . . . .	23
5.10.2.2 GPIOx_MediumSpeed . . . . .	23
5.10.2.3 GPIOx_HighSpeed . . . . .	23
5.10.2.4 GPIOx_VeryHighSpeed . . . . .	24
5.11 GPIO Pull Types . . . . .	24
5.11.1 Detailed Description . . . . .	24
5.11.2 Macro Definition Documentation . . . . .	24
5.11.2.1 GPIOx_NoPull . . . . .	24
5.11.2.2 GPIOx_PullUp . . . . .	24
5.11.2.3 GPIOx_PullDown . . . . .	25
5.12 GPIO Output Values . . . . .	25
5.12.1 Detailed Description . . . . .	25
5.12.2 Macro Definition Documentation . . . . .	25
5.12.2.1 GPIOx_HIGH . . . . .	25
5.12.2.2 GPIOx_LOW . . . . .	25
5.13 GPIO Output PINs . . . . .	26
5.13.1 Detailed Description . . . . .	26
5.13.2 Macro Definition Documentation . . . . .	26
5.13.2.1 GPIOx_PIN0 . . . . .	26
5.13.2.2 GPIOx_PIN1 . . . . .	27
5.13.2.3 GPIOx_PIN2 . . . . .	27
5.13.2.4 GPIOx_PIN3 . . . . .	27
5.13.2.5 GPIOx_PIN4 . . . . .	27
5.13.2.6 GPIOx_PIN5 . . . . .	28
5.13.2.7 GPIOx_PIN6 . . . . .	28
5.13.2.8 GPIOx_PIN7 . . . . .	28
5.13.2.9 GPIOx_PIN8 . . . . .	28
5.13.2.10 GPIOx_PIN9 . . . . .	29
5.13.2.11 GPIOx_PIN10 . . . . .	29
5.13.2.12 GPIOx_PIN11 . . . . .	29
5.13.2.13 GPIOx_PIN12 . . . . .	29
5.13.2.14 GPIOx_PIN13 . . . . .	30
5.13.2.15 GPIOx_PIN14 . . . . .	30
5.13.2.16 GPIOx_PIN15 . . . . .	30
5.14 GPIO Alternate Functions . . . . .	30
5.14.1 Detailed Description . . . . .	31
5.14.2 Macro Definition Documentation . . . . .	31
5.14.2.1 GPIOx_AF0 . . . . .	31
5.14.2.2 GPIOx_AF1 . . . . .	31

5.14.2.3 GPIOx_AF2	32
5.14.2.4 GPIOx_AF3	32
5.14.2.5 GPIOx_AF4	32
5.14.2.6 GPIOx_AF5	32
5.14.2.7 GPIOx_AF6	33
5.14.2.8 GPIOx_AF7	33
5.14.2.9 GPIOx_AF8	33
5.14.2.10 GPIOx_AF9	33
5.14.2.11 GPIOx_AF10	34
5.14.2.12 GPIOx_AF11	34
5.14.2.13 GPIOx_AF12	34
5.14.2.14 GPIOx_AF13	34
5.14.2.15 GPIOx_AF14	35
5.14.2.16 GPIOx_AF15	35
5.15 GPIO Addresses	35
5.15.1 Detailed Description	35
5.15.2 Macro Definition Documentation	35
5.15.2.1 GPIOA_BASE_ADDRESS	35
5.15.2.2 GPIOB_BASE_ADDRESS	36
5.15.2.3 GPIOC_BASE_ADDRESS	36
5.16 GPIO Registers	36
5.16.1 Detailed Description	36
5.16.2 Macro Definition Documentation	36
5.16.2.1 GPIOA	36
5.16.2.2 GPIOB	37
5.16.2.3 GPIOC	37
<b>6 Data Structure Documentation</b>	<b>39</b>
6.1 GPIOx_MemoryMapType Struct Reference	39
6.1.1 Detailed Description	39
6.1.2 Field Documentation	40
6.1.2.1 MODERx	40
6.1.2.2 OTYPERx	40
6.1.2.3 OSPEEDRx	40
6.1.2.4 PUPDRx	40
6.1.2.5 IDRx	40
6.1.2.6 ODRx	41
6.1.2.7 BSRRx	41
6.1.2.8 LCKRx	41
6.1.2.9 AFRLx	41
6.1.2.10 AFRHx	41
6.2 MGPIOx_ConfigType Struct Reference	42

6.2.1 Detailed Description . . . . .	42
6.2.2 Field Documentation . . . . .	42
6.2.2.1 Port . . . . .	42
6.2.2.2 Pin . . . . .	43
6.2.2.3 Mode . . . . .	43
6.2.2.4 OutputType . . . . .	43
6.2.2.5 OutputSpeed . . . . .	43
6.2.2.6 InputType . . . . .	44
6.2.2.7 AF_Type . . . . .	44
6.3 RCC_MemoryMapType Struct Reference . . . . .	44
6.3.1 Detailed Description . . . . .	45
6.3.2 Field Documentation . . . . .	45
6.3.2.1 CR . . . . .	45
6.3.2.2 PLLCFGR . . . . .	45
6.3.2.3 CFGR . . . . .	45
6.3.2.4 CIR . . . . .	46
6.3.2.5 AHB1RSTR . . . . .	46
6.3.2.6 AHB2RSTR . . . . .	46
6.3.2.7 Reserved1 . . . . .	46
6.3.2.8 Reserved2 . . . . .	46
6.3.2.9 APB1RSTR . . . . .	46
6.3.2.10 APB2RSTR . . . . .	47
6.3.2.11 Reserved3 . . . . .	47
6.3.2.12 Reserved4 . . . . .	47
6.3.2.13 AHB1ENR . . . . .	47
6.3.2.14 AHB2ENR . . . . .	47
6.3.2.15 Reserved5 . . . . .	47
6.3.2.16 Reserved6 . . . . .	48
6.3.2.17 APB1ENR . . . . .	48
6.3.2.18 APB2ENR . . . . .	48
6.3.2.19 Reserved7 . . . . .	48
6.3.2.20 Reserved8 . . . . .	48
6.3.2.21 AHB1LPENR . . . . .	48
6.3.2.22 AHB2LPENR . . . . .	49
6.3.2.23 Reserved9 . . . . .	49
6.3.2.24 Reserved10 . . . . .	49
6.3.2.25 APB1LPENR . . . . .	49
6.3.2.26 APB2LPENR . . . . .	49
6.3.2.27 Reserved11 . . . . .	49
6.3.2.28 Reserved12 . . . . .	50
6.3.2.29 BDCR . . . . .	50
6.3.2.30 CSR . . . . .	50

6.3.2.31 Reserved13	50
6.3.2.32 Reserved14	50
6.3.2.33 SSCGR	50
6.3.2.34 PLLI2SCFGR	50
6.3.2.35 DCKCFGR	50
<b>7 File Documentation</b>	<b>51</b>
7.1 README.md File Reference	51
7.2 src/COTS/LIB/LSTD_BITMATH.h File Reference	51
7.2.1 Detailed Description	51
7.3 LSTD_BITMATH.h	52
7.4 src/COTS/LIB/LSTD_COMPILER.h File Reference	52
7.4.1 Detailed Description	52
7.5 LSTD_COMPILER.h	53
7.6 src/COTS/LIB/LSTD_MCU_UTILITIES.h File Reference	53
7.6.1 Detailed Description	53
7.7 LSTD_MCU_UTILITIES.h	54
7.8 src/COTS/LIB/LSTD_TYPES.h File Reference	54
7.8.1 Detailed Description	54
7.9 LSTD_TYPES.h	55
7.10 src/COTS/LIB/LSTD_VALUES.h File Reference	55
7.10.1 Detailed Description	55
7.11 LSTD_VALUES.h	56
7.12 src/COTS/MCAL/GPIO/GPIO_config.h File Reference	56
7.12.1 Detailed Description	56
7.12.2 Macro Definition Documentation	57
7.12.2.1 GPIOA_PIN_POS	57
7.12.2.2 GPIOB_PIN_POS	57
7.12.2.3 LCKK_BIT_POS	57
7.13 GPIO_config.h	57
7.14 src/COTS/MCAL/GPIO/GPIO_interface.h File Reference	57
7.14.1 Detailed Description	60
7.14.2 Function Documentation	61
7.14.2.1 MGPIOn_vLockedPins()	61
7.14.2.2 MGPIOn_vSetPinMode()	61
7.14.2.3 MGPIOn_vSetPinOutputType()	62
7.14.2.4 MGPIOn_vSetPinOutputSpeed()	62
7.14.2.5 MGPIOn_vSetPinInputPullType()	62
7.14.2.6 MGPIOn_u8GetPinValue()	64
7.14.2.7 MGPIOn_vSetPinValue()	64
7.14.2.8 MGPIOn_vSetResetAtomic()	64
7.14.2.9 MGPIOn_vSetAlternateFunctionON()	65



7.14.2.10 MGPI0x_vSetPortConfigLock()	65
7.14.2.11 MGPI0x_vInit()	65
7.14.2.12 MGPI0x_vTogglePinValue()	66
7.15 GPIO_interface.h	66
7.16 src/COTS/MCAL/GPIO/GPIO_private.h File Reference	68
7.16.1 Detailed Description	68
7.17 GPIO_private.h	69
7.18 src/COTS/MCAL/GPIO/GPIO_program.c File Reference	69
7.18.1 Detailed Description	70
7.18.2 Function Documentation	70
7.18.2.1 MGPI0x_vLockedPins()	70
7.18.2.2 MGPI0x_vSetPinMode()	71
7.18.2.3 MGPI0x_vSetPinOutputType()	71
7.18.2.4 MGPI0x_vSetPinOutputSpeed()	72
7.18.2.5 MGPI0x_vSetPinInputPullType()	73
7.18.2.6 MGPI0x_u8GetPinValue()	73
7.18.2.7 MGPI0x_vSetPinValue()	74
7.18.2.8 MGPI0x_vSetResetAtomic()	74
7.18.2.9 MGPI0x_vSetAlternateFunctionON()	75
7.18.2.10 MGPI0x_vInit()	77
7.18.2.11 MGPI0x_vTogglePinValue()	77
7.19 GPIO_program.c	77
7.20 src/COTS/MCAL/RCC/MRCC_config.h File Reference	83
7.20.1 Detailed Description	84
7.20.2 Macro Definition Documentation	84
7.20.2.1 PLLI2S	84
7.20.2.2 PLL	85
7.20.2.3 CSS	85
7.20.2.4 HSEBYP	85
7.20.2.5 HSE_EN	85
7.20.2.6 HSI_EN	85
7.20.2.7 PLLQ	85
7.20.2.8 PLLSRC	86
7.20.2.9 PLLP	86
7.20.2.10 PLLN	86
7.20.2.11 PLLM	86
7.20.2.12 MCO2	86
7.20.2.13 MCO2PRE	86
7.20.2.14 MCO1PRE	87
7.20.2.15 I2SSRC	87
7.20.2.16 MCO1	87
7.20.2.17 RTCPRE	87

7.20.2.18 PPRE2 . . . . .	87
7.20.2.19 PPRE1 . . . . .	87
7.20.2.20 HPRE . . . . .	88
7.20.2.21 SWS . . . . .	88
7.20.2.22 SW . . . . .	88
7.20.2.23 DMA2CLK . . . . .	88
7.20.2.24 DMA1CLK . . . . .	88
7.20.2.25 CRCCLK . . . . .	88
7.20.2.26 GPIOHCLK . . . . .	89
7.20.2.27 GPIOECLK . . . . .	89
7.20.2.28 GPIODCLK . . . . .	89
7.20.2.29 GPIOCCLK . . . . .	89
7.20.2.30 GPIOBCLK . . . . .	89
7.20.2.31 GPIOACLK . . . . .	89
7.20.2.32 OTGFS . . . . .	90
7.20.2.33 PWREN . . . . .	90
7.20.2.34 I2C3EN . . . . .	90
7.20.2.35 I2C2EN . . . . .	90
7.20.2.36 I2C1EN . . . . .	90
7.20.2.37 USART2EN . . . . .	90
7.20.2.38 SPI3EN . . . . .	91
7.20.2.39 SPI2EN . . . . .	91
7.20.2.40 WWDGEN . . . . .	91
7.20.2.41 TIM5EN . . . . .	91
7.20.2.42 TIM4EN . . . . .	91
7.20.2.43 TIM3EN . . . . .	91
7.20.2.44 TIM2EN . . . . .	92
7.20.2.45 TIM11EN . . . . .	92
7.20.2.46 TIM10EN . . . . .	92
7.20.2.47 TIM9EN . . . . .	92
7.20.2.48 SYSCFGEN . . . . .	92
7.20.2.49 SPI4EN . . . . .	92
7.20.2.50 SPI1EN . . . . .	93
7.20.2.51 SDIOEN . . . . .	93
7.20.2.52 ADC1EN . . . . .	93
7.20.2.53 USART6EN . . . . .	93
7.20.2.54 USART1EN . . . . .	93
7.20.2.55 TIM1EN . . . . .	93
7.21 MRCC_config.h . . . . .	94
7.22 src/COTS/MCAL/RCC/MRCC_interface.h File Reference . . . . .	101
7.22.1 Detailed Description . . . . .	103
7.22.2 Macro Definition Documentation . . . . .	103

7.22.2.1	RCC_AHB1	103
7.22.2.2	RCC_AHB2	103
7.22.2.3	RCC_APB1	103
7.22.2.4	RCC_APB2	104
7.22.2.5	RCC_AHB1LPENR	104
7.22.2.6	AHB1ENR_DMA2EN	104
7.22.2.7	AHB1ENR_DMA1EN	104
7.22.2.8	AHB1ENR_CRCEN	104
7.22.2.9	AHB1ENR_GPIOHEN	104
7.22.2.10	AHB1ENR_GPIOEEN	105
7.22.2.11	AHB1ENR_GPIODEN	105
7.22.2.12	AHB1ENR_GPIOCEN	105
7.22.2.13	AHB1ENR_GPIOBEN	105
7.22.2.14	AHB1ENR_GPIOAEN	105
7.22.2.15	AHB2ENR_OTGFSEN	105
7.22.2.16	APB1ENR_PWREN	106
7.22.2.17	APB1ENR_I2C3EN	106
7.22.2.18	APB1ENR_I2C2EN	106
7.22.2.19	APB1ENR_I2C1EN	106
7.22.2.20	APB1ENR_USART2EN	106
7.22.2.21	APB1ENR_SPI3EN	106
7.22.2.22	APB1ENR_SPI2EN	107
7.22.2.23	APB1ENR_WWDGEN	107
7.22.2.24	APB1ENR_TIM5EN	107
7.22.2.25	APB1ENR_TIM4EN	107
7.22.2.26	APB1ENR_TIM3EN	107
7.22.2.27	APB1ENR_TIM2EN	107
7.22.2.28	APB2ENR_TIM11EN	108
7.22.2.29	APB2ENR_TIM10EN	108
7.22.2.30	APB2ENR_TIM9EN	108
7.22.2.31	APB2ENR_SYSCFGEN	108
7.22.2.32	APB2ENR_SPI4EN	108
7.22.2.33	APB2ENR_SPI1EN	108
7.22.2.34	APB2ENR_SDIOEN	109
7.22.2.35	APB2ENR_ADC1EN	109
7.22.2.36	APB2ENR_USART6EN	109
7.22.2.37	APB2ENR_USART1EN	109
7.22.2.38	APB2ENR_TIM1EN	109
7.22.2.39	AHB1LPENR_FLITFLPEN	109
7.22.3	Function Documentation	110
7.22.3.1	MRCC_vInit()	110
7.22.3.2	MRCC_vEnablePeripheralCLK()	114

7.22.3.3 MRCC_vDisablePeripheralCLK()	115
7.23 MRCC_interface.h	116
7.24 src/COTS/MCAL/RCC/MRCC_private.h File Reference	116
7.24.1 Detailed Description	121
7.24.2 Macro Definition Documentation	121
7.24.2.1 RCC_BASE_ADDRESS	121
7.24.2.2 RCC	121
7.24.2.3 RCC_CR_PLLI2SRDY	122
7.24.2.4 RCC_CR_PLLI2SON	122
7.24.2.5 RCC_CR_PLLRDY	122
7.24.2.6 RCC_CR_PLLON	122
7.24.2.7 RCC_CR_CSSON	122
7.24.2.8 RCC_CR_HSEBYP	122
7.24.2.9 RCC_CR_HSERDY	123
7.24.2.10 RCC_CR_HSEON	123
7.24.2.11 RCC_CR_HSIRDY	123
7.24.2.12 RCC_CR_HSION	123
7.24.2.13 RCC_PLLCFGR_PLLQ_b0	123
7.24.2.14 RCC_PLLCFGR_PLLQ_b1	123
7.24.2.15 RCC_PLLCFGR_PLLQ_b2	124
7.24.2.16 RCC_PLLCFGR_PLLQ_b3	124
7.24.2.17 RCC_PLLCFGR_PLLSRC	124
7.24.2.18 RCC_PLLCFGR_PLLP_b0	124
7.24.2.19 RCC_PLLCFGR_PLLP_b1	124
7.24.2.20 RCC_PLLCFGR_PLLN_b0	124
7.24.2.21 RCC_PLLCFGR_PLLN_b1	125
7.24.2.22 RCC_PLLCFGR_PLLN_b2	125
7.24.2.23 RCC_PLLCFGR_PLLN_b3	125
7.24.2.24 RCC_PLLCFGR_PLLN_b4	125
7.24.2.25 RCC_PLLCFGR_PLLN_b5	125
7.24.2.26 RCC_PLLCFGR_PLLN_b6	125
7.24.2.27 RCC_PLLCFGR_PLLN_b7	126
7.24.2.28 RCC_PLLCFGR_PLLN_b8	126
7.24.2.29 RCC_PLLCFGR_PLLM_b0	126
7.24.2.30 RCC_PLLCFGR_PLLM_b1	126
7.24.2.31 RCC_PLLCFGR_PLLM_b2	126
7.24.2.32 RCC_PLLCFGR_PLLM_b3	126
7.24.2.33 RCC_PLLCFGR_PLLM_b4	127
7.24.2.34 RCC_PLLCFGR_PLLM_b5	127
7.24.2.35 RCC_CFGR_MOC2_b0	127
7.24.2.36 RCC_CFGR_MOC2_b1	127
7.24.2.37 RCC_CFGR_MOC2PRE_b0	127

7.24.2.38 RCC_CFGR_MOC2PRE_b1 . . . . .	127
7.24.2.39 RCC_CFGR_MOC2PRE_b2 . . . . .	128
7.24.2.40 RCC_CFGR_MOC1PRE_b0 . . . . .	128
7.24.2.41 RCC_CFGR_MOC1PRE_b1 . . . . .	128
7.24.2.42 RCC_CFGR_MOC1PRE_b2 . . . . .	128
7.24.2.43 RCC_CFGR_I2SSRC . . . . .	128
7.24.2.44 RCC_CFGR_MOC1_b0 . . . . .	128
7.24.2.45 RCC_CFGR_MOC1_b1 . . . . .	129
7.24.2.46 RCC_CFGR_RTCPRE_b0 . . . . .	129
7.24.2.47 RCC_CFGR_RTCPRE_b1 . . . . .	129
7.24.2.48 RCC_CFGR_RTCPRE_b2 . . . . .	129
7.24.2.49 RCC_CFGR_RTCPRE_b3 . . . . .	129
7.24.2.50 RCC_CFGR_RTCPRE_b4 . . . . .	129
7.24.2.51 RCC_CFGR_PPRE2_b0 . . . . .	130
7.24.2.52 RCC_CFGR_PPRE2_b1 . . . . .	130
7.24.2.53 RCC_CFGR_PPRE2_b2 . . . . .	130
7.24.2.54 RCC_CFGR_PPRE1_b0 . . . . .	130
7.24.2.55 RCC_CFGR_PPRE1_b1 . . . . .	130
7.24.2.56 RCC_CFGR_PPRE1_b2 . . . . .	130
7.24.2.57 RCC_CFGR_HPRE_b0 . . . . .	131
7.24.2.58 RCC_CFGR_HPRE_b1 . . . . .	131
7.24.2.59 RCC_CFGR_HPRE_b2 . . . . .	131
7.24.2.60 RCC_CFGR_HPRE_b3 . . . . .	131
7.24.2.61 RCC_CFGR_SWS_b0 . . . . .	131
7.24.2.62 RCC_CFGR_SWS_b1 . . . . .	131
7.24.2.63 RCC_CFGR_SW_b0 . . . . .	132
7.24.2.64 RCC_CFGR_SW_b1 . . . . .	132
7.24.2.65 RCC_AHB1ENR_DMA2EN . . . . .	132
7.24.2.66 RCC_AHB1ENR_DMA1EN . . . . .	132
7.24.2.67 RCC_AHB1ENR_CRCEN . . . . .	132
7.24.2.68 RCC_AHB1ENR_GPIOHEN . . . . .	132
7.24.2.69 RCC_AHB1ENR_GPIOEEN . . . . .	133
7.24.2.70 RCC_AHB1ENR_GPIODEN . . . . .	133
7.24.2.71 RCC_AHB1ENR_GPIOCEN . . . . .	133
7.24.2.72 RCC_AHB1ENR_GPIOBEN . . . . .	133
7.24.2.73 RCC_AHB1ENR_GPIOAEN . . . . .	133
7.24.2.74 RCC_AHB2ENR_OTGFSEN . . . . .	133
7.24.2.75 RCC_APB1ENR_PWREN . . . . .	134
7.24.2.76 RCC_APB1ENR_I2C3EN . . . . .	134
7.24.2.77 RCC_APB1ENR_I2C2EN . . . . .	134
7.24.2.78 RCC_APB1ENR_I2C1EN . . . . .	134
7.24.2.79 RCC_APB1ENR_USART2EN . . . . .	134

7.24.2.80 RCC_APB1ENR_SPI3EN . . . . .	134
7.24.2.81 RCC_APB1ENR_SPI2EN . . . . .	135
7.24.2.82 RCC_APB1ENR_WWDGEN . . . . .	135
7.24.2.83 RCC_APB1ENR_TIM5EN . . . . .	135
7.24.2.84 RCC_APB1ENR_TIM4EN . . . . .	135
7.24.2.85 RCC_APB1ENR_TIM3EN . . . . .	135
7.24.2.86 RCC_APB1ENR_TIM2EN . . . . .	135
7.24.2.87 RCC_APB2ENR_TIM11EN . . . . .	136
7.24.2.88 RCC_APB2ENR_TIM10EN . . . . .	136
7.24.2.89 RCC_APB2ENR_TIM9EN . . . . .	136
7.24.2.90 RCC_APB2ENR_SYSCFGEN . . . . .	136
7.24.2.91 RCC_APB2ENR_SPI4EN . . . . .	136
7.24.2.92 RCC_APB2ENR_SPI1EN . . . . .	136
7.24.2.93 RCC_APB2ENR_SDIOEN . . . . .	137
7.24.2.94 RCC_APB2ENR_ADC1EN . . . . .	137
7.24.2.95 RCC_APB2ENR_USART6EN . . . . .	137
7.24.2.96 RCC_APB2ENR_USART1EN . . . . .	137
7.24.2.97 RCC_APB2ENR_TIM1EN . . . . .	137
7.24.2.98 RCC_AHB1LPENR_FLITFLPEN . . . . .	137
7.24.2.99 ENABLE . . . . .	138
7.24.2.100 DISABLE . . . . .	138
7.24.2.101 BYBASED . . . . .	138
7.24.2.102 NOTBYBASED . . . . .	138
7.24.2.103 SYSCLK . . . . .	138
7.24.2.104 PLLI2SCLK . . . . .	138
7.24.2.105 HSE . . . . .	139
7.24.2.106 PLLCLK [1/2] . . . . .	139
7.24.2.107 NoDivision . . . . .	139
7.24.2.108 DivisionBy2 . . . . .	139
7.24.2.109 DivisionBy3 . . . . .	139
7.24.2.110 DivisionBy4 . . . . .	139
7.24.2.111 DivisionBy5 . . . . .	140
7.24.2.112 I2S_CKIN . . . . .	140
7.24.2.113 HSI . . . . .	140
7.24.2.114 LSE . . . . .	140
7.24.2.115 PLLCLK [2/2] . . . . .	140
7.24.2.116 SYSCLKby2 . . . . .	140
7.24.2.117 SYSCLKby4 . . . . .	141
7.24.2.118 SYSCLKby8 . . . . .	141
7.24.2.119 SYSCLKby16 . . . . .	141
7.24.2.120 SYSCLKby64 . . . . .	141
7.24.2.121 SYSCLKby128 . . . . .	141

7.24.2.122 SYSCLKby256 . . . . .	141
7.24.2.123 SYSCLKby512 . . . . .	142
7.24.2.124 NoCLK0 . . . . .	142
7.24.2.125 NoCLK1 . . . . .	142
7.24.2.126 HSEby2 . . . . .	142
7.24.2.127 HSEby3 . . . . .	142
7.24.2.128 HSEby4 . . . . .	142
7.24.2.129 HSEby5 . . . . .	143
7.24.2.130 HSEby6 . . . . .	143
7.24.2.131 HSEby7 . . . . .	143
7.24.2.132 HSEby8 . . . . .	143
7.24.2.133 HSEby9 . . . . .	143
7.24.2.134 HSEby10 . . . . .	143
7.24.2.135 HSEby11 . . . . .	144
7.24.2.136 HSEby12 . . . . .	144
7.24.2.137 HSEby13 . . . . .	144
7.24.2.138 HSEby14 . . . . .	144
7.24.2.139 HSEby15 . . . . .	144
7.24.2.140 HSEby16 . . . . .	144
7.24.2.141 HSEby17 . . . . .	145
7.24.2.142 HSEby18 . . . . .	145
7.24.2.143 HSEby19 . . . . .	145
7.24.2.144 HSEby20 . . . . .	145
7.24.2.145 HSEby21 . . . . .	145
7.24.2.146 HSEby22 . . . . .	145
7.24.2.147 HSEby23 . . . . .	146
7.24.2.148 HSEby24 . . . . .	146
7.24.2.149 HSEby25 . . . . .	146
7.24.2.150 HSEby26 . . . . .	146
7.24.2.151 HSEby27 . . . . .	146
7.24.2.152 HSEby28 . . . . .	146
7.24.2.153 HSEby29 . . . . .	147
7.24.2.154 HSEby30 . . . . .	147
7.24.2.155 HSEby31 . . . . .	147
7.24.2.156 AHBby2 . . . . .	147
7.24.2.157 AHBby4 . . . . .	147
7.24.2.158 AHBby8 . . . . .	147
7.24.2.159 AHBby16 . . . . .	148
7.24.2.160 Equal_0 . . . . .	148
7.24.2.161 Equal_1 . . . . .	148
7.24.2.162 Equal_2 . . . . .	148
7.24.2.163 Equal_3 . . . . .	148

7.24.2.164 Equal_4 . . . . .	148
7.24.2.165 Equal_5 . . . . .	149
7.24.2.166 Equal_6 . . . . .	149
7.24.2.167 Equal_7 . . . . .	149
7.24.2.168 Equal_8 . . . . .	149
7.24.2.169 Equal_9 . . . . .	149
7.24.2.170 Equal_10 . . . . .	149
7.24.2.171 Equal_11 . . . . .	150
7.24.2.172 Equal_12 . . . . .	150
7.24.2.173 Equal_13 . . . . .	150
7.24.2.174 Equal_14 . . . . .	150
7.24.2.175 Equal_15 . . . . .	150
7.24.2.176 Equal_16 . . . . .	150
7.24.2.177 Equal_17 . . . . .	151
7.24.2.178 Equal_18 . . . . .	151
7.24.2.179 Equal_19 . . . . .	151
7.24.2.180 Equal_20 . . . . .	151
7.24.2.181 Equal_21 . . . . .	151
7.24.2.182 Equal_22 . . . . .	151
7.24.2.183 Equal_23 . . . . .	152
7.24.2.184 Equal_24 . . . . .	152
7.24.2.185 Equal_25 . . . . .	152
7.24.2.186 Equal_26 . . . . .	152
7.24.2.187 Equal_27 . . . . .	152
7.24.2.188 Equal_28 . . . . .	152
7.24.2.189 Equal_29 . . . . .	153
7.24.2.190 Equal_30 . . . . .	153
7.24.2.191 Equal_31 . . . . .	153
7.24.2.192 Equal_32 . . . . .	153
7.24.2.193 Equal_33 . . . . .	153
7.24.2.194 Equal_34 . . . . .	153
7.24.2.195 Equal_35 . . . . .	154
7.24.2.196 Equal_36 . . . . .	154
7.24.2.197 Equal_37 . . . . .	154
7.24.2.198 Equal_38 . . . . .	154
7.24.2.199 Equal_39 . . . . .	154
7.24.2.200 Equal_40 . . . . .	154
7.24.2.201 Equal_41 . . . . .	155
7.24.2.202 Equal_42 . . . . .	155
7.24.2.203 Equal_43 . . . . .	155
7.24.2.204 Equal_44 . . . . .	155
7.24.2.205 Equal_45 . . . . .	155



7.24.2.206 Equal_46 . . . . .	155
7.24.2.207 Equal_47 . . . . .	156
7.24.2.208 Equal_48 . . . . .	156
7.24.2.209 Equal_49 . . . . .	156
7.24.2.210 Equal_50 . . . . .	156
7.24.2.211 Equal_51 . . . . .	156
7.24.2.212 Equal_52 . . . . .	156
7.24.2.213 Equal_53 . . . . .	157
7.24.2.214 Equal_54 . . . . .	157
7.24.2.215 Equal_55 . . . . .	157
7.24.2.216 Equal_56 . . . . .	157
7.24.2.217 Equal_57 . . . . .	157
7.24.2.218 Equal_58 . . . . .	157
7.24.2.219 Equal_59 . . . . .	158
7.24.2.220 Equal_60 . . . . .	158
7.24.2.221 Equal_61 . . . . .	158
7.24.2.222 Equal_62 . . . . .	158
7.24.2.223 Equal_63 . . . . .	158
7.25 MRCC_private.h . . . . .	159
7.26 src/COTS/MCAL/RCC/MRCC_program.c File Reference . . . . .	163
7.26.1 Detailed Description . . . . .	163
7.26.2 Function Documentation . . . . .	163
7.26.2.1 MRCC_vInit() . . . . .	164
7.26.2.2 MRCC_vEnablePeriphralCLK() . . . . .	168
7.26.2.3 MRCC_vDisablePeriphralCLK() . . . . .	169
7.27 MRCC_program.c . . . . .	169
<b>Index</b>	<b>177</b>



# Chapter 1

## ADAS Graudation Project

### 1.1 Description

The graduation project is ADAS and consists of 3 systems:

1. Adaptive Cruise Control
2. Adaptive Light Control
3. Driver Health Care

#### 1.1.1 Adaptive Cruise Control

Adaptive cruise control (ACC) is an enhancement of conventional cruise control. ACC automatically adjusts the speed of your car to match the speed of the car in front of you. If the car ahead slows down, ACC can automatically match it.

#### 1.1.2 Adaptive Light Control

Adaptive light control is a system that changes the brightness of the car's headlights automatically based on the environment's light as well as the cars that are coming in the opposite way. The lighter the environment, the darker the headlights.

#### 1.1.3 Driver Health Care

Driver health care is a system that is responsible for the detection of the driver's sleepiness and give the suitable alters to warn the driver as well as the other cars in real-time using Image Processing and Machine Learning techniques.

### 1.2 Hardware specifications

This project is developed using:

- Micro controller: STM32F401CCU6



## Chapter 2

# Module Index

### 2.1 Modules

Here is a list of all modules:

Bit Manipulation Math Macros . . . . .	9
Bit Group Manipulation Math Macros . . . . .	10
Compiler standard macros . . . . .	12
Utilities macros . . . . .	15
Standard types . . . . .	15
Standard values . . . . .	18
GPIO Modes . . . . .	20
GPIO Ports . . . . .	21
GPIO Output Types . . . . .	22
GPIO PIN Speed . . . . .	23
GPIO Pull Types . . . . .	24
GPIO Output Values . . . . .	25
GPIO Output PINs . . . . .	26
GPIO Alternate Functions . . . . .	30
GPIO Addresses . . . . .	35
GPIO Registers . . . . .	36



## Chapter 3

# Data Structure Index

### 3.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">GPIOx_MemoryMapType</a>	
MDIO Configuration structure for a speific PIN initialization . . . . .	39
<a href="#">MGPIOx_ConfigType</a>	
MDIO Configuration structure for a speific PIN initialization . . . . .	42
<a href="#">RCC_MemoryMapType</a> . . . . .	44





## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

src/COTS/LIB/LSTD_BITMATH.h	
This file contains the bit math manipulation macro-functions . . . . .	51
src/COTS/LIB/LSTD_COMPILER.h	
This file contains the compiler standard macros . . . . .	52
src/COTS/LIB/LSTD_MCU_UTILITIES.h	
This file contains the MCU utility macro-functions . . . . .	53
src/COTS/LIB/LSTD_TYPES.h	
This file contains the standard types . . . . .	54
src/COTS/LIB/LSTD_VALUES.h	
This file contains the standard values . . . . .	55
src/COTS/MCAL/GPIO/GPIO_config.h	
This file contains the GPIO configurations . . . . .	56
src/COTS/MCAL/GPIO/GPIO_interface.h	
This file contains the interfacing information for the GPIO module . . . . .	57
src/COTS/MCAL/GPIO/GPIO_private.h	
This file contains the registers information and addresses for the GPIO module . . . . .	68
src/COTS/MCAL/GPIO/GPIO_program.c	
This file contains the source code of the interfacing for the GPIO modules . . . . .	69
src/COTS/MCAL/RCC/MRCC_config.h	
This file contains the RCC configurations . . . . .	83
src/COTS/MCAL/RCC/MRCC_interface.h	
This file contains the interfacing information for the RCC module . . . . .	101
src/COTS/MCAL/RCC/MRCC_private.h	
This file contains the registers information and addresses for the RCC module . . . . .	116
src/COTS/MCAL/RCC/MRCC_program.c	
This file contains the source code of the interfacing for the RCC module . . . . .	163



## Chapter 5

# Module Documentation

## 5.1 Bit Manipulation Math Macros

### Macros

- #define [SET\\_BIT](#)(Reg, bitnum) (Reg) |= (1 << (bitnum))
- #define [CLR\\_BIT](#)(Reg, bitnum) (Reg) &= ~(1 << (bitnum))
- #define [TOGGLE\\_BIT](#)(Reg, bitnum) (Reg) ^= (1 << (bitnum))
- #define [GET\\_BIT](#)(Reg, bitnum) (((Reg)>>(bitnum)) & 1)

### 5.1.1 Detailed Description

### 5.1.2 Macro Definition Documentation

#### 5.1.2.1 SET\_BIT

```
#define SET_BIT(  
    Reg,  
    bitnum ) (Reg) |= (1 << (bitnum))  
  
#include <src/COTS/LIB/LSTD_BITMATH.h>
```

Sets a certain bit's value

Definition at line 23 of file [LSTD\\_BITMATH.h](#).

### 5.1.2.2 CLR\_BIT

```
#define CLR_BIT(  
    Reg,  
    bitnum ) (Reg) &= ~(1 << (bitnum))
```

```
#include <src/COTS/LIB/LSTD_BITMATH.h>
```

Clears a certain bit's value to

Definition at line 30 of file [LSTD\\_BITMATH.h](#).

### 5.1.2.3 TOGGLE\_BIT

```
#define TOGGLE_BIT(  
    Reg,  
    bitnum ) (Reg) ^= (1 << (bitnum))
```

```
#include <src/COTS/LIB/LSTD_BITMATH.h>
```

Toggle a bit to 0 if it's 1, 1 otherwise

Definition at line 37 of file [LSTD\\_BITMATH.h](#).

### 5.1.2.4 GET\_BIT

```
#define GET_BIT(  
    Reg,  
    bitnum ) (((Reg)>>(bitnum)) & 1)
```

```
#include <src/COTS/LIB/LSTD_BITMATH.h>
```

Returns the value of the bit whether it's 1 or 0

Definition at line 44 of file [LSTD\\_BITMATH.h](#).

## 5.2 Bit Group Manipulation Math Macros

### Macros

- #define [SET\\_BITS](#)(Reg, bits, bitnum, factor) (Reg) |= ((bits) << (bitnum \* factor))
- #define [CLR\\_BITS](#)(Reg, bits, bitnum, factor) (Reg) &= ~(bits) << (bitnum \* factor))
- #define [TOGGLE\\_BITS](#)(Reg, bits, bitnum, factor) (Reg) ^= ((bits) << (bitnum \* factor))
- #define [GET\\_BITS](#)(Reg, bits, bitnum, factor) (((Reg) >> (bitnum \* factor)) & (bits))

## 5.2.1 Detailed Description

## 5.2.2 Macro Definition Documentation

### 5.2.2.1 SET\_BITS

```
#define SET_BITS(  
    Reg,  
    bits,  
    bitnum,  
    factor ) (Reg) |= ((bits) << (bitnum * factor))
```

```
#include <src/COTS/LIB/LSTD_BITMATH.h>
```

Sets the values of a group of bits

Definition at line 59 of file [LSTD\\_BITMATH.h](#).

### 5.2.2.2 CLR\_BITS

```
#define CLR_BITS(  
    Reg,  
    bits,  
    bitnum,  
    factor ) (Reg) &= ~((bits) << (bitnum * factor))
```

```
#include <src/COTS/LIB/LSTD_BITMATH.h>
```

Clears the value of a group of bits

Definition at line 66 of file [LSTD\\_BITMATH.h](#).

### 5.2.2.3 TOGGLE\_BITS

```
#define TOGGLE_BITS(  
    Reg,  
    bits,  
    bitnum,  
    factor ) (Reg) ^= ((bits) << (bitnum * factor))
```

```
#include <src/COTS/LIB/LSTD_BITMATH.h>
```

Toggles the value of a group of bits

Definition at line 73 of file [LSTD\\_BITMATH.h](#).

#### 5.2.2.4 GET\_BITS

```
#define GET_BITS(  
    Reg,  
    bits,  
    bitnum,  
    factor ) (((Reg) >> (bitnum * factor)) & (bits))  
  
#include <src/COTS/LIB/LSTD_BITMATH.h>
```

Returns the value of a group of bits

Definition at line 80 of file [LSTD\\_BITMATH.h](#).

### 5.3 Compiler standard macros

#### Macros

- #define [VAR](#)(vartype) vartype
- #define [FUNC](#)(rettype) rettype
- #define [P2VAR](#)(ptrtype) ptrtype \*
- #define [P2CONST](#)(ptrtype) const ptrtype \*
- #define [CONSTP2VAR](#)(ptrtype) ptrtype \* const
- #define [CONSTP2CONST](#)(ptrtype) const ptrtype \* const
- #define [P2FUNC](#)(rettype, fctname) rettype (\* fctname)
- #define [CONST](#)(consttype) const consttype
- #define [STATIC](#) static

#### 5.3.1 Detailed Description

#### 5.3.2 Macro Definition Documentation

##### 5.3.2.1 VAR

```
#define VAR(  
    vartype ) vartype  
  
#include <src/COTS/LIB/LSTD_COMPILER.h>
```

Declare a variable with the specified type

Definition at line 23 of file [LSTD\\_COMPILER.h](#).

### 5.3.2.2 FUNC

```
#define FUNC(  
    rettype ) rettype  
  
#include <src/COTS/LIB/LSTD_COMPILER.h>
```

Declare a function's return type

Definition at line 30 of file [LSTD\\_COMPILER.h](#).

### 5.3.2.3 P2VAR

```
#define P2VAR(  
    ptrtype ) ptrtype *  
  
#include <src/COTS/LIB/LSTD_COMPILER.h>
```

Declare a pointer-to-variable with the specified type

Definition at line 37 of file [LSTD\\_COMPILER.h](#).

### 5.3.2.4 P2CONST

```
#define P2CONST(  
    ptrtype ) const ptrtype *  
  
#include <src/COTS/LIB/LSTD_COMPILER.h>
```

Declare a constant pointer-to-variable with the specified type

Definition at line 44 of file [LSTD\\_COMPILER.h](#).

### 5.3.2.5 CONSTP2VAR

```
#define CONSTP2VAR(  
    ptrtype ) ptrtype * const  
  
#include <src/COTS/LIB/LSTD_COMPILER.h>
```

Declare a pointer-to-variable constant with the specified type

Definition at line 51 of file [LSTD\\_COMPILER.h](#).

### 5.3.2.6 CONSTP2CONST

```
#define CONSTP2CONST(  
    ptrtype ) const ptrtype * const  
  
#include <src/COTS/LIB/LSTD_COMPILER.h>
```

Declare a constant pointer-to-variable constant with the specified type

Definition at line 58 of file [LSTD\\_COMPILER.h](#).

### 5.3.2.7 P2FUNC

```
#define P2FUNC(  
    rettype,  
    fctname ) rettype (* fctname)  
  
#include <src/COTS/LIB/LSTD_COMPILER.h>
```

Declare a pointer-to-function with the specified type

Definition at line 65 of file [LSTD\\_COMPILER.h](#).

### 5.3.2.8 CONST

```
#define CONST(  
    consttype ) const consttype  
  
#include <src/COTS/LIB/LSTD_COMPILER.h>
```

Declare a standard constant variable with the specified type

Definition at line 73 of file [LSTD\\_COMPILER.h](#).

### 5.3.2.9 STATIC

```
#define STATIC static  
  
#include <src/COTS/LIB/LSTD_COMPILER.h>
```

Declare a standard static variable

Definition at line 82 of file [LSTD\\_COMPILER.h](#).



## 5.4 Utilities macros

### Macros

- `#define MY_MS_DELAY(T) do{ u32 Timer = (T * 500); while (Timer--) { asm ("nop"); } } while(0);`

#### 5.4.1 Detailed Description

#### 5.4.2 Macro Definition Documentation

##### 5.4.2.1 MY\_MS\_DELAY

```
#define MY_MS_DELAY(  
    T ) do{ u32 Timer = (T * 500); while (Timer--) { asm ("nop"); } } while(0);
```

```
#include <src/COTS/LIB/LSTD_MCU_UTILITIES.h>
```

Declare a delay with a `T` microseconds with a NOP delay

Definition at line 23 of file [LSTD\\_MCU\\_UTILITIES.h](#).

## 5.5 Standard types

### Typedefs

- typedef unsigned char [bool\\_t](#)
- typedef unsigned char [u8\\_t](#)
- typedef unsigned short int [u16\\_t](#)
- typedef unsigned long int [u32\\_t](#)
- typedef signed char [s8\\_t](#)
- typedef signed short int [s16\\_t](#)
- typedef signed long int [s32\\_t](#)
- typedef float [fl32\\_t](#)
- typedef double [fl64\\_t](#)

#### 5.5.1 Detailed Description

#### 5.5.2 Typedef Documentation

#### 5.5.2.1 bool\_t

bool\_t

```
#include <src/COTS/LIB/LSTD_TYPES.h>
```

Type definition for boolean

Definition at line 23 of file [LSTD\\_TYPES.h](#).

#### 5.5.2.2 u8\_t

u8\_t

```
#include <src/COTS/LIB/LSTD_TYPES.h>
```

Type definition for 8-bit unsigned INT

Definition at line 30 of file [LSTD\\_TYPES.h](#).

#### 5.5.2.3 u16\_t

u16\_t

```
#include <src/COTS/LIB/LSTD_TYPES.h>
```

Type definition for 16-bit unsigned INT

Definition at line 37 of file [LSTD\\_TYPES.h](#).

#### 5.5.2.4 u32\_t

u32\_t

```
#include <src/COTS/LIB/LSTD_TYPES.h>
```

Type definition for 32-bit unsigned INT

Definition at line 44 of file [LSTD\\_TYPES.h](#).

#### 5.5.2.5 s8\_t

s8\_t

```
#include <src/COTS/LIB/LSTD_TYPES.h>
```

Type definition for 8-bit signed INT

Definition at line 51 of file [LSTD\\_TYPES.h](#).

#### 5.5.2.6 s16\_t

s16\_t

```
#include <src/COTS/LIB/LSTD_TYPES.h>
```

Type definition for 16-bit signed INT

Definition at line 58 of file [LSTD\\_TYPES.h](#).

#### 5.5.2.7 s32\_t

s32\_t

```
#include <src/COTS/LIB/LSTD_TYPES.h>
```

Type definition for 32-bit signed INT

Definition at line 65 of file [LSTD\\_TYPES.h](#).

#### 5.5.2.8 fl32\_t

fl32\_t

```
#include <src/COTS/LIB/LSTD_TYPES.h>
```

Type definition for 32-bit float

Definition at line 72 of file [LSTD\\_TYPES.h](#).

### 5.5.2.9 fl64\_t

fl64\_t

```
#include <src/COTS/LIB/LSTD_TYPES.h>
```

Type definition for 64-bit float

Definition at line 79 of file LSTD\_TYPES.h.

## 5.6 Standard values

### Macros

- #define TRUE (1)
- #define FALSE (0)
- #define NULL (P2VAR(void) 0)
- #define INITIAL\_ZERO (0)
- #define FLAG\_SET (1)
- #define FLAG\_CLEARED (0)

### 5.6.1 Detailed Description

### 5.6.2 Macro Definition Documentation

#### 5.6.2.1 TRUE

```
#define TRUE (1)
```

```
#include <src/COTS/LIB/LSTD_VALUES.h>
```

Type definition for TRUE

Definition at line 24 of file LSTD\_VALUES.h.

#### 5.6.2.2 FALSE

```
#define FALSE (0)
```

```
#include <src/COTS/LIB/LSTD_VALUES.h>
```

Type definition for FALSE

Definition at line 33 of file LSTD\_VALUES.h.

### 5.6.2.3 NULL

```
#define NULL (P2VAR(void) 0)
```

```
#include <src/COTS/LIB/LSTD_VALUES.h>
```

Type definition for NULL

Definition at line 42 of file [LSTD\\_VALUES.h](#).

### 5.6.2.4 INITIAL\_ZERO

```
#define INITIAL_ZERO (0)
```

```
#include <src/COTS/LIB/LSTD_VALUES.h>
```

Type definition for INITIAL\_ZERO

Definition at line 51 of file [LSTD\\_VALUES.h](#).

### 5.6.2.5 FLAG\_SET

```
#define FLAG_SET (1)
```

```
#include <src/COTS/LIB/LSTD_VALUES.h>
```

Type definition for FLAG\_SET

Definition at line 60 of file [LSTD\\_VALUES.h](#).

### 5.6.2.6 FLAG\_CLEARED

```
#define FLAG_CLEARED (0)
```

```
#include <src/COTS/LIB/LSTD_VALUES.h>
```

Type definition for FLAG\_CLEARED

Definition at line 69 of file [LSTD\\_VALUES.h](#).

## 5.7 GPIO Modes

### Macros

- `#define GPIOx_MODE_INPUT (0b00)`  
*Control input mode.*
- `#define GPIOx_MODE_OUTPUT (0b01)`  
*Control output mode.*
- `#define GPIOx_MODE_AF (0b10)`  
*Control alternate function mode.*
- `#define GPIOx_MODE_ANALOG (0b11)`  
*Control analog mode.*

### 5.7.1 Detailed Description

### 5.7.2 Macro Definition Documentation

#### 5.7.2.1 GPIOx\_MODE\_INPUT

```
#define GPIOx_MODE_INPUT (0b00)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

Control input mode.

Definition at line 158 of file [GPIO\\_interface.h](#).

#### 5.7.2.2 GPIOx\_MODE\_OUTPUT

```
#define GPIOx_MODE_OUTPUT (0b01)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

Control output mode.

Definition at line 165 of file [GPIO\\_interface.h](#).

#### 5.7.2.3 GPIOx\_MODE\_AF

```
#define GPIOx_MODE_AF (0b10)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

Control alternate function mode.

Definition at line 172 of file [GPIO\\_interface.h](#).

#### 5.7.2.4 GPIOx\_MODE\_ANALOG

```
#define GPIOx_MODE_ANALOG (0b11)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

Control analog mode.

Definition at line 179 of file [GPIO\\_interface.h](#).

## 5.8 GPIO Ports

### Macros

- #define [GPIO\\_PORTA](#) (0)  
*GPIO Port A.*
- #define [GPIO\\_PORTB](#) (1)  
*GPIO Port B.*
- #define [GPIO\\_PORTC](#) (2)  
*GPIO Port C.*

### 5.8.1 Detailed Description

### 5.8.2 Macro Definition Documentation

#### 5.8.2.1 GPIO\_PORTA

```
#define GPIO_PORTA (0)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Port A.

Definition at line 194 of file [GPIO\\_interface.h](#).

#### 5.8.2.2 GPIO\_PORTB

```
#define GPIO_PORTB (1)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Port B.

Definition at line 201 of file [GPIO\\_interface.h](#).

### 5.8.2.3 GPIO\_PORTC

```
#define GPIO_PORTC (2)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Port C.

Definition at line 208 of file [GPIO\\_interface.h](#).

## 5.9 GPIO Output Types

### Macros

- #define [GPIOx\\_OPENDRAIN](#) (1)  
*GPIO open-drain.*
- #define [GPIOx\\_PUSHPULL](#) (2)  
*GPIO push-pull.*

### 5.9.1 Detailed Description

### 5.9.2 Macro Definition Documentation

#### 5.9.2.1 GPIOx\_OPENDRAIN

```
#define GPIOx_OPENDRAIN (1)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO open-drain.

Definition at line 223 of file [GPIO\\_interface.h](#).

#### 5.9.2.2 GPIOx\_PUSHPULL

```
#define GPIOx_PUSHPULL (2)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO push-pull.

Definition at line 230 of file [GPIO\\_interface.h](#).



## 5.10 GPIO PIN Speed

### Macros

- `#define GPIOx_LowSpeed (0b00)`  
*GPIO low speed.*
- `#define GPIOx_MediumSpeed (0b01)`  
*GPIO medium speed.*
- `#define GPIOx_HighSpeed (0b10)`  
*GPIO high speed.*
- `#define GPIOx_VeryHighSpeed (0b11)`  
*GPIO very high speed.*

#### 5.10.1 Detailed Description

#### 5.10.2 Macro Definition Documentation

##### 5.10.2.1 GPIOx\_LowSpeed

```
#define GPIOx_LowSpeed (0b00)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO low speed.

Definition at line 245 of file [GPIO\\_interface.h](#).

##### 5.10.2.2 GPIOx\_MediumSpeed

```
#define GPIOx_MediumSpeed (0b01)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO medium speed.

Definition at line 252 of file [GPIO\\_interface.h](#).

##### 5.10.2.3 GPIOx\_HighSpeed

```
#define GPIOx_HighSpeed (0b10)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO high speed.

Definition at line 259 of file [GPIO\\_interface.h](#).

#### 5.10.2.4 GPIOx\_VeryHighSpeed

```
#define GPIOx_VeryHighSpeed (0b11)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO very high speed.

Definition at line 266 of file [GPIO\\_interface.h](#).

### 5.11 GPIO Pull Types

#### Macros

- #define [GPIOx\\_NoPull](#) (0b00)  
*GPIO No PULL.*
- #define [GPIOx\\_PullUp](#) (0b01)  
*GPIO Pull UP.*
- #define [GPIOx\\_PullDown](#) (0b10)  
*GPIO Pull Down.*

#### 5.11.1 Detailed Description

#### 5.11.2 Macro Definition Documentation

##### 5.11.2.1 GPIOx\_NoPull

```
#define GPIOx_NoPull (0b00)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO No PULL.

Definition at line 281 of file [GPIO\\_interface.h](#).

##### 5.11.2.2 GPIOx\_PullUp

```
#define GPIOx_PullUp (0b01)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Pull UP.

Definition at line 288 of file [GPIO\\_interface.h](#).

### 5.11.2.3 GPIOx\_PullDown

```
#define GPIOx_PullDown (0b10)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Pull Down.

Definition at line 295 of file [GPIO\\_interface.h](#).

## 5.12 GPIO Output Values

### Macros

- #define [GPIOx\\_HIGH](#) (1)  
*GPIO output high.*
- #define [GPIOx\\_LOW](#) (2)  
*GPIO output low.*

### 5.12.1 Detailed Description

### 5.12.2 Macro Definition Documentation

#### 5.12.2.1 GPIOx\_HIGH

```
#define GPIOx_HIGH (1)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO output high.

Definition at line 310 of file [GPIO\\_interface.h](#).

#### 5.12.2.2 GPIOx\_LOW

```
#define GPIOx_LOW (2)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO output low.

Definition at line 317 of file [GPIO\\_interface.h](#).

## 5.13 GPIO Output PINs

### Macros

- `#define GPIOx_PIN0` (0)  
*GPIO PIN 0.*
- `#define GPIOx_PIN1` (1)  
*GPIO PIN 1.*
- `#define GPIOx_PIN2` (2)  
*GPIO PIN 2.*
- `#define GPIOx_PIN3` (3)  
*GPIO PIN 3.*
- `#define GPIOx_PIN4` (4)  
*GPIO PIN 4.*
- `#define GPIOx_PIN5` (5)  
*GPIO PIN 5.*
- `#define GPIOx_PIN6` (6)  
*GPIO PIN 6.*
- `#define GPIOx_PIN7` (7)  
*GPIO PIN 7.*
- `#define GPIOx_PIN8` (8)  
*GPIO PIN 8.*
- `#define GPIOx_PIN9` (9)  
*brief GPIO PIN 9*
- `#define GPIOx_PIN10` (10)  
*GPIO PIN 10.*
- `#define GPIOx_PIN11` (11)  
*GPIO PIN 11.*
- `#define GPIOx_PIN12` (12)  
*GPIO PIN 12.*
- `#define GPIOx_PIN13` (13)  
*GPIO PIN 13.*
- `#define GPIOx_PIN14` (14)  
*GPIO PIN 14.*
- `#define GPIOx_PIN15` (15)  
*GPIO PIN 15.*

### 5.13.1 Detailed Description

### 5.13.2 Macro Definition Documentation

#### 5.13.2.1 GPIOx\_PIN0

```
#define GPIOx_PIN0 (0)
```

`#include <src/COTS/MCAL/GPIO/GPIO_interface.h>`

GPIO PIN 0.

Definition at line 332 of file [GPIO\\_interface.h](#).

### 5.13.2.2 GPIOx\_PIN1

```
#define GPIOx_PIN1 (1)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 1.

Definition at line 339 of file [GPIO\\_interface.h](#).

### 5.13.2.3 GPIOx\_PIN2

```
#define GPIOx_PIN2 (2)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 2.

Definition at line 346 of file [GPIO\\_interface.h](#).

### 5.13.2.4 GPIOx\_PIN3

```
#define GPIOx_PIN3 (3)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 3.

Definition at line 353 of file [GPIO\\_interface.h](#).

### 5.13.2.5 GPIOx\_PIN4

```
#define GPIOx_PIN4 (4)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 4.

Definition at line 360 of file [GPIO\\_interface.h](#).

#### 5.13.2.6 GPIOx\_PIN5

```
#define GPIOx_PIN5 (5)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 5.

Definition at line 367 of file [GPIO\\_interface.h](#).

#### 5.13.2.7 GPIOx\_PIN6

```
#define GPIOx_PIN6 (6)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 6.

Definition at line 374 of file [GPIO\\_interface.h](#).

#### 5.13.2.8 GPIOx\_PIN7

```
#define GPIOx_PIN7 (7)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 7.

Definition at line 381 of file [GPIO\\_interface.h](#).

#### 5.13.2.9 GPIOx\_PIN8

```
#define GPIOx_PIN8 (8)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 8.

Definition at line 388 of file [GPIO\\_interface.h](#).

#### 5.13.2.10 GPIOx\_PIN9

```
#define GPIOx_PIN9 (9)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

brief GPIO PIN 9

Definition at line 395 of file [GPIO\\_interface.h](#).

#### 5.13.2.11 GPIOx\_PIN10

```
#define GPIOx_PIN10 (10)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 10.

Definition at line 402 of file [GPIO\\_interface.h](#).

#### 5.13.2.12 GPIOx\_PIN11

```
#define GPIOx_PIN11 (11)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 11.

Definition at line 409 of file [GPIO\\_interface.h](#).

#### 5.13.2.13 GPIOx\_PIN12

```
#define GPIOx_PIN12 (12)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 12.

Definition at line 416 of file [GPIO\\_interface.h](#).

#### 5.13.2.14 GPIOx\_PIN13

```
#define GPIOx_PIN13 (13)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 13.

Definition at line 423 of file [GPIO\\_interface.h](#).

#### 5.13.2.15 GPIOx\_PIN14

```
#define GPIOx_PIN14 (14)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 14.

Definition at line 430 of file [GPIO\\_interface.h](#).

#### 5.13.2.16 GPIOx\_PIN15

```
#define GPIOx_PIN15 (15)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO PIN 15.

Definition at line 437 of file [GPIO\\_interface.h](#).

## 5.14 GPIO Alternate Functions

### Macros

- `#define GPIOx_AF0 (0)`  
*GPIO Alternate function 0.*
- `#define GPIOx_AF1 (1)`  
*GPIO Alternate function 1.*
- `#define GPIOx_AF2 (2)`  
*GPIO Alternate function 2.*
- `#define GPIOx_AF3 (3)`  
*GPIO Alternate function 3.*
- `#define GPIOx_AF4 (4)`  
*GPIO Alternate function 4.*
- `#define GPIOx_AF5 (5)`  
*GPIO Alternate function 5.*



- `#define GPIOx_AF6` (6)  
*GPIO Alternate function 6.*
- `#define GPIOx_AF7` (7)  
*GPIO Alternate function 7.*
- `#define GPIOx_AF8` (8)  
*GPIO Alternate function 8.*
- `#define GPIOx_AF9` (9)  
*GPIO Alternate function 9.*
- `#define GPIOx_AF10` (10)  
*GPIO Alternate function 10.*
- `#define GPIOx_AF11` (11)  
*GPIO Alternate function 11.*
- `#define GPIOx_AF12` (12)  
*GPIO Alternate function 12.*
- `#define GPIOx_AF13` (13)  
*GPIO Alternate function 13.*
- `#define GPIOx_AF14` (14)  
*GPIO Alternate function 14.*
- `#define GPIOx_AF15` (15)  
*GPIO Alternate function 15.*

### 5.14.1 Detailed Description

### 5.14.2 Macro Definition Documentation

#### 5.14.2.1 GPIOx\_AF0

```
#define GPIOx_AF0 (0)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 0.

Definition at line 452 of file [GPIO\\_interface.h](#).

#### 5.14.2.2 GPIOx\_AF1

```
#define GPIOx_AF1 (1)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 1.

Definition at line 459 of file [GPIO\\_interface.h](#).

#### 5.14.2.3 GPIOx\_AF2

```
#define GPIOx_AF2 (2)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 2.

Definition at line 466 of file [GPIO\\_interface.h](#).

#### 5.14.2.4 GPIOx\_AF3

```
#define GPIOx_AF3 (3)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 3.

Definition at line 473 of file [GPIO\\_interface.h](#).

#### 5.14.2.5 GPIOx\_AF4

```
#define GPIOx_AF4 (4)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 4.

Definition at line 480 of file [GPIO\\_interface.h](#).

#### 5.14.2.6 GPIOx\_AF5

```
#define GPIOx_AF5 (5)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 5.

Definition at line 487 of file [GPIO\\_interface.h](#).

#### 5.14.2.7 GPIOx\_AF6

```
#define GPIOx_AF6 (6)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 6.

Definition at line 494 of file [GPIO\\_interface.h](#).

#### 5.14.2.8 GPIOx\_AF7

```
#define GPIOx_AF7 (7)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 7.

Definition at line 501 of file [GPIO\\_interface.h](#).

#### 5.14.2.9 GPIOx\_AF8

```
#define GPIOx_AF8 (8)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 8.

Definition at line 508 of file [GPIO\\_interface.h](#).

#### 5.14.2.10 GPIOx\_AF9

```
#define GPIOx_AF9 (9)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 9.

Definition at line 515 of file [GPIO\\_interface.h](#).

#### 5.14.2.11 GPIOx\_AF10

```
#define GPIOx_AF10 (10)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 10.

Definition at line 522 of file [GPIO\\_interface.h](#).

#### 5.14.2.12 GPIOx\_AF11

```
#define GPIOx_AF11 (11)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 11.

Definition at line 529 of file [GPIO\\_interface.h](#).

#### 5.14.2.13 GPIOx\_AF12

```
#define GPIOx_AF12 (12)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 12.

Definition at line 536 of file [GPIO\\_interface.h](#).

#### 5.14.2.14 GPIOx\_AF13

```
#define GPIOx_AF13 (13)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 13.

Definition at line 543 of file [GPIO\\_interface.h](#).

#### 5.14.2.15 GPIOx\_AF14

```
#define GPIOx_AF14 (14)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 14.

Definition at line 550 of file [GPIO\\_interface.h](#).

#### 5.14.2.16 GPIOx\_AF15

```
#define GPIOx_AF15 (15)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_interface.h>
```

GPIO Alternate function 15.

Definition at line 557 of file [GPIO\\_interface.h](#).

## 5.15 GPIO Addresses

### Macros

- `#define GPIOA_BASE_ADDRESS (0x40020000)`
- `#define GPIOB_BASE_ADDRESS (0x40020400)`
- `#define GPIOC_BASE_ADDRESS (0x40020800)`

### 5.15.1 Detailed Description

### 5.15.2 Macro Definition Documentation

#### 5.15.2.1 GPIOA\_BASE\_ADDRESS

```
#define GPIOA_BASE_ADDRESS (0x40020000)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_private.h>
```

Port A Base address

Definition at line 83 of file [GPIO\\_private.h](#).

### 5.15.2.2 GPIOB\_BASE\_ADDRESS

```
#define GPIOB_BASE_ADDRESS (0x40020400)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_private.h>
```

Port B Base address

Definition at line 90 of file [GPIO\\_private.h](#).

### 5.15.2.3 GPIOC\_BASE\_ADDRESS

```
#define GPIOC_BASE_ADDRESS (0x40020800)
```

```
#include <src/COTS/MCAL/GPIO/GPIO_private.h>
```

Port C Base address

Definition at line 97 of file [GPIO\\_private.h](#).

## 5.16 GPIO Registers

### Macros

- #define [GPIOA](#) ((volatile [P2VAR\(GPIOx\\_MemoryMapType\)](#))([GPIOA\\_BASE\\_ADDRESS](#)))
- #define [GPIOB](#) ((volatile [P2VAR\(GPIOx\\_MemoryMapType\)](#))([GPIOB\\_BASE\\_ADDRESS](#)))
- #define [GPIOC](#) ((volatile [P2VAR\(GPIOx\\_MemoryMapType\)](#))([GPIOC\\_BASE\\_ADDRESS](#)))

### 5.16.1 Detailed Description

### 5.16.2 Macro Definition Documentation

#### 5.16.2.1 GPIOA

```
#define GPIOA ((volatile P2VAR\(GPIOx\_MemoryMapType\))(GPIOA\_BASE\_ADDRESS))
```

```
#include <src/COTS/MCAL/GPIO/GPIO_private.h>
```

GPIO register for port A

Definition at line 112 of file [GPIO\\_private.h](#).

### 5.16.2.2 GPIOB

```
#define GPIOB ((volatile P2VAR(GPIOx_MemoryMapType)) (GPIOB_BASE_ADDRESS))
```

```
#include <src/COTS/MCAL/GPIO/GPIO_private.h>
```

GPIO register for port B

Definition at line 119 of file [GPIO\\_private.h](#).

### 5.16.2.3 GPIOC

```
#define GPIOC ((volatile P2VAR(GPIOx_MemoryMapType)) (GPIOC_BASE_ADDRESS))
```

```
#include <src/COTS/MCAL/GPIO/GPIO_private.h>
```

GPIO register for port C

Definition at line 126 of file [GPIO\\_private.h](#).





## Chapter 6

# Data Structure Documentation

### 6.1 GPIOx\_MemoryMapType Struct Reference

MDIO Configuration structure for a specific PIN initialization.

```
#include "GPIO_private.h"
```

#### Data Fields

- [u32\\_t MODERx](#)  
*Control PIN mode whether **INPUT** or **OUTPUT***
- [u32\\_t OTYPERx](#)  
*Select the output type between push-pull or open-drain.*
- [u32\\_t OSPEEDRx](#)  
*Configure the speed that the PIN is connected to.*
- [u32\\_t PUPDRx](#)  
*Control the pull-up or pull-down of the pin, despite its direction.*
- [u32\\_t IDRx](#)  
*Read the input data.*
- [u32\\_t ODRx](#)  
*Write the output data.*
- [u32\\_t BSRRx](#)  
*(Re)set each bit in the output data register*
- [u32\\_t LCKRx](#)  
*Lock the GPIO control registers.*
- [u32\\_t AFRLx](#)  
*Control alternate function **LOW** register.*
- [u32\\_t AFRHx](#)  
*Control alternate function **HIGH** register.*

#### 6.1.1 Detailed Description

MDIO Configuration structure for a specific PIN initialization.

Definition at line 18 of file [GPIO\\_private.h](#).

## 6.1.2 Field Documentation

### 6.1.2.1 MODERx

`u32_t` MODERx

Control PIN mode whether **INPUT** or **OUTPUT**

Definition at line 23 of file [GPIO\\_private.h](#).

### 6.1.2.2 OTYPERx

`u32_t` OTYPERx

Select the output type between push-pull or open-drain.

Definition at line 28 of file [GPIO\\_private.h](#).

### 6.1.2.3 OSPEEDRx

`u32_t` OSPEEDRx

Configure the speed that the PIN is connected to.

Definition at line 33 of file [GPIO\\_private.h](#).

### 6.1.2.4 PUPDRx

`u32_t` PUPDRx

Control the pull-up or pull-down of the pin, despite its direction.

Definition at line 38 of file [GPIO\\_private.h](#).

### 6.1.2.5 IDRx

`u32_t` IDRx

Read the input data.

Definition at line 43 of file [GPIO\\_private.h](#).

#### 6.1.2.6 ODRx

`u32_t` ODRx

Write the output data.

Definition at line 48 of file [GPIO\\_private.h](#).

#### 6.1.2.7 BSRRx

`u32_t` BSRRx

(Re)set each bit in the output data register

Definition at line 53 of file [GPIO\\_private.h](#).

#### 6.1.2.8 LCKRx

`u32_t` LCKRx

Lock the GPIO control registers.

Definition at line 58 of file [GPIO\\_private.h](#).

#### 6.1.2.9 AFRLx

`u32_t` AFRLx

Control alternate function **LOW** register.

Definition at line 63 of file [GPIO\\_private.h](#).

#### 6.1.2.10 AFRHx

`u32_t` AFRHx

Control alternate function **HIGH** register.

Definition at line 68 of file [GPIO\\_private.h](#).

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

- [src/COTS/MCAL/GPIO/GPIO\\_private.h](#)

## 6.2 MGPIOx\_ConfigType Struct Reference

MDIO Configuration structure for a speific PIN initialization.

```
#include "GPIO_interface.h"
```

### Data Fields

- [u8\\_t Port](#)  
*Configures a speific GPIO port.*
- [u8\\_t Pin](#)  
*Configures a speific GPIO pin.*
- [u8\\_t Mode](#)  
*Configures a speific GPIO mode.*
- [u8\\_t OutputType](#)  
*Configures a speific GPIO output's type.*
- [u8\\_t OutputSpeed](#)  
*Configures a speific GPIO output's speed.*
- [u8\\_t InputType](#)  
*Configures a speific GPIO input type.*
- [u8\\_t AF\\_Type](#)  
*Configures a speific GPIO Alternative function.*

### 6.2.1 Detailed Description

MDIO Configuration structure for a speific PIN initialization.

Definition at line 18 of file [GPIO\\_interface.h](#).

### 6.2.2 Field Documentation

#### 6.2.2.1 Port

[u8\\_t](#) Port

Configures a speific GPIO port.

Definition at line 23 of file [GPIO\\_interface.h](#).

Referenced by [MGPIOx\\_vInit\(\)](#).

### 6.2.2.2 Pin

`u8_t` Pin

Configures a speific GPIO pin.

Definition at line 27 of file [GPIO\\_interface.h](#).

Referenced by [MGPIOx\\_vlnit\(\)](#).

### 6.2.2.3 Mode

`u8_t` Mode

Configures a speific GPIO mode.

Definition at line 31 of file [GPIO\\_interface.h](#).

Referenced by [MGPIOx\\_vlnit\(\)](#).

### 6.2.2.4 OutputType

`u8_t` OutputType

Configures a speific GPIO output's type.

Definition at line 35 of file [GPIO\\_interface.h](#).

Referenced by [MGPIOx\\_vlnit\(\)](#).

### 6.2.2.5 OutputSpeed

`u8_t` OutputSpeed

Configures a speific GPIO output's speed.

Definition at line 39 of file [GPIO\\_interface.h](#).

Referenced by [MGPIOx\\_vlnit\(\)](#).

### 6.2.2.6 InputType

`u8_t` InputType

Configures a speific GPIO input type.

Definition at line 43 of file [GPIO\\_interface.h](#).

Referenced by [MGPIOx\\_vInit\(\)](#).

### 6.2.2.7 AF\_Type

`u8_t` AF\_Type

Configures a speific GPIO Alternative function.

Definition at line 47 of file [GPIO\\_interface.h](#).

Referenced by [MGPIOx\\_vInit\(\)](#).

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

- [src/COTS/MCAL/GPIO/GPIO\\_interface.h](#)

## 6.3 RCC\_MemoryMapType Struct Reference

```
#include "MRCC_private.h"
```

### Data Fields

- [u32\\_t](#) CR
- [u32\\_t](#) PLLCFGR
- [u32\\_t](#) CFGR
- [u32\\_t](#) CIR
- [u32\\_t](#) AHB1RSTR
- [u32\\_t](#) AHB2RSTR
- [u32\\_t](#) Reserved1
- [u32\\_t](#) Reserved2
- [u32\\_t](#) APB1RSTR
- [u32\\_t](#) APB2RSTR
- [u32\\_t](#) Reserved3
- [u32\\_t](#) Reserved4
- [u32\\_t](#) AHB1ENR
- [u32\\_t](#) AHB2ENR
- [u32\\_t](#) Reserved5
- [u32\\_t](#) Reserved6
- [u32\\_t](#) APB1ENR
- [u32\\_t](#) APB2ENR
- [u32\\_t](#) Reserved7

- [u32\\_t Reserved8](#)
- [u32\\_t AHB1LPENR](#)
- [u32\\_t AHB2LPENR](#)
- [u32\\_t Reserved9](#)
- [u32\\_t Reserved10](#)
- [u32\\_t APB1LPENR](#)
- [u32\\_t APB2LPENR](#)
- [u32\\_t Reserved11](#)
- [u32\\_t Reserved12](#)
- [u32\\_t BDCR](#)
- [u32\\_t CSR](#)
- [u32\\_t Reserved13](#)
- [u32\\_t Reserved14](#)
- [u32\\_t SSCGR](#)
- [u32\\_t PLLI2SCFGR](#)
- [u32\\_t DCKCFGR](#)

### 6.3.1 Detailed Description

Definition at line 19 of file [MRCC\\_private.h](#).

### 6.3.2 Field Documentation

#### 6.3.2.1 CR

[u32\\_t](#) CR

Definition at line 22 of file [MRCC\\_private.h](#).

#### 6.3.2.2 PLLCFGR

[u32\\_t](#) PLLCFGR

Definition at line 23 of file [MRCC\\_private.h](#).

#### 6.3.2.3 CFGR

[u32\\_t](#) CFGR

Definition at line 24 of file [MRCC\\_private.h](#).

#### 6.3.2.4 CIR

`u32_t` CIR

Definition at line 25 of file [MRCC\\_private.h](#).

#### 6.3.2.5 AHB1RSTR

`u32_t` AHB1RSTR

Definition at line 26 of file [MRCC\\_private.h](#).

#### 6.3.2.6 AHB2RSTR

`u32_t` AHB2RSTR

Definition at line 27 of file [MRCC\\_private.h](#).

#### 6.3.2.7 Reserved1

`u32_t` Reserved1

Definition at line 29 of file [MRCC\\_private.h](#).

#### 6.3.2.8 Reserved2

`u32_t` Reserved2

Definition at line 30 of file [MRCC\\_private.h](#).

#### 6.3.2.9 APB1RSTR

`u32_t` APB1RSTR

Definition at line 32 of file [MRCC\\_private.h](#).



#### 6.3.2.10 APB2RSTR

`u32_t` APB2RSTR

Definition at line 33 of file [MRCC\\_private.h](#).

#### 6.3.2.11 Reserved3

`u32_t` Reserved3

Definition at line 35 of file [MRCC\\_private.h](#).

#### 6.3.2.12 Reserved4

`u32_t` Reserved4

Definition at line 36 of file [MRCC\\_private.h](#).

#### 6.3.2.13 AHB1ENR

`u32_t` AHB1ENR

Definition at line 38 of file [MRCC\\_private.h](#).

#### 6.3.2.14 AHB2ENR

`u32_t` AHB2ENR

Definition at line 39 of file [MRCC\\_private.h](#).

#### 6.3.2.15 Reserved5

`u32_t` Reserved5

Definition at line 41 of file [MRCC\\_private.h](#).

#### 6.3.2.16 Reserved6

`u32_t` Reserved6

Definition at line 42 of file [MRCC\\_private.h](#).

#### 6.3.2.17 APB1ENR

`u32_t` APB1ENR

Definition at line 44 of file [MRCC\\_private.h](#).

#### 6.3.2.18 APB2ENR

`u32_t` APB2ENR

Definition at line 45 of file [MRCC\\_private.h](#).

#### 6.3.2.19 Reserved7

`u32_t` Reserved7

Definition at line 47 of file [MRCC\\_private.h](#).

#### 6.3.2.20 Reserved8

`u32_t` Reserved8

Definition at line 48 of file [MRCC\\_private.h](#).

#### 6.3.2.21 AHB1LPENR

`u32_t` AHB1LPENR

Definition at line 50 of file [MRCC\\_private.h](#).

#### 6.3.2.22 AHB2LPENR

`u32_t` AHB2LPENR

Definition at line 51 of file [MRCC\\_private.h](#).

#### 6.3.2.23 Reserved9

`u32_t` Reserved9

Definition at line 53 of file [MRCC\\_private.h](#).

#### 6.3.2.24 Reserved10

`u32_t` Reserved10

Definition at line 54 of file [MRCC\\_private.h](#).

#### 6.3.2.25 APB1LPENR

`u32_t` APB1LPENR

Definition at line 56 of file [MRCC\\_private.h](#).

#### 6.3.2.26 APB2LPENR

`u32_t` APB2LPENR

Definition at line 57 of file [MRCC\\_private.h](#).

#### 6.3.2.27 Reserved11

`u32_t` Reserved11

Definition at line 59 of file [MRCC\\_private.h](#).

#### 6.3.2.28 Reserved12

`u32_t` Reserved12

Definition at line 60 of file [MRCC\\_private.h](#).

#### 6.3.2.29 BDCR

`u32_t` BDCR

Definition at line 62 of file [MRCC\\_private.h](#).

#### 6.3.2.30 CSR

`u32_t` CSR

Definition at line 63 of file [MRCC\\_private.h](#).

#### 6.3.2.31 Reserved13

`u32_t` Reserved13

Definition at line 65 of file [MRCC\\_private.h](#).

#### 6.3.2.32 Reserved14

`u32_t` Reserved14

Definition at line 66 of file [MRCC\\_private.h](#).

#### 6.3.2.33 SSCGR

`u32_t` SSCGR

Definition at line 68 of file [MRCC\\_private.h](#).

#### 6.3.2.34 PLLI2SCFGR

`u32_t` PLLI2SCFGR

Definition at line 69 of file [MRCC\\_private.h](#).

#### 6.3.2.35 DCKCFGR

`u32_t` DCKCFGR

Definition at line 70 of file [MRCC\\_private.h](#).

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

- [src/COTS/MCAL/RCC/MRCC\\_private.h](#)

## Chapter 7

# File Documentation

### 7.1 README.md File Reference

### 7.2 src/COTS/LIB/LSTD\_BITMATH.h File Reference

This file contains the bit math manipulation macro-functions.

#### Macros

- #define [SET\\_BIT](#)(Reg, bitnum) (Reg) |= (1 << (bitnum))
- #define [CLR\\_BIT](#)(Reg, bitnum) (Reg) &= ~(1 << (bitnum))
- #define [TOGGLE\\_BIT](#)(Reg, bitnum) (Reg) ^= (1 << (bitnum))
- #define [GET\\_BIT](#)(Reg, bitnum) (((Reg)>>(bitnum)) & 1)
- #define [SET\\_BITS](#)(Reg, bits, bitnum, factor) (Reg) |= ((bits) << (bitnum \* factor))
- #define [CLR\\_BITS](#)(Reg, bits, bitnum, factor) (Reg) &= ~((bits) << (bitnum \* factor))
- #define [TOGGLE\\_BITS](#)(Reg, bits, bitnum, factor) (Reg) ^= ((bits) << (bitnum \* factor))
- #define [GET\\_BITS](#)(Reg, bits, bitnum, factor) (((Reg) >> (bitnum \* factor)) & (bits))

#### 7.2.1 Detailed Description

This file contains the bit math manipulation macro-functions.

#### Author

Ali El Bana

#### Version

1.0

#### Date

10/29/2022

Definition in file [LSTD\\_BITMATH.h](#).

## 7.3 LSTD\_BITMATH.h

[Go to the documentation of this file.](#)

```

00001
00009 #ifndef COTS_LIB_LSTD_BITMATH_H_
00010 #define COTS_LIB_LSTD_BITMATH_H_
00011
00023 #define SET_BIT(Reg, bitnum) (Reg) |= (1 << (bitnum))
00024
00030 #define CLR_BIT(Reg, bitnum) (Reg) &= ~(1 << (bitnum))
00031
00037 #define TOGGLE_BIT(Reg, bitnum) (Reg) ^= (1 << (bitnum))
00038
00044 #define GET_BIT(Reg, bitnum) (((Reg)>>(bitnum)) & 1)
00045
00059 #define SET_BITS(Reg, bits, bitnum, factor) (Reg) |= ((bits) << (bitnum * factor))
00060
00066 #define CLR_BITS(Reg, bits, bitnum, factor) (Reg) &= ~((bits) << (bitnum * factor))
00067
00073 #define TOGGLE_BITS(Reg, bits, bitnum, factor) (Reg) ^= ((bits) << (bitnum * factor))
00074
00080 #define GET_BITS(Reg, bits, bitnum, factor) (((Reg) >> (bitnum * factor)) & (bits))
00081
00084 #endif /* COTS_LIB_LSTD_BITMATH_H_ */

```

## 7.4 src/COTS/LIB/LSTD\_COMPILER.h File Reference

This file contains the compiler standard macros.

### Macros

- #define [VAR](#)(vartype) vartype
- #define [FUNC](#)(rettype) rettype
- #define [P2VAR](#)(ptrtype) ptrtype \*
- #define [P2CONST](#)(ptrtype) const ptrtype \*
- #define [CONSTP2VAR](#)(ptrtype) ptrtype \* const
- #define [CONSTP2CONST](#)(ptrtype) const ptrtype \* const
- #define [P2FUNC](#)(rettype, fctname) rettype (\* fctname)
- #define [CONST](#)(consttype) const consttype
- #define [STATIC](#) static

### 7.4.1 Detailed Description

This file contains the compiler standard macros.

#### Author

Mohamed Alaa

#### Version

1.0

#### Date

11/04/2022

Definition in file [LSTD\\_COMPILER.h](#).

## 7.5 LSTD\_COMPILER.h

[Go to the documentation of this file.](#)

```

00001
00009 #ifndef COTS_LIB_LSTD_COMPILER_H_
00010 #define COTS_LIB_LSTD_COMPILER_H_
00011
00023 #define VAR(vartype) vartype
00024
00030 #define FUNC(rettype) rettype
00031
00037 #define P2VAR(ptrtype) ptrtype *
00038
00044 #define P2CONST(ptrtype) const ptrtype *
00045
00051 #define CONSTP2VAR(ptrtype) ptrtype * const
00052
00058 #define CONSTP2CONST(ptrtype) const ptrtype * const
00059
00065 #define P2FUNC(rettype, fctname) rettype (* fctname)
00066
00072 #ifndef CONST
00073 #define CONST(consttype) const consttype
00074 #endif
00075
00081 #ifndef STATIC
00082 #define STATIC static
00083 #endif
00084
00087 #endif /* COTS_LIB_LSTD_COMPILER_H_ */

```

## 7.6 src/COTS/LIB/LSTD\_MCU\_UTILITIES.h File Reference

This file contains the MCU utility macro-functions.

### Macros

- #define [MY\\_MS\\_DELAY](#)(T) do{ u32 Timer = (T \* 500); while (Timer--) { asm ("nop"); } } while(0);

### 7.6.1 Detailed Description

This file contains the MCU utility macro-functions.

#### Author

Ali El Bana

#### Version

1.0

#### Date

10/29/2022

Definition in file [LSTD\\_MCU\\_UTILITIES.h](#).

## 7.7 LSTD\_MCU\_UTILITIES.h

[Go to the documentation of this file.](#)

```
00001
00009 #ifndef COTS_LIB_LSTD_MCU_UTILITIES_H_
00010 #define COTS_LIB_LSTD_MCU_UTILITIES_H_
00023 #define MY_MS_DELAY(T)    do{ u32 Timer = (T * 500); while (Timer-->0) { asm ("nop"); } } while(0);
00024
00027 #endif /* COTS_LIB_LSTD_MCU_UTILITIES_H_ */
```

## 7.8 src/COTS/LIB/LSTD\_TYPES.h File Reference

This file contains the standard types.

### Typedefs

- typedef unsigned char [bool\\_t](#)
- typedef unsigned char [u8\\_t](#)
- typedef unsigned short int [u16\\_t](#)
- typedef unsigned long int [u32\\_t](#)
- typedef signed char [s8\\_t](#)
- typedef signed short int [s16\\_t](#)
- typedef signed long int [s32\\_t](#)
- typedef float [fl32\\_t](#)
- typedef double [fl64\\_t](#)

### 7.8.1 Detailed Description

This file contains the standard types.

#### Author

Ali El Bana

#### Version

1.0

#### Date

10/29/2022

Definition in file [LSTD\\_TYPES.h](#).



## 7.9 LSTD\_TYPES.h

[Go to the documentation of this file.](#)

```
00001
00009 #ifndef COTS_LIB_LSTD_TYPES_H_
00010 #define COTS_LIB_LSTD_TYPES_H_
00011
00023 typedef unsigned char bool_t;
00024
00030 typedef unsigned char u8_t;
00031
00037 typedef unsigned short int u16_t;
00038
00044 typedef unsigned long int u32_t;
00045
00051 typedef signed char s8_t;
00052
00058 typedef signed short int s16_t;
00059
00065 typedef signed long int s32_t;
00066
00072 typedef float f132_t;
00073
00079 typedef double f164_t;
00080
00083 #endif /* COTS_LIB_LSTD_TYPES_H_ */
```

## 7.10 src/COTS/LIB/LSTD\_VALUES.h File Reference

This file contains the standard values.

### Macros

- `#define TRUE` (1)
- `#define FALSE` (0)
- `#define NULL` (`P2VAR(void) 0`)
- `#define INITIAL_ZERO` (0)
- `#define FLAG_SET` (1)
- `#define FLAG_CLEARED` (0)

### 7.10.1 Detailed Description

This file contains the standard values.

#### Author

Ali El Bana

#### Version

1.0

#### Date

10/29/2022

Definition in file [LSTD\\_VALUES.h](#).

## 7.11 LSTD\_VALUES.h

[Go to the documentation of this file.](#)

```
00001
00009 #ifndef COTS_LIB_LSTD_VALUES_H_
00010 #define COTS_LIB_LSTD_VALUES_H_
00011
00023 #ifndef TRUE
00024 #define TRUE (1)
00025 #endif /* ifndef TRUE */
00026
00032 #ifndef FALSE
00033 #define FALSE (0)
00034 #endif /* ifndef FALSE */
00035
00041 #ifndef NULL
00042 #define NULL (P2VAR(void) 0)
00043 #endif /* ifndef NULL */
00044
00050 #ifndef INITIAL_ZERO
00051 #define INITIAL_ZERO (0)
00052 #endif /* ifndef INITIAL_ZERO */
00053
00059 #ifndef FLAG_SET
00060 #define FLAG_SET (1)
00061 #endif /* ifndef FLAG_SET */
00062
00068 #ifndef FLAG_CLEARED
00069 #define FLAG_CLEARED (0)
00070 #endif /* ifndef FLAG_CLEARED */
00071
00074 #endif /* COTS_LIB_LSTD_VALUES_H_ */
```

## 7.12 src/COTS/MCAL/GPIO/GPIO\_config.h File Reference

This file contains the GPIO configurations.

### Macros

- #define [GPIOA\\_PIN\\_POS](#) (0b1110000000000000)
- #define [GPIOB\\_PIN\\_POS](#) (0b00000000000011100)
- #define [LCKK\\_BIT\\_POS](#) (16U)

### 7.12.1 Detailed Description

This file contains the GPIO configurations.

#### Author

Ali El Bana & Mo Alaa

#### Version

2.0

#### Date

08/22/2022

Definition in file [GPIO\\_config.h](#).

## 7.12.2 Macro Definition Documentation

### 7.12.2.1 GPIOA\_PIN\_POS

```
#define GPIOA_PIN_POS (0b1110000000000000)
```

Definition at line 17 of file [GPIO\\_config.h](#).

### 7.12.2.2 GPIOB\_PIN\_POS

```
#define GPIOB_PIN_POS (0b00000000000011100)
```

Definition at line 18 of file [GPIO\\_config.h](#).

### 7.12.2.3 LCKK\_BIT\_POS

```
#define LCKK_BIT_POS (16U)
```

Definition at line 19 of file [GPIO\\_config.h](#).

## 7.13 GPIO\_config.h

[Go to the documentation of this file.](#)

```
00001
00009 #ifndef _GPIO_config_H
00010 #define _GPIO_config_H
00011
00012
00013 /*****
00014                                     // GPIOx configurations //
00015 *****/
00016
00017 #define GPIOA_PIN_POS (0b1110000000000000) //I want to lock PA13,14,15.
00018 #define GPIOB_PIN_POS (0b00000000000011100) //I want to lock PB2,3,4.
00019 #define LCKK_BIT_POS (16U) //Position of LCKK bit.
00020
00021
00022 #endif // _GPIO_config_H
```

## 7.14 src/COTS/MCAL/GPIO/GPIO\_interface.h File Reference

This file contains the interfacing information for the GPIO module.

## Data Structures

- struct [MGPIOx\\_ConfigType](#)  
*MDIO Configuration structure for a speific PIN initialization.*

## Macros

- #define [GPIOx\\_MODE\\_INPUT](#) (0b00)  
*Control input mode.*
- #define [GPIOx\\_MODE\\_OUTPUT](#) (0b01)  
*Control output mode.*
- #define [GPIOx\\_MODE\\_AF](#) (0b10)  
*Control alternate function mode.*
- #define [GPIOx\\_MODE\\_ANALOG](#) (0b11)  
*Control analog mode.*
- #define [GPIO\\_PORTA](#) (0)  
*GPIO Port A.*
- #define [GPIO\\_PORTB](#) (1)  
*GPIO Port B.*
- #define [GPIO\\_PORTC](#) (2)  
*GPIO Port C.*
- #define [GPIOx\\_OPENDRAIN](#) (1)  
*GPIO open-drain.*
- #define [GPIOx\\_PUSHPULL](#) (2)  
*GPIO push-pull.*
- #define [GPIOx\\_LowSpeed](#) (0b00)  
*GPIO low speed.*
- #define [GPIOx\\_MediumSpeed](#) (0b01)  
*GPIO medium speed.*
- #define [GPIOx\\_HighSpeed](#) (0b10)  
*GPIO high speed.*
- #define [GPIOx\\_VeryHighSpeed](#) (0b11)  
*GPIO very high speed.*
- #define [GPIOx\\_NoPull](#) (0b00)  
*GPIO No PULL.*
- #define [GPIOx\\_PullUp](#) (0b01)  
*GPIO Pull UP.*
- #define [GPIOx\\_PullDown](#) (0b10)  
*GPIO Pull Down.*
- #define [GPIOx\\_HIGH](#) (1)  
*GPIO output high.*
- #define [GPIOx\\_LOW](#) (2)  
*GPIO output low.*
- #define [GPIOx\\_PIN0](#) (0)  
*GPIO PIN 0.*
- #define [GPIOx\\_PIN1](#) (1)  
*GPIO PIN 1.*
- #define [GPIOx\\_PIN2](#) (2)  
*GPIO PIN 2.*
- #define [GPIOx\\_PIN3](#) (3)

- GPIO PIN 3.*
- #define [GPIOx\\_PIN4](#) (4)
- GPIO PIN 4.*
- #define [GPIOx\\_PIN5](#) (5)
- GPIO PIN 5.*
- #define [GPIOx\\_PIN6](#) (6)
- GPIO PIN 6.*
- #define [GPIOx\\_PIN7](#) (7)
- GPIO PIN 7.*
- #define [GPIOx\\_PIN8](#) (8)
- GPIO PIN 8.*
- #define [GPIOx\\_PIN9](#) (9)
- brief GPIO PIN 9*
- #define [GPIOx\\_PIN10](#) (10)
- GPIO PIN 10.*
- #define [GPIOx\\_PIN11](#) (11)
- GPIO PIN 11.*
- #define [GPIOx\\_PIN12](#) (12)
- GPIO PIN 12.*
- #define [GPIOx\\_PIN13](#) (13)
- GPIO PIN 13.*
- #define [GPIOx\\_PIN14](#) (14)
- GPIO PIN 14.*
- #define [GPIOx\\_PIN15](#) (15)
- GPIO PIN 15.*
- #define [GPIOx\\_AF0](#) (0)
- GPIO Alternate function 0.*
- #define [GPIOx\\_AF1](#) (1)
- GPIO Alternate function 1.*
- #define [GPIOx\\_AF2](#) (2)
- GPIO Alternate function 2.*
- #define [GPIOx\\_AF3](#) (3)
- GPIO Alternate function 3.*
- #define [GPIOx\\_AF4](#) (4)
- GPIO Alternate function 4.*
- #define [GPIOx\\_AF5](#) (5)
- GPIO Alternate function 5.*
- #define [GPIOx\\_AF6](#) (6)
- GPIO Alternate function 6.*
- #define [GPIOx\\_AF7](#) (7)
- GPIO Alternate function 7.*
- #define [GPIOx\\_AF8](#) (8)
- GPIO Alternate function 8.*
- #define [GPIOx\\_AF9](#) (9)
- GPIO Alternate function 9.*
- #define [GPIOx\\_AF10](#) (10)
- GPIO Alternate function 10.*
- #define [GPIOx\\_AF11](#) (11)
- GPIO Alternate function 11.*
- #define [GPIOx\\_AF12](#) (12)
- GPIO Alternate function 12.*

- `#define GPIOx_AF13` (13)  
*GPIO Alternate function 13.*
- `#define GPIOx_AF14` (14)  
*GPIO Alternate function 14.*
- `#define GPIOx_AF15` (15)  
*GPIO Alternate function 15.*

## Functions

- void `MGPIOx_vLockedPins` (void)
- void `MGPIOx_vSetPinMode` (VAR(u8\_t) A\_u8PortID, VAR(u8\_t) A\_u8PinID, VAR(u8\_t) A\_u8Mode)  
*Sets a certain pin's mode on a speific port.*
- void `MGPIOx_vSetPinOutputType` (VAR(u8\_t) A\_u8PortID, VAR(u8\_t) A\_u8PinID, VAR(u8\_t) A\_u8OutputType)  
*Sets a certain pin's output type on a speific port.*
- void `MGPIOx_vSetPinOutputSpeed` (VAR(u8\_t) A\_u8PortID, VAR(u8\_t) A\_u8PinID, VAR(u8\_t) A\_u8OutputSpeed)  
*Sets a certain pin's output speed on a speific port.*
- void `MGPIOx_vSetPinInputPullType` (VAR(u8\_t) A\_u8PortID, VAR(u8\_t) A\_u8PinID, VAR(u8\_t) A\_u8InputPullType)  
*Sets a certain pin's input pull type on a speific port.*
- u8\_t `MGPIOx_u8GetPinValue` (VAR(u8\_t) A\_u8PortID, VAR(u8\_t) A\_u8PinID)  
*Gets the value currently on a certain pin.*
- void `MGPIOx_vSetPinValue` (VAR(u8\_t) A\_u8PortID, VAR(u8\_t) A\_u8PinID, VAR(u8\_t) A\_u8PinValue)  
*Sets a certain pin's output value on a speific port.*
- void `MGPIOx_vSetResetAtomic` (VAR(u8\_t) A\_u8PortID, VAR(u8\_t) A\_u8PinID, VAR(u8\_t) A\_u8SetResetPinValue)  
*Resets a certain pin's output value on a speific port.*
- void `MGPIOx_vSetAlternateFunctionON` (VAR(u8\_t) A\_u8PortID, VAR(u8\_t) A\_u8PinID, VAR(u8\_t) A\_u8AFID)  
*Applys an alternative function on a certain pin.*
- void `MGPIOx_vSetPortConfigLock` (VAR(u8\_t) A\_u8PortID)  
*Updates a port's configuration lock.*
- void `MGPIOx_vInit` (P2VAR(MGPIOx\_ConfigType) A\_xPinConfig)  
*Initialize the GPIO with a certain configuration.*
- void `MGPIOx_vTogglePinValue` (VAR(u8\_t) A\_u8PortID, VAR(u8\_t) A\_u8PinID)  
*Toggles a certain's pin's value on a certain port.*

### 7.14.1 Detailed Description

This file contains the interfacing information for the GPIO module.

#### Author

Ali El Bana & Mo Alaa

#### Version

2.0

#### Date

11/9/2022

Definition in file [GPIO\\_interface.h](#).

## 7.14.2 Function Documentation

### 7.14.2.1 MGPIOX\_vLockedPins()

```
void MGPIOX_vLockedPins (
    void )
```

Locks the prohibited GPIO PINs.

Definition at line 24 of file [GPIO\\_program.c](#).

```
00025 {
00026
00027     VAR(volatile u32_t) L_u32LockGPIOA = INITIAL_ZERO ;
00028
00029     /* Lock key write sequence */
00030
00031     /* WR LCKR[16] = 01 + LCKR[13,14,15] = 01 */
00032     L_u32LockGPIOA = ( (1UL << LCKK_BIT_POS) | (GPIOA_PIN_POS) ) ;
00033
00034     GPIOA->LCKRx = L_u32LockGPIOA ;
00035
00036     /* WR LCKR[16] = 00 + LCKR[13,14,15] should not change */
00037     GPIOA->LCKRx = (GPIOA_PIN_POS) ;
00038
00039     /* WR LCKR[16] = 01 + LCKR[13,14,15] should not change */
00040     GPIOA->LCKRx = L_u32LockGPIOA ;
00041
00042     /* RD LCKR */
00043     L_u32LockGPIOA = GPIOA->LCKRx ;
00044
00045
00046
00047     VAR(volatile u32_t) L_u32LockGPIOB = INITIAL_ZERO ;
00048
00049     /* Lock key write sequence */
00050
00051     /* WR LCKR[16] = 01 + LCKR[2,3,4] = 01 */
00052     L_u32LockGPIOB = ( (1UL << LCKK_BIT_POS) | (GPIOB_PIN_POS) ) ;
00053
00054     GPIOB->LCKRx = L_u32LockGPIOB ;
00055
00056     /* WR LCKR[16] = 00 + LCKR[2,3,4] should not change */
00057     GPIOB->LCKRx = (GPIOB_PIN_POS) ;
00058
00059     /* WR LCKR[16] = 01 + LCKR[2,3,4] should not change */
00060     GPIOB->LCKRx = L_u32LockGPIOB ;
00061
00062     /* RD LCKR */
00063     L_u32LockGPIOB = GPIOB->LCKRx ;
00064
00065 }
```

References [GPIOA](#), [GPIOA\\_PIN\\_POS](#), [GPIOB](#), [GPIOB\\_PIN\\_POS](#), [INITIAL\\_ZERO](#), [LCKK\\_BIT\\_POS](#), and [VAR](#).

### 7.14.2.2 MGPIOX\_vSetPinMode()

```
void MGPIOX_vSetPinMode (
    VAR(u8_t) A_u8PortID,
    VAR(u8_t) A_u8PinID,
    VAR(u8_t) A_u8Mode )
```

Sets a certain pin's mode on a specific port.

**Parameters**

in	<i>A_u8PortID</i>	The port that the pin belongs to
in	<i>A_u8PinID</i>	The pin to update its mode
in	<i>A_u8Mode</i>	The mode to apply the pin

**7.14.2.3 MGPIOX\_vSetPinOutputType()**

```
void MGPIOX_vSetPinOutputType (
    VAR(u8_t) A_u8PortID,
    VAR(u8_t) A_u8PinID,
    VAR(u8_t) A_u8OutputType )
```

Sets a certain pin's output type on a specific port.

**Parameters**

in	<i>A_u8PortID</i>	The port that the pin belongs to
in	<i>A_u8PinID</i>	The pin to update its mode
in	<i>A_u8OutputType</i>	The output type to apply on the pin

**7.14.2.4 MGPIOX\_vSetPinOutputSpeed()**

```
void MGPIOX_vSetPinOutputSpeed (
    VAR(u8_t) A_u8PortID,
    VAR(u8_t) A_u8PinID,
    VAR(u8_t) A_u8OutputSpeed )
```

Sets a certain pin's output speed on a specific port.

**Parameters**

in	<i>A_u8PortID</i>	The port that the pin belongs to
in	<i>A_u8PinID</i>	The pin to update its mode
in	<i>A_u8OutputSpeed</i>	The output speed to apply on the pin

**7.14.2.5 MGPIOX\_vSetPinInputPullType()**

```
void MGPIOX_vSetPinInputPullType (
    VAR(u8_t) A_u8PortID,
```



```
VAR(u8_t) A_u8PinID,  
VAR(u8_t) A_u8InputPullType )
```

Sets a certain pin's input pull type on a speific port.

**Parameters**

in	<i>A_u8PortID</i>	The port that the pin belongs to
in	<i>A_u8PinID</i>	The pin to update its mode
in	<i>A_u8InputPullType</i>	The input pull type to apply on the pin

**7.14.2.6 MGPIOX\_u8GetPinValue()**

```
u8_t MGPIOX_u8GetPinValue (
    VAR(u8_t) A_u8PortID,
    VAR(u8_t) A_u8PinID )
```

Gets the value currently on a certain pin.

**Parameters**

in	<i>A_u8PortID</i>	The port that the pin belongs to
in	<i>A_u8PinID</i>	The pin to update its mode

**Returns**

The current value on the pin

**7.14.2.7 MGPIOX\_vSetPinValue()**

```
void MGPIOX_vSetPinValue (
    VAR(u8_t) A_u8PortID,
    VAR(u8_t) A_u8PinID,
    VAR(u8_t) A_u8PinValue )
```

Sets a certain pin's output value on a speific port.

**Parameters**

in	<i>A_u8PortID</i>	The port that the pin belongs to
in	<i>A_u8PinID</i>	The pin to update its mode
in	<i>A_u8PinValue</i>	The set value to set on the pin

**7.14.2.8 MGPIOX\_vSetResetAtomic()**

```
void MGPIOX_vSetResetAtomic (
    VAR(u8_t) A_u8PortID,
```

```

VAR(u8_t) A_u8PinID,
VAR(u8_t) A_u8SetResetPinValue )

```

Resets a certain pin's output value on a specific port.

#### Parameters

in	<i>A_u8PortID</i>	The port that the pin belongs to
in	<i>A_u8PinID</i>	The pin to update its mode
in	<i>A_u8SetResetPinValue</i>	The reset value to set on the pin

#### 7.14.2.9 MGPIOX\_vSetAlternateFunctionON()

```

void MGPIOX_vSetAlternateFunctionON (
    VAR(u8_t) A_u8PortID,
    VAR(u8_t) A_u8PinID,
    VAR(u8_t) A_u8AFID )

```

Applies an alternative function on a certain pin.

#### Parameters

in	<i>A_u8PortID</i>	The port that the pin belongs to
in	<i>A_u8PinID</i>	The pin to update its mode
in	<i>A_u8AFID</i>	The alternative function to apply on the pin

#### 7.14.2.10 MGPIOX\_vSetPortConfigLock()

```

void MGPIOX_vSetPortConfigLock (
    VAR(u8_t) A_u8PortID )

```

Updates a port's configuration lock.

#### Parameters

in	<i>A_u8PortID</i>	The port to update the pin's mode
----	-------------------	-----------------------------------

#### 7.14.2.11 MGPIOX\_vInit()

```

void MGPIOX_vInit (
    P2VAR(MGPIOX_ConfigType) A_xPinConfig )

```

Initialize the GPIO with a certain configuration.

## Parameters

in	<i>A_xPinConfig</i>	The initialization configuration for the GPIO
----	---------------------	---

## 7.14.2.12 MGPI0x\_vTogglePinValue()

```
void MGPI0x_vTogglePinValue (
    VAR(u8_t) A_u8PortID,
    VAR(u8_t) A_u8PinID )
```

Toggles a certain's pin's value on a certain port.

## Parameters

in	<i>A_u8PortID</i>	The port that the pin belongs to
in	<i>A_u8PinID</i>	The pin to toggle its value

## 7.15 GPIO\_interface.h

[Go to the documentation of this file.](#)

```
00001
00009 /* Header file guard */
00010 #ifndef _GPIO_interface_H
00011 #define _GPIO_interface_H
00012
00018 typedef struct
00019 {
00023     u8_t Port;
00027     u8_t Pin;
00031     u8_t Mode;
00035     u8_t OutputType;
00039     u8_t OutputSpeed;
00043     u8_t InputType;
00047     u8_t AF_Type;
00048
00049 } MGPI0x_ConfigType;
00050
00051 /*****
00052  * Functions prototypes
00053  */
00054
00058 void MGPI0x_vLockedPins(void);
00059
00066 void MGPI0x_vSetPinMode(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t) A_u8Mode);
00067
00074 void MGPI0x_vSetPinOutputType(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t) A_u8OutputType);
00075
00082 void MGPI0x_vSetPinOutputSpeed(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t) A_u8OutputSpeed);
00083
00090 void MGPI0x_vSetPinInputPullType(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t)
    A_u8InputPullType);
00091
00098 u8_t MGPI0x_u8GetPinValue(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID);
00099
00106 void MGPI0x_vSetPinValue(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t) A_u8PinValue);
00107
00114 void MGPI0x_vSetResetAtomic(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t)
    A_u8SetResetPinValue);
00115
00122 void MGPI0x_vSetAlternateFunctionON(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t) A_u8AFID);
00123
00128 void MGPI0x_vSetPortConfigLock(VAR(u8_t) A_u8PortID);
00129
```

```

00134 void MGPIOn_vInit (P2VAR (MGPIOn_ConfigType) A_xPinConfig);
00135
00141 void MGPIOn_vTogglePinValue (VAR (u8_t) A_u8PortID, VAR (u8_t) A_u8PinID);
00142
00143 /*****
00144  */
00145 /*****
00146
00158 #define GPIOx_MODE_INPUT (0b00)
00159
00165 #define GPIOx_MODE_OUTPUT (0b01)
00166
00172 #define GPIOx_MODE_AF (0b10)
00173
00179 #define GPIOx_MODE_ANALOG (0b11)
00180
00194 #define GPIO_PORTA (0)
00195
00201 #define GPIO_PORTB (1)
00202
00208 #define GPIO_PORTC (2)
00209
00223 #define GPIOx_OPENDRAIN (1)
00224
00230 #define GPIOx_PUSHPULL (2)
00231
00245 #define GPIOx_LowSpeed (0b00)
00246
00252 #define GPIOx_MediumSpeed (0b01)
00253
00259 #define GPIOx_HighSpeed (0b10)
00260
00266 #define GPIOx_VeryHighSpeed (0b11)
00267
00281 #define GPIOx_NoPull (0b00)
00282
00288 #define GPIOx_PullUp (0b01)
00289
00295 #define GPIOx_PullDown (0b10)
00296
00310 #define GPIOx_HIGH (1)
00311
00317 #define GPIOx_LOW (2)
00318
00332 #define GPIOx_PIN0 (0)
00333
00339 #define GPIOx_PIN1 (1)
00340
00346 #define GPIOx_PIN2 (2)
00347
00353 #define GPIOx_PIN3 (3)
00354
00360 #define GPIOx_PIN4 (4)
00361
00367 #define GPIOx_PIN5 (5)
00368
00374 #define GPIOx_PIN6 (6)
00375
00381 #define GPIOx_PIN7 (7)
00382
00388 #define GPIOx_PIN8 (8)
00389
00395 #define GPIOx_PIN9 (9)
00396
00402 #define GPIOx_PIN10 (10)
00403
00409 #define GPIOx_PIN11 (11)
00410
00416 #define GPIOx_PIN12 (12)
00417
00423 #define GPIOx_PIN13 (13)
00424
00430 #define GPIOx_PIN14 (14)
00431
00437 #define GPIOx_PIN15 (15)
00438
00452 #define GPIOx_AF0 (0)
00453
00459 #define GPIOx_AF1 (1)
00460
00466 #define GPIOx_AF2 (2)
00467
00473 #define GPIOx_AF3 (3)
00474
00480 #define GPIOx_AF4 (4)
00481
00487 #define GPIOx_AF5 (5)

```

```

00488
00494 #define GPIOx_AF6 (6)
00495
00501 #define GPIOx_AF7 (7)
00502
00508 #define GPIOx_AF8 (8)
00509
00515 #define GPIOx_AF9 (9)
00516
00522 #define GPIOx_AF10 (10)
00523
00529 #define GPIOx_AF11 (11)
00530
00536 #define GPIOx_AF12 (12)
00537
00543 #define GPIOx_AF13 (13)
00544
00550 #define GPIOx_AF14 (14)
00551
00557 #define GPIOx_AF15 (15)
00558
00561 #endif // _GPIO_interface_H

```

## 7.16 src/COTS/MCAL/GPIO/GPIO\_private.h File Reference

This file contains the registers information and addresses for the GPIO module.

### Data Structures

- struct [GPIOx\\_MemoryMapType](#)  
*MDIO Configuration structure for a speific PIN initialization.*

### Macros

- #define [GPIOA\\_BASE\\_ADDRESS](#) (0x40020000)
- #define [GPIOB\\_BASE\\_ADDRESS](#) (0x40020400)
- #define [GPIOC\\_BASE\\_ADDRESS](#) (0x40020800)
- #define [GPIOA](#) ((volatile [P2VAR\(GPIOx\\_MemoryMapType\)](#))([GPIOA\\_BASE\\_ADDRESS](#)))
- #define [GPIOB](#) ((volatile [P2VAR\(GPIOx\\_MemoryMapType\)](#))([GPIOB\\_BASE\\_ADDRESS](#)))
- #define [GPIOC](#) ((volatile [P2VAR\(GPIOx\\_MemoryMapType\)](#))([GPIOC\\_BASE\\_ADDRESS](#)))

#### 7.16.1 Detailed Description

This file contains the registers information and addresses for the GPIO module.

#### Author

Ali El Bana & Mo Alaa

#### Version

2.0

#### Date

08/22/2022

Definition in file [GPIO\\_private.h](#).

## 7.17 GPIO\_private.h

[Go to the documentation of this file.](#)

```

00001
00009 /* Header file guard */
00010 #ifndef _GPIO_private_H
00011 #define _GPIO_private_H
00012
00018 typedef struct
00019 {
00023     u32_t MODERx;
00024
00028     u32_t OTYPERx;
00029
00033     u32_t OSPEEDRx;
00034
00038     u32_t PUPDRx;
00039
00043     u32_t IDRx;
00044
00048     u32_t ODRx;
00049
00053     u32_t BSRRx;
00054
00058     u32_t LCKRx;
00059
00063     u32_t AFRLx;
00064
00068     u32_t AFRHx;
00069
00070 } GPIOx_MemoryMapType;
00071
00083 #define GPIOA_BASE_ADDRESS (0x40020000)
00084
00090 #define GPIOB_BASE_ADDRESS (0x40020400)
00091
00097 #define GPIOC_BASE_ADDRESS (0x40020800)
00098
00112 #define GPIOA ((volatile P2VAR(GPIOx_MemoryMapType)) (GPIOA_BASE_ADDRESS))
00113
00119 #define GPIOB ((volatile P2VAR(GPIOx_MemoryMapType)) (GPIOB_BASE_ADDRESS))
00120
00126 #define GPIOC ((volatile P2VAR(GPIOx_MemoryMapType)) (GPIOC_BASE_ADDRESS))
00127
00130 #endif //_GPIO_private_H

```

## 7.18 src/COTS/MCAL/GPIO/GPIO\_program.c File Reference

This file contains the source code of the interfacing for the GPIO modules.

```

#include "../LIB/LSTD_TYPES.h"
#include "../LIB/LSTD_COMPILER.h"
#include "../LIB/LSTD_VALUES.h"
#include "../LIB/LSTD_BITMATH.h"
#include "GPIO_interface.h"
#include "GPIO_private.h"
#include "GPIO_config.h"

```

### Functions

- void [MGPIOx\\_vLockedPins](#) (void)
- void [MGPIOx\\_vSetPinMode](#) (u8\_t A\_u8PortID, u8\_t A\_u8PinID, u8\_t A\_u8Mode)
- void [MGPIOx\\_vSetPinOutputType](#) (u8\_t A\_u8PortID, u8\_t A\_u8PinID, u8\_t A\_u8OutputType)
- void [MGPIOx\\_vSetPinOutputSpeed](#) (u8\_t A\_u8PortID, u8\_t A\_u8PinID, u8\_t A\_u8OutputSpeed)
- void [MGPIOx\\_vSetPinInputPullType](#) (u8\_t A\_u8PortID, u8\_t A\_u8PinID, u8\_t A\_u8InputPullType)
- u8\_t [MGPIOx\\_u8GetPinValue](#) (u8\_t A\_u8PortID, u8\_t A\_u8PinID)

- void [MGPIOx\\_vSetPinValue](#) ([u8\\_t A\\_u8PortID](#), [u8\\_t A\\_u8PinID](#), [u8\\_t A\\_u8PinValue](#))
- void [MGPIOx\\_vSetResetAtomic](#) ([u8\\_t A\\_u8PortID](#), [u8\\_t A\\_u8PinID](#), [u8\\_t A\\_u8SetResetPinValue](#))
- void [MGPIOx\\_vSetAlternateFunctionON](#) ([u8\\_t A\\_u8PortID](#), [u8\\_t A\\_u8PinID](#), [u8\\_t A\\_u8AFID](#))
- void [MGPIOx\\_vInit](#) ([MGPIOx\\_ConfigType](#) \*[A\\_xPinConfig](#))
- void [MGPIOx\\_vTogglePinValue](#) ([u8\\_t A\\_u8PortID](#), [u8\\_t A\\_u8PinID](#))

### 7.18.1 Detailed Description

This file contains the source code of the interfacing for the GPIO modules.

#### Author

Ali El Bana & Mo Alaa

#### Version

2.0

#### Date

08/22/2022

Definition in file [GPIO\\_program.c](#).

### 7.18.2 Function Documentation

#### 7.18.2.1 MGPIOx\_vLockedPins()

```
void MGPIOx_vLockedPins (
    void )
```

Locks the prohibited GPIO PINs.

Definition at line 24 of file [GPIO\\_program.c](#).

```
00025 {
00026
00027     VAR(volatile u32_t) L_u32LockGPIOA = INITIAL_ZERO ;
00028
00029     /* Lock key write sequence */
00030
00031     /* WR LCKR[16] = 010 + LCKR[13,14,15] = 010 */
00032     L_u32LockGPIOA = ( (1UL << LCKK_BIT_POS) | (GPIOA_PIN_POS) ) ;
00033
00034     GPIOA->LCKRx = L_u32LockGPIOA ;
00035
00036     /* WR LCKR[16] = 000 + LCKR[13,14,15] should not change */
00037     GPIOA->LCKRx = (GPIOA_PIN_POS) ;
00038
00039     /* WR LCKR[16] = 010 + LCKR[13,14,15] should not change */
00040     GPIOA->LCKRx = L_u32LockGPIOA ;
00041
00042     /* RD LCKR */
00043     L_u32LockGPIOA = GPIOA->LCKRx ;
00044
00045
00046
00047     VAR(volatile u32_t) L_u32LockGPIOB = INITIAL_ZERO ;
```



```

00048
00049     /* Lock key write sequence */
00050
00051     /* WR LCKR[16] = 010 + LCKR[2,3,4] = 010 */
00052     L_u32LockGPIOB = ( (1UL << LCKK_BIT_POS) | (GPIOB_PIN_POS) ) ;
00053
00054     GPIOB->LCKRx = L_u32LockGPIOB ;
00055
00056     /* WR LCKR[16] = 000 + LCKR[2,3,4] should not change*/
00057     GPIOB->LCKRx = (GPIOB_PIN_POS) ;
00058
00059     /* WR LCKR[16] = 010 + LCKR[2,3,4] should not change*/
00060     GPIOB->LCKRx = L_u32LockGPIOB ;
00061
00062     /* RD LCKR */
00063     L_u32LockGPIOB = GPIOB->LCKRx ;
00064
00065 }

```

References [GPIOA](#), [GPIOA\\_PIN\\_POS](#), [GPIOB](#), [GPIOB\\_PIN\\_POS](#), [INITIAL\\_ZERO](#), [LCKK\\_BIT\\_POS](#), and [VAR](#).

### 7.18.2.2 MGPIOX\_vSetPinMode()

```

void MGPIOX_vSetPinMode (
    u8_t A_u8PortID,
    u8_t A_u8PinID,
    u8_t A_u8Mode )

```

Definition at line 70 of file [GPIO\\_program.c](#).

```

00071 {
00072
00073     switch (A_u8PortID)
00074     {
00075         case GPIO_PORTA:
00076             CLR_BITS(GPIOA->MODERx, 0b11, A_u8PinID, 2);
00077             SET_BITS(GPIOA->MODERx, A_u8Mode, A_u8PinID, 2);
00078             break;
00079
00080         case GPIO_PORTB:
00081             CLR_BITS(GPIOB->MODERx, 0b11, A_u8PinID, 2);
00082             SET_BITS(GPIOB->MODERx, A_u8Mode, A_u8PinID, 2);
00083             break;
00084
00085         case GPIO_PORTC:
00086             CLR_BITS(GPIOC->MODERx, 0b11, A_u8PinID, 2);
00087             SET_BITS(GPIOC->MODERx, A_u8Mode, A_u8PinID, 2);
00088             break;
00089     }
00090
00091 }

```

References [CLR\\_BITS](#), [GPIO\\_PORTA](#), [GPIO\\_PORTB](#), [GPIO\\_PORTC](#), [GPIOA](#), [GPIOB](#), [GPIOC](#), and [SET\\_BITS](#).

Referenced by [MGPIOX\\_vInit\(\)](#).

### 7.18.2.3 MGPIOX\_vSetPinOutputType()

```

void MGPIOX_vSetPinOutputType (
    u8_t A_u8PortID,
    u8_t A_u8PinID,
    u8_t A_u8OutputType )

```

Definition at line 96 of file [GPIO\\_program.c](#).

```

00097 {

```

```

00098
00099     switch (A_u8OutputType)
00100     {
00101         case GPIOx_OPENDRAIN:
00102             switch (A_u8PortID)
00103             {
00104                 case GPIO_PORTA:
00105                     SET_BIT(GPIOA->OTYPERx, A_u8PinID);
00106                     break;
00107
00108                 case GPIO_PORTB:
00109                     SET_BIT(GPIOB->OTYPERx, A_u8PinID);
00110                     break;
00111
00112                 case GPIO_PORTC:
00113                     SET_BIT(GPIOC->OTYPERx, A_u8PinID);
00114                     break;
00115             }
00116             break;
00117
00118         case GPIOx_PUSH_PULL:
00119             switch (A_u8PortID)
00120             {
00121                 case GPIO_PORTA:
00122                     CLR_BIT(GPIOA->OTYPERx, A_u8PinID);
00123                     break;
00124
00125                 case GPIO_PORTB:
00126                     CLR_BIT(GPIOB->OTYPERx, A_u8PinID);
00127                     break;
00128
00129                 case GPIO_PORTC:
00130                     CLR_BIT(GPIOC->OTYPERx, A_u8PinID);
00131                     break;
00132             }
00133             break;
00134     }
00135 }
00136 }

```

References [CLR\\_BIT](#), [GPIO\\_PORTA](#), [GPIO\\_PORTB](#), [GPIO\\_PORTC](#), [GPIOA](#), [GPIOB](#), [GPIOC](#), [GPIOx\\_OPENDRAIN](#), [GPIOx\\_PUSH\\_PULL](#), and [SET\\_BIT](#).

Referenced by [MGPIOx\\_vInit\(\)](#).

#### 7.18.2.4 MGPIOx\_vSetPinOutputSpeed()

```

void MGPIOx_vSetPinOutputSpeed (
    u8_t A_u8PortID,
    u8_t A_u8PinID,
    u8_t A_u8OutputSpeed )

```

Definition at line 141 of file [GPIO\\_program.c](#).

```

00142 {
00143     switch (A_u8PortID)
00144     {
00145         case GPIO_PORTA:
00146             CLR_BITS(GPIOA->OSPEEDRx, 0b11, A_u8PinID, 2);
00147             SET_BITS(GPIOA->OSPEEDRx, A_u8OutputSpeed, A_u8PinID, 2);
00148             break;
00149
00150         case GPIO_PORTB:
00151             CLR_BITS(GPIOB->OSPEEDRx, 0b11, A_u8PinID, 2);
00152             SET_BITS(GPIOB->OSPEEDRx, A_u8OutputSpeed, A_u8PinID, 2);
00153             break;
00154
00155         case GPIO_PORTC:
00156             CLR_BITS(GPIOC->OSPEEDRx, 0b11, A_u8PinID, 2);
00157             SET_BITS(GPIOC->OSPEEDRx, A_u8OutputSpeed, A_u8PinID, 2);
00158             break;
00159     }
00160 }
00161
00162 }

```

References [CLR\\_BITS](#), [GPIO\\_PORTA](#), [GPIO\\_PORTB](#), [GPIO\\_PORTC](#), [GPIOA](#), [GPIOB](#), [GPIOC](#), and [SET\\_BITS](#).

Referenced by [MGPIOx\\_vInit\(\)](#).

### 7.18.2.5 MGPIOX\_vSetPinInputPullType()

```
void MGPIOX_vSetPinInputPullType (
    u8_t A_u8PortID,
    u8_t A_u8PinID,
    u8_t A_u8InputPullType )
```

Definition at line 167 of file [GPIO\\_program.c](#).

```
00168 {
00169
00170     switch (A_u8PortID)
00171     {
00172         case GPIO_PORTA:
00173             CLR_BITS(GPIOA->PUPDRx, 0b11, A_u8PinID, 2);
00174             SET_BITS(GPIOA->PUPDRx, A_u8InputPullType, A_u8PinID, 2);
00175             break;
00176
00177         case GPIO_PORTB:
00178             CLR_BITS(GPIOB->PUPDRx, 0b11, A_u8PinID, 2);
00179             SET_BITS(GPIOB->PUPDRx, A_u8InputPullType, A_u8PinID, 2);
00180             break;
00181
00182         case GPIO_PORTC:
00183             CLR_BITS(GPIOC->PUPDRx, 0b11, A_u8PinID, 2);
00184             SET_BITS(GPIOC->PUPDRx, A_u8InputPullType, A_u8PinID, 2);
00185             break;
00186     }
00187
00188 }
```

References [CLR\\_BITS](#), [GPIO\\_PORTA](#), [GPIO\\_PORTB](#), [GPIO\\_PORTC](#), [GPIOA](#), [GPIOB](#), [GPIOC](#), and [SET\\_BITS](#).

Referenced by [MGPIOX\\_vInit\(\)](#).

### 7.18.2.6 MGPIOX\_u8GetPinValue()

```
u8_t MGPIOX_u8GetPinValue (
    u8_t A_u8PortID,
    u8_t A_u8PinID )
```

Definition at line 193 of file [GPIO\\_program.c](#).

```
00194 {
00195
00196     VAR(u8_t) L_u8PinValue = INITIAL_ZERO;
00197
00198     switch (A_u8PortID)
00199     {
00200         case GPIO_PORTA:
00201             L_u8PinValue = GET_BIT(GPIOA->IDRx, A_u8PinID);
00202             break;
00203
00204         case GPIO_PORTB:
00205             L_u8PinValue = GET_BIT(GPIOB->IDRx, A_u8PinID);
00206             break;
00207
00208         case GPIO_PORTC:
00209             L_u8PinValue = GET_BIT(GPIOC->IDRx, A_u8PinID);
00210             break;
00211     }
00212
00213     return L_u8PinValue;
00214
00215 }
```

References [GET\\_BIT](#), [GPIO\\_PORTA](#), [GPIO\\_PORTB](#), [GPIO\\_PORTC](#), [GPIOA](#), [GPIOB](#), [GPIOC](#), [INITIAL\\_ZERO](#), and [VAR](#).

### 7.18.2.7 MGPIOR\_vSetPinValue()

```
void MGPIOR_vSetPinValue (
    u8_t A_u8PortID,
    u8_t A_u8PinID,
    u8_t A_u8PinValue )
```

Definition at line 220 of file [GPIO\\_program.c](#).

```
00221 {
00222
00223     switch (A_u8PinValue)
00224     {
00225         case GPIOx_HIGH:
00226             switch (A_u8PortID)
00227             {
00228                 case GPIO_PORTA:
00229                     SET_BIT(GPIOA->ODRx, A_u8PinID);
00230                     break;
00231
00232                 case GPIO_PORTB:
00233                     SET_BIT(GPIOB->ODRx, A_u8PinID);
00234                     break;
00235
00236                 case GPIO_PORTC:
00237                     SET_BIT(GPIOC->ODRx, A_u8PinID);
00238                     break;
00239             }
00240             break;
00241
00242         case GPIOx_LOW:
00243             switch( A_u8PortID )
00244             {
00245                 case GPIO_PORTA:
00246                     CLR_BIT(GPIOA->ODRx, A_u8PinID);
00247                     break;
00248
00249                 case GPIO_PORTB:
00250                     CLR_BIT(GPIOB->ODRx, A_u8PinID);
00251                     break;
00252
00253                 case GPIO_PORTC:
00254                     CLR_BIT(GPIOC->ODRx, A_u8PinID);
00255                     break;
00256             }
00257             break;
00258     }
00259
00260 }
```

References [CLR\\_BIT](#), [GPIO\\_PORTA](#), [GPIO\\_PORTB](#), [GPIO\\_PORTC](#), [GPIOA](#), [GPIOB](#), [GPIOC](#), [GPIOx\\_HIGH](#), [GPIOx\\_LOW](#), and [SET\\_BIT](#).

### 7.18.2.8 MGPIOR\_vSetResetAtomic()

```
void MGPIOR_vSetResetAtomic (
    u8_t A_u8PortID,
    u8_t A_u8PinID,
    u8_t A_u8SetResetPinValue )
```

Definition at line 265 of file [GPIO\\_program.c](#).

```
00266 {
00267
00268     switch (A_u8PortID)
00269     {
00270
00271         case GPIO_PORTA:
00272
00273             switch (A_u8SetResetPinValue)
00274             {
00275                 case GPIOx_HIGH:
00276                     GPIOA->BSRRx = (1 << A_u8PinID);
```

```

00277         break;
00278
00279         case GPIOx_LOW:
00280             GPIOA->BSRRx = (1 << (A_u8PinID + 16));
00281             break;
00282     }
00283
00284     break;
00285
00286     case GPIO_PORTB:
00287
00288         switch (A_u8SetResetPinValue)
00289         {
00290             case GPIOx_HIGH:
00291                 GPIOB->BSRRx = (1 << A_u8PinID);
00292                 break;
00293
00294             case GPIOx_LOW:
00295                 GPIOB->BSRRx = (1 << (A_u8PinID + 16));
00296                 break;
00297         }
00298
00299     break;
00300
00301     case GPIO_PORTC:
00302
00303         switch (A_u8SetResetPinValue)
00304         {
00305             case GPIOx_HIGH:
00306                 GPIOC->BSRRx = (1 << A_u8PinID);
00307                 break;
00308
00309             case GPIOx_LOW:
00310                 GPIOC->BSRRx = (1 << (A_u8PinID + 16));
00311                 break;
00312         }
00313
00314     break;
00315 }
00316
00317 }

```

References [GPIO\\_PORTA](#), [GPIO\\_PORTB](#), [GPIO\\_PORTC](#), [GPIOA](#), [GPIOB](#), [GPIOC](#), [GPIOx\\_HIGH](#), and [GPIOx\\_LOW](#).

### 7.18.2.9 MGPIOx\_vSetAlternateFunctionON()

```

void MGPIOx_vSetAlternateFunctionON (
    u8_t A_u8PortID,
    u8_t A_u8PinID,
    u8_t A_u8AFID )

```

Definition at line 322 of file [GPIO\\_program.c](#).

```

00323 {
00324
00325     switch (A_u8PortID)
00326     {
00327         case GPIO_PORTA:
00328             switch (A_u8PinID)
00329             {
00330                 case GPIOx_PIN0:
00331                 case GPIOx_PIN1:
00332                 case GPIOx_PIN2:
00333                 case GPIOx_PIN3:
00334                 case GPIOx_PIN4:
00335                 case GPIOx_PIN5:
00336                 case GPIOx_PIN6:
00337                 case GPIOx_PIN7:
00338                     CLR_BITS(GPIOA->AFRLx, 0b1111, A_u8PinID, 4);
00339                     SET_BITS(GPIOA->AFRLx, A_u8AFID, A_u8PinID, 4);
00340                     break;
00341
00342                 case GPIOx_PIN8:
00343                 case GPIOx_PIN9:
00344                 case GPIOx_PIN10:
00345                 case GPIOx_PIN11:

```

```

00346         case GPIOx_PIN12:
00347         case GPIOx_PIN13:
00348         case GPIOx_PIN14:
00349         case GPIOx_PIN15:
00350             CLR_BITS(GPIOA->AFRHx, 0b1111, A_u8PinID, 4);
00351             SET_BITS(GPIOA->AFRHx, A_u8AFID, A_u8PinID, 4);
00352             break;
00353     }
00354     break;
00355
00356     case GPIO_PORTB:
00357         switch (A_u8PinID)
00358         {
00359             case GPIOx_PIN0:
00360             case GPIOx_PIN1:
00361             case GPIOx_PIN2:
00362             case GPIOx_PIN3:
00363             case GPIOx_PIN4:
00364             case GPIOx_PIN5:
00365             case GPIOx_PIN6:
00366             case GPIOx_PIN7:
00367                 CLR_BITS(GPIOB->AFRLx, 0b1111, A_u8PinID, 4);
00368                 SET_BITS(GPIOB->AFRLx, A_u8AFID, A_u8PinID, 4);
00369                 break;
00370
00371             case GPIOx_PIN8:
00372             case GPIOx_PIN9:
00373             case GPIOx_PIN10:
00374             case GPIOx_PIN11:
00375             case GPIOx_PIN12:
00376             case GPIOx_PIN13:
00377             case GPIOx_PIN14:
00378             case GPIOx_PIN15:
00379                 CLR_BITS(GPIOB->AFRHx, 0b1111, A_u8PinID-8, 4);
00380                 SET_BITS(GPIOB->AFRHx, A_u8AFID, A_u8PinID-8, 4);
00381                 break;
00382         }
00383     break;
00384
00385     case GPIO_PORTC:
00386         switch (A_u8PinID)
00387         {
00388             case GPIOx_PIN0:
00389             case GPIOx_PIN1:
00390             case GPIOx_PIN2:
00391             case GPIOx_PIN3:
00392             case GPIOx_PIN4:
00393             case GPIOx_PIN5:
00394             case GPIOx_PIN6:
00395             case GPIOx_PIN7:
00396                 CLR_BITS(GPIOC->AFRLx, 0b1111, A_u8PinID, 4);
00397                 SET_BITS(GPIOC->AFRLx, A_u8AFID, A_u8PinID, 4);
00398                 break;
00399
00400             case GPIOx_PIN8:
00401             case GPIOx_PIN9:
00402             case GPIOx_PIN10:
00403             case GPIOx_PIN11:
00404             case GPIOx_PIN12:
00405             case GPIOx_PIN13:
00406             case GPIOx_PIN14:
00407             case GPIOx_PIN15:
00408                 CLR_BITS(GPIOC->AFRHx, 0b1111, A_u8PinID, 4);
00409                 SET_BITS(GPIOC->AFRHx, A_u8AFID, A_u8PinID, 4);
00410                 break;
00411         }
00412     break;
00413 }
00414
00415 }

```

References [CLR\\_BITS](#), [GPIO\\_PORTA](#), [GPIO\\_PORTB](#), [GPIO\\_PORTC](#), [GPIOA](#), [GPIOB](#), [GPIOC](#), [GPIOx\\_PIN0](#), [GPIOx\\_PIN1](#), [GPIOx\\_PIN10](#), [GPIOx\\_PIN11](#), [GPIOx\\_PIN12](#), [GPIOx\\_PIN13](#), [GPIOx\\_PIN14](#), [GPIOx\\_PIN15](#), [GPIOx\\_PIN2](#), [GPIOx\\_PIN3](#), [GPIOx\\_PIN4](#), [GPIOx\\_PIN5](#), [GPIOx\\_PIN6](#), [GPIOx\\_PIN7](#), [GPIOx\\_PIN8](#), [GPIOx\\_PIN9](#), and [SET\\_BITS](#).

Referenced by [MGPIOx\\_vInit\(\)](#).

### 7.18.2.10 MGPIOn\_vInit()

```
void MGPIOn_vInit (
    MGPIOn_ConfigType * A_xPinConfig )
```

Definition at line 420 of file [GPIO\\_program.c](#).

```
00421 {
00422
00423     MGPIOn_vSetPinMode          (A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->Mode
00424 );
00425     MGPIOn_vSetPinOutputType    (A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->OutputType
00426 );
00427     MGPIOn_vSetPinOutputSpeed   (A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->OutputSpeed
00428 );
00429     MGPIOn_vSetPinInputPullType (A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->InputType
00430 );
00431     MGPIOn_vSetAlternateFunctionON (A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->AF_Type
00432 );
00433 }
```

References [MGPIOn\\_ConfigType::AF\\_Type](#), [MGPIOn\\_ConfigType::InputType](#), [MGPIOn\\_vSetAlternateFunctionON\(\)](#), [MGPIOn\\_vSetPinInputPullType\(\)](#), [MGPIOn\\_vSetPinMode\(\)](#), [MGPIOn\\_vSetPinOutputSpeed\(\)](#), [MGPIOn\\_vSetPinOutputType\(\)](#), [MGPIOn\\_ConfigType::Mode](#), [MGPIOn\\_ConfigType::OutputSpeed](#), [MGPIOn\\_ConfigType::OutputType](#), [MGPIOn\\_ConfigType::Pin](#), and [MGPIOn\\_ConfigType::Port](#).

### 7.18.2.11 MGPIOn\_vTogglePinValue()

```
void MGPIOn_vTogglePinValue (
    u8_t A_u8PortID,
    u8_t A_u8PinID )
```

Definition at line 434 of file [GPIO\\_program.c](#).

```
00435 {
00436
00437     switch (A_u8PortID)
00438     {
00439         case GPIO_PORTA:
00440             TOGGLE_BIT(GPIOA->ODRx, A_u8PinID);
00441             break;
00442         case GPIO_PORTB:
00443             TOGGLE_BIT(GPIOB->ODRx, A_u8PinID);
00444             break;
00445         case GPIO_PORTC:
00446             TOGGLE_BIT(GPIOC->ODRx, A_u8PinID);
00447             break;
00448     }
00449 }
00450
00451
00452 }
```

References [GPIO\\_PORTA](#), [GPIO\\_PORTB](#), [GPIO\\_PORTC](#), [GPIOA](#), [GPIOB](#), [GPIOC](#), and [TOGGLE\\_BIT](#).

## 7.19 GPIO\_program.c

[Go to the documentation of this file.](#)

```
00001
00002 /*****
00003  *
00004  * Include headers
00005  *
00006  *****/
00007 #include "../LIB/LSTD_TYPES.h"
00008 #include "../LIB/LSTD_COMPILER.h"
00009 #include "../LIB/LSTD_VALUES.h"
00010 #include "../LIB/LSTD_BITMATH.h"
```

```

00016 #include "GPIO_interface.h"
00017 #include "GPIO_private.h"
00018 #include "GPIO_config.h"
00019
00020 /*****
00021  * Functions' implementations
00022  */
00023
00024 FUNC(void) MGPIOX_vLockedPins(void)
00025 {
00026     VAR(volatile u32_t) L_u32LockGPIOA = INITIAL_ZERO ;
00027
00028     /* Lock key write sequence */
00029
00030     /* WR LCKR[16] = 010 + LCKR[13,14,15] = 010 */
00031     L_u32LockGPIOA = ( (1UL << LCKK_BIT_POS) | (GPIOA_PIN_POS) ) ;
00032
00033     GPIOA->LCKRx = L_u32LockGPIOA ;
00034
00035     /* WR LCKR[16] = 000 + LCKR[13,14,15] should not change */
00036     GPIOA->LCKRx = (GPIOA_PIN_POS) ;
00037
00038     /* WR LCKR[16] = 010 + LCKR[13,14,15] should not change */
00039     GPIOA->LCKRx = L_u32LockGPIOA ;
00040
00041     /* RD LCKR */
00042     L_u32LockGPIOA = GPIOA->LCKRx ;
00043
00044     VAR(volatile u32_t) L_u32LockGPIOB = INITIAL_ZERO ;
00045
00046     /* Lock key write sequence */
00047
00048     /* WR LCKR[16] = 010 + LCKR[2,3,4] = 010 */
00049     L_u32LockGPIOB = ( (1UL << LCKK_BIT_POS) | (GPIOB_PIN_POS) ) ;
00050
00051     GPIOB->LCKRx = L_u32LockGPIOB ;
00052
00053     /* WR LCKR[16] = 000 + LCKR[2,3,4] should not change */
00054     GPIOB->LCKRx = (GPIOB_PIN_POS) ;
00055
00056     /* WR LCKR[16] = 010 + LCKR[2,3,4] should not change */
00057     GPIOB->LCKRx = L_u32LockGPIOB ;
00058
00059     /* RD LCKR */
00060     L_u32LockGPIOB = GPIOB->LCKRx ;
00061
00062     /* RD LCKR */
00063     L_u32LockGPIOB = GPIOB->LCKRx ;
00064
00065 }
00066
00067 /*****
00068  */
00069
00070 FUNC(void) MGPIOX_vSetPinMode(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t) A_u8Mode)
00071 {
00072     switch (A_u8PortID)
00073     {
00074         case GPIO_PORTA:
00075             CLR_BITS(GPIOA->MODERx, 0b11, A_u8PinID, 2);
00076             SET_BITS(GPIOA->MODERx, A_u8Mode, A_u8PinID, 2);
00077             break;
00078
00079         case GPIO_PORTB:
00080             CLR_BITS(GPIOB->MODERx, 0b11, A_u8PinID, 2);
00081             SET_BITS(GPIOB->MODERx, A_u8Mode, A_u8PinID, 2);
00082             break;
00083
00084         case GPIO_PORTC:
00085             CLR_BITS(GPIOC->MODERx, 0b11, A_u8PinID, 2);
00086             SET_BITS(GPIOC->MODERx, A_u8Mode, A_u8PinID, 2);
00087             break;
00088     }
00089
00090 }
00091
00092 /*****
00093  */
00094 /*****
00095  */
00096
00097 FUNC(void) MGPIOX_vSetPinOutputType(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t)
00098 A_u8OutputType)
00099 {
00100     switch (A_u8OutputType)
00101     {
00102         case GPIOx_OPENDRAIN:
00103             switch (A_u8PortID)

```



```

00103         {
00104             case GPIO_PORTA:
00105                 SET_BIT(GPIOA->OTYPERx, A_u8PinID);
00106                 break;
00107
00108             case GPIO_PORTB:
00109                 SET_BIT(GPIOB->OTYPERx, A_u8PinID);
00110                 break;
00111
00112             case GPIO_PORTC:
00113                 SET_BIT(GPIOC->OTYPERx, A_u8PinID);
00114                 break;
00115         }
00116         break;
00117
00118     case GPIOx_PUSHPULL:
00119         switch(A_u8PortID)
00120         {
00121             case GPIO_PORTA:
00122                 CLR_BIT(GPIOA->OTYPERx, A_u8PinID);
00123                 break;
00124
00125             case GPIO_PORTB:
00126                 CLR_BIT(GPIOB->OTYPERx, A_u8PinID);
00127                 break;
00128
00129             case GPIO_PORTC:
00130                 CLR_BIT(GPIOC->OTYPERx, A_u8PinID);
00131                 break;
00132         }
00133         break;
00134     }
00135 }
00136 }
00137
00138 /*****
00139 /*****
00140
00141 FUNC(void) MGPIOp_vSetPinOutputSpeed(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t)
A_u8OutputSpeed)
00142 {
00143
00144     switch(A_u8PortID)
00145     {
00146         case GPIO_PORTA:
00147             CLR_BITS(GPIOA->OSPEEDRx, 0b11, A_u8PinID, 2);
00148             SET_BITS(GPIOA->OSPEEDRx, A_u8OutputSpeed, A_u8PinID, 2);
00149             break;
00150
00151         case GPIO_PORTB:
00152             CLR_BITS(GPIOB->OSPEEDRx, 0b11, A_u8PinID, 2);
00153             SET_BITS(GPIOB->OSPEEDRx, A_u8OutputSpeed, A_u8PinID, 2);
00154             break;
00155
00156         case GPIO_PORTC:
00157             CLR_BITS(GPIOC->OSPEEDRx, 0b11, A_u8PinID, 2);
00158             SET_BITS(GPIOC->OSPEEDRx, A_u8OutputSpeed, A_u8PinID, 2);
00159             break;
00160     }
00161 }
00162 }
00163
00164 /*****
00165 /*****
00166
00167 FUNC(void) MGPIOp_vSetPinInputPullType(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t)
A_u8InputPullType)
00168 {
00169
00170     switch (A_u8PortID)
00171     {
00172         case GPIO_PORTA:
00173             CLR_BITS(GPIOA->PUPDRx, 0b11, A_u8PinID, 2);
00174             SET_BITS(GPIOA->PUPDRx, A_u8InputPullType, A_u8PinID, 2);
00175             break;
00176
00177         case GPIO_PORTB:
00178             CLR_BITS(GPIOB->PUPDRx, 0b11, A_u8PinID, 2);
00179             SET_BITS(GPIOB->PUPDRx, A_u8InputPullType, A_u8PinID, 2);
00180             break;
00181
00182         case GPIO_PORTC:
00183             CLR_BITS(GPIOC->PUPDRx, 0b11, A_u8PinID, 2);
00184             SET_BITS(GPIOC->PUPDRx, A_u8InputPullType, A_u8PinID, 2);
00185             break;
00186     }
00187 }

```

```

00188 }
00189
00190 /*****
00191 /*****
00192
00193 FUNC(u8_t) MGPIIx_u8GetPinValue(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID)
00194 {
00195     VAR(u8_t) L_u8PinValue = INITIAL_ZERO;
00196     switch (A_u8PortID)
00197     {
00200         case GPIO_PORTA:
00201             L_u8PinValue = GET_BIT(GPIOA->IDRx, A_u8PinID);
00202             break;
00203
00204         case GPIO_PORTB:
00205             L_u8PinValue = GET_BIT(GPIOB->IDRx, A_u8PinID);
00206             break;
00207
00208         case GPIO_PORTC:
00209             L_u8PinValue = GET_BIT(GPIOC->IDRx, A_u8PinID);
00210             break;
00211     }
00212     return L_u8PinValue;
00213 }
00214
00215 }
00216
00217 /*****
00218 /*****
00219
00220 FUNC(void) MGPIIx_vSetPinValue(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t) A_u8PinValue)
00221 {
00222     switch (A_u8PinValue)
00223     {
00224         case GPIOx_HIGH:
00225             switch (A_u8PortID)
00226             {
00227                 case GPIO_PORTA:
00228                     SET_BIT(GPIOA->ODRx, A_u8PinID);
00229                     break;
00230
00231                 case GPIO_PORTB:
00232                     SET_BIT(GPIOB->ODRx, A_u8PinID);
00233                     break;
00234
00235                 case GPIO_PORTC:
00236                     SET_BIT(GPIOC->ODRx, A_u8PinID);
00237                     break;
00238             }
00239             break;
00240
00241         case GPIOx_LOW:
00242             switch (A_u8PortID)
00243             {
00244                 case GPIO_PORTA:
00245                     CLR_BIT(GPIOA->ODRx, A_u8PinID);
00246                     break;
00247
00248                 case GPIO_PORTB:
00249                     CLR_BIT(GPIOB->ODRx, A_u8PinID);
00250                     break;
00251
00252                 case GPIO_PORTC:
00253                     CLR_BIT(GPIOC->ODRx, A_u8PinID);
00254                     break;
00255             }
00256             break;
00257     }
00258 }
00259
00260 }
00261
00262 /*****
00263 /*****
00264
00265 FUNC(void) MGPIIx_vSetResetAtomic(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t)
00266     A_u8SetResetPinValue)
00267 {
00268     switch (A_u8PortID)
00269     {
00270         case GPIO_PORTA:
00271             switch (A_u8SetResetPinValue)

```

```

00274     {
00275         case GPIOx_HIGH:
00276             GPIOA->BSRRx = (1 « A_u8PinID);
00277             break;
00278
00279         case GPIOx_LOW:
00280             GPIOA->BSRRx = (1 « (A_u8PinID + 16));
00281             break;
00282     }
00283
00284     break;
00285
00286     case GPIO_PORTB:
00287
00288         switch (A_u8SetResetPinValue)
00289         {
00290             case GPIOx_HIGH:
00291                 GPIOB->BSRRx = (1 « A_u8PinID);
00292                 break;
00293
00294             case GPIOx_LOW:
00295                 GPIOB->BSRRx = (1 « (A_u8PinID + 16));
00296                 break;
00297         }
00298
00299         break;
00300
00301     case GPIO_PORTC:
00302
00303         switch (A_u8SetResetPinValue)
00304         {
00305             case GPIOx_HIGH:
00306                 GPIOC->BSRRx = (1 « A_u8PinID);
00307                 break;
00308
00309             case GPIOx_LOW:
00310                 GPIOC->BSRRx = (1 « (A_u8PinID + 16));
00311                 break;
00312         }
00313
00314         break;
00315     }
00316 }
00317 }
00318
00319 /*****
00320 /*****
00321
00322 FUNC(void) MGPIOp_vSetAlternateFunctionON(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID, VAR(u8_t)
A_u8AFID)
00323 {
00324
00325     switch (A_u8PortID)
00326     {
00327         case GPIO_PORTA:
00328             switch (A_u8PinID)
00329             {
00330                 case GPIOx_PIN0:
00331                 case GPIOx_PIN1:
00332                 case GPIOx_PIN2:
00333                 case GPIOx_PIN3:
00334                 case GPIOx_PIN4:
00335                 case GPIOx_PIN5:
00336                 case GPIOx_PIN6:
00337                 case GPIOx_PIN7:
00338                     CLR_BITS(GPIOA->AFRLx, 0b1111, A_u8PinID, 4);
00339                     SET_BITS(GPIOA->AFRLx, A_u8AFID, A_u8PinID, 4);
00340                     break;
00341
00342                 case GPIOx_PIN8:
00343                 case GPIOx_PIN9:
00344                 case GPIOx_PIN10:
00345                 case GPIOx_PIN11:
00346                 case GPIOx_PIN12:
00347                 case GPIOx_PIN13:
00348                 case GPIOx_PIN14:
00349                 case GPIOx_PIN15:
00350                     CLR_BITS(GPIOA->AFRHx, 0b1111, A_u8PinID, 4);
00351                     SET_BITS(GPIOA->AFRHx, A_u8AFID, A_u8PinID, 4);
00352                     break;
00353             }
00354             break;
00355
00356         case GPIO_PORTB:
00357             switch (A_u8PinID)
00358             {
00359                 case GPIOx_PIN0:

```

```

00360         case GPIOx_PIN1:
00361         case GPIOx_PIN2:
00362         case GPIOx_PIN3:
00363         case GPIOx_PIN4:
00364         case GPIOx_PIN5:
00365         case GPIOx_PIN6:
00366         case GPIOx_PIN7:
00367             CLR_BITS(GPIOB->AFRLx, 0b1111, A_u8PinID, 4);
00368             SET_BITS(GPIOB->AFRLx, A_u8AFID, A_u8PinID, 4);
00369             break;
00370
00371         case GPIOx_PIN8:
00372         case GPIOx_PIN9:
00373         case GPIOx_PIN10:
00374         case GPIOx_PIN11:
00375         case GPIOx_PIN12:
00376         case GPIOx_PIN13:
00377         case GPIOx_PIN14:
00378         case GPIOx_PIN15:
00379             CLR_BITS(GPIOB->AFRHx, 0b1111, A_u8PinID-8, 4);
00380             SET_BITS(GPIOB->AFRHx, A_u8AFID, A_u8PinID-8, 4);
00381             break;
00382     }
00383     break;
00384
00385     case GPIO_PORTC:
00386         switch (A_u8PinID)
00387         {
00388             case GPIOx_PIN0:
00389             case GPIOx_PIN1:
00390             case GPIOx_PIN2:
00391             case GPIOx_PIN3:
00392             case GPIOx_PIN4:
00393             case GPIOx_PIN5:
00394             case GPIOx_PIN6:
00395             case GPIOx_PIN7:
00396                 CLR_BITS(GPIOC->AFRLx, 0b1111, A_u8PinID, 4);
00397                 SET_BITS(GPIOC->AFRLx, A_u8AFID, A_u8PinID, 4);
00398                 break;
00399
00400             case GPIOx_PIN8:
00401             case GPIOx_PIN9:
00402             case GPIOx_PIN10:
00403             case GPIOx_PIN11:
00404             case GPIOx_PIN12:
00405             case GPIOx_PIN13:
00406             case GPIOx_PIN14:
00407             case GPIOx_PIN15:
00408                 CLR_BITS(GPIOC->AFRHx, 0b1111, A_u8PinID, 4);
00409                 SET_BITS(GPIOC->AFRHx, A_u8AFID, A_u8PinID, 4);
00410                 break;
00411         }
00412     }
00413 }
00414
00415 }
00416
00417 /*****
00418 /*****
00419
00420 FUNC(void) MGPIoX_vInit(P2VAR(MGPIoX_ConfigType) A_xPinConfig)
00421 {
00422
00423     MGPIoX_vSetPinMode(A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->Mode
00424 );
00425     MGPIoX_vSetPinOutputType(A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->OutputType
00426 );
00427     MGPIoX_vSetPinOutputSpeed(A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->OutputSpeed
00428 );
00429     MGPIoX_vSetPinInputPullType(A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->InputType
00430 );
00431     MGPIoX_vSetAlternateFunctionON(A_xPinConfig->Port, A_xPinConfig->Pin, A_xPinConfig->AF_Type
00432 );
00433 }
00434
00435 /*****
00436 /*****
00437
00438 FUNC(void) MGPIoX_vTogglePinValue(VAR(u8_t) A_u8PortID, VAR(u8_t) A_u8PinID)
00439 {
00440     switch (A_u8PortID)
00441     {
00442         case GPIO_PORTA:
00443             TOGGLE_BIT(GPIOA->ODRx, A_u8PinID);
00444             break;

```

```

00442
00443         case GPIO_PORTB:
00444             TOGGLE_BIT(GPIOB->ODRx, A_u8PinID);
00445             break;
00446
00447         case GPIO_PORTC:
00448             TOGGLE_BIT(GPIOC->ODRx, A_u8PinID);
00449             break;
00450     }
00451 }
00452 }
00453
00454 /*****
00455 /*****
00456
00457
00458

```

## 7.20 src/COTS/MCAL/RCC/MRCC\_config.h File Reference

This file contains the RCC configurations.

### Macros

- #define [PLLI2S DISABLE](#)
- #define [PLL DISABLE](#)
- #define [CSS ENABLE](#)
- #define [HSEBYP NOTBYBASED](#)
- #define [HSE\\_EN DISABLE](#)
- #define [HSI\\_EN ENABLE](#)
- #define [PLLQ Equal\\_2](#)
- #define [PLLSRC HSE](#)
- #define [PLLP PLLCLK](#)
- #define [PLLN Equal\\_200](#)
- #define [PLLM Equal\\_2](#)
- #define [MCO2 SYCLK](#)
- #define [MCO2PRE NoDivision](#)
- #define [MCO1PRE NoDivision](#)
- #define [I2SSRC PLLI2SCLK](#)
- #define [MCO1 HSE](#)
- #define [RTCPRE HSEby2](#)
- #define [PPRE2 NoDivision](#)
- #define [PPRE1 NoDivision](#)
- #define [HPRE SYCLKby2](#)
- #define [SWS HSI](#)
- #define [SW HSI](#)
- #define [DMA2CLK DISABLE](#)
- #define [DMA1CLK DISABLE](#)
- #define [CRCCLK DISABLE](#)
- #define [GPIOHCLK DISABLE](#)
- #define [GPIOECLK DISABLE](#)
- #define [GPIODCLK DISABLE](#)
- #define [GPIOCCLK DISABLE](#)
- #define [GPIOBCLK DISABLE](#)
- #define [GPIOACLK DISABLE](#)
- #define [OTGFS DISABLE](#)
- #define [PWREN DISABLE](#)

- `#define I2C3EN DISABLE`
- `#define I2C2EN DISABLE`
- `#define I2C1EN DISABLE`
- `#define USART2EN DISABLE`
- `#define SPI3EN DISABLE`
- `#define SPI2EN DISABLE`
- `#define WWDGEN DISABLE`
- `#define TIM5EN DISABLE`
- `#define TIM4EN DISABLE`
- `#define TIM3EN DISABLE`
- `#define TIM2EN DISABLE`
- `#define TIM11EN DISABLE`
- `#define TIM10EN DISABLE`
- `#define TIM9EN DISABLE`
- `#define SYSCFGEN DISABLE`
- `#define SPI4EN DISABLE`
- `#define SPI1EN DISABLE`
- `#define SDIOEN DISABLE`
- `#define ADC1EN DISABLE`
- `#define USART6EN DISABLE`
- `#define USART1EN DISABLE`
- `#define TIM1EN DISABLE`

### 7.20.1 Detailed Description

This file contains the RCC configurations.

#### Author

Ali El Bana & Mo Alaa

#### Version

2.0

#### Date

11/9/2022

Definition in file [MRCC\\_config.h](#).

### 7.20.2 Macro Definition Documentation

#### 7.20.2.1 PLLI2S

```
#define PLLI2S DISABLE
```

Definition at line 21 of file [MRCC\\_config.h](#).

### 7.20.2.2 PLL

```
#define PLL DISABLE
```

Definition at line [29](#) of file [MRCC\\_config.h](#).

### 7.20.2.3 CSS

```
#define CSS ENABLE
```

Definition at line [37](#) of file [MRCC\\_config.h](#).

### 7.20.2.4 HSEBYP

```
#define HSEBYP NOTBYBASED
```

Definition at line [45](#) of file [MRCC\\_config.h](#).

### 7.20.2.5 HSE\_EN

```
#define HSE_EN DISABLE
```

Definition at line [53](#) of file [MRCC\\_config.h](#).

### 7.20.2.6 HSI\_EN

```
#define HSI_EN ENABLE
```

Definition at line [61](#) of file [MRCC\\_config.h](#).

### 7.20.2.7 PLLQ

```
#define PLLQ Equal\_2
```

Definition at line [84](#) of file [MRCC\\_config.h](#).

#### 7.20.2.8 PLLSRC

```
#define PLLSRC HSE
```

Definition at line 92 of file [MRCC\\_config.h](#).

#### 7.20.2.9 PLLP

```
#define PLLP PLLCLK
```

Definition at line 102 of file [MRCC\\_config.h](#).

#### 7.20.2.10 PLLN

```
#define PLLN Equal_200
```

Definition at line 173 of file [MRCC\\_config.h](#).

#### 7.20.2.11 PLLM

```
#define PLLM Equal_2
```

Definition at line 240 of file [MRCC\\_config.h](#).

#### 7.20.2.12 MCO2

```
#define MCO2 SYSCLK
```

Definition at line 253 of file [MRCC\\_config.h](#).

#### 7.20.2.13 MCO2PRE

```
#define MCO2PRE NoDivision
```

Definition at line 264 of file [MRCC\\_config.h](#).



#### 7.20.2.14 MCO1PRE

```
#define MCO1PRE NoDivision
```

Definition at line 275 of file [MRCC\\_config.h](#).

#### 7.20.2.15 I2SSRC

```
#define I2SSRC PLLI2SCLK
```

Definition at line 283 of file [MRCC\\_config.h](#).

#### 7.20.2.16 MCO1

```
#define MCO1 HSE
```

Definition at line 293 of file [MRCC\\_config.h](#).

#### 7.20.2.17 RTCPRE

```
#define RTCPRE HSEby2
```

Definition at line 332 of file [MRCC\\_config.h](#).

#### 7.20.2.18 PPRE2

```
#define PPRE2 NoDivision
```

Definition at line 343 of file [MRCC\\_config.h](#).

#### 7.20.2.19 PPRE1

```
#define PPRE1 NoDivision
```

Definition at line 354 of file [MRCC\\_config.h](#).

#### 7.20.2.20 HPRE

```
#define HPRE SYSCLKby2
```

Definition at line 369 of file [MRCC\\_config.h](#).

#### 7.20.2.21 SWS

```
#define SWS HSI
```

Definition at line 378 of file [MRCC\\_config.h](#).

#### 7.20.2.22 SW

```
#define SW HSI
```

Definition at line 387 of file [MRCC\\_config.h](#).

#### 7.20.2.23 DMA2CLK

```
#define DMA2CLK DISABLE
```

Definition at line 398 of file [MRCC\\_config.h](#).

#### 7.20.2.24 DMA1CLK

```
#define DMA1CLK DISABLE
```

Definition at line 406 of file [MRCC\\_config.h](#).

#### 7.20.2.25 CRCCLK

```
#define CRCCLK DISABLE
```

Definition at line 414 of file [MRCC\\_config.h](#).

### 7.20.2.26 GPIOHCLK

```
#define GPIOHCLK DISABLE
```

Definition at line 422 of file [MRCC\\_config.h](#).

### 7.20.2.27 GPIOECLK

```
#define GPIOECLK DISABLE
```

Definition at line 430 of file [MRCC\\_config.h](#).

### 7.20.2.28 GPIODCLK

```
#define GPIODCLK DISABLE
```

Definition at line 438 of file [MRCC\\_config.h](#).

### 7.20.2.29 GPIOCCLK

```
#define GPIOCCLK DISABLE
```

Definition at line 446 of file [MRCC\\_config.h](#).

### 7.20.2.30 GPIOBCLK

```
#define GPIOBCLK DISABLE
```

Definition at line 454 of file [MRCC\\_config.h](#).

### 7.20.2.31 GPIOACLK

```
#define GPIOACLK DISABLE
```

Definition at line 462 of file [MRCC\\_config.h](#).

#### 7.20.2.32 OTGFS

```
#define OTGFS DISABLE
```

Definition at line 473 of file [MRCC\\_config.h](#).

#### 7.20.2.33 PWREN

```
#define PWREN DISABLE
```

Definition at line 484 of file [MRCC\\_config.h](#).

#### 7.20.2.34 I2C3EN

```
#define I2C3EN DISABLE
```

Definition at line 492 of file [MRCC\\_config.h](#).

#### 7.20.2.35 I2C2EN

```
#define I2C2EN DISABLE
```

Definition at line 500 of file [MRCC\\_config.h](#).

#### 7.20.2.36 I2C1EN

```
#define I2C1EN DISABLE
```

Definition at line 508 of file [MRCC\\_config.h](#).

#### 7.20.2.37 USART2EN

```
#define USART2EN DISABLE
```

Definition at line 516 of file [MRCC\\_config.h](#).

### 7.20.2.38 SPI3EN

```
#define SPI3EN DISABLE
```

Definition at line 524 of file [MRCC\\_config.h](#).

### 7.20.2.39 SPI2EN

```
#define SPI2EN DISABLE
```

Definition at line 532 of file [MRCC\\_config.h](#).

### 7.20.2.40 WWDGEN

```
#define WWDGEN DISABLE
```

Definition at line 540 of file [MRCC\\_config.h](#).

### 7.20.2.41 TIM5EN

```
#define TIM5EN DISABLE
```

Definition at line 548 of file [MRCC\\_config.h](#).

### 7.20.2.42 TIM4EN

```
#define TIM4EN DISABLE
```

Definition at line 556 of file [MRCC\\_config.h](#).

### 7.20.2.43 TIM3EN

```
#define TIM3EN DISABLE
```

Definition at line 564 of file [MRCC\\_config.h](#).

#### 7.20.2.44 TIM2EN

```
#define TIM2EN DISABLE
```

Definition at line 572 of file [MRCC\\_config.h](#).

#### 7.20.2.45 TIM11EN

```
#define TIM11EN DISABLE
```

Definition at line 583 of file [MRCC\\_config.h](#).

#### 7.20.2.46 TIM10EN

```
#define TIM10EN DISABLE
```

Definition at line 591 of file [MRCC\\_config.h](#).

#### 7.20.2.47 TIM9EN

```
#define TIM9EN DISABLE
```

Definition at line 599 of file [MRCC\\_config.h](#).

#### 7.20.2.48 SYSCFGEN

```
#define SYSCFGEN DISABLE
```

Definition at line 607 of file [MRCC\\_config.h](#).

#### 7.20.2.49 SPI4EN

```
#define SPI4EN DISABLE
```

Definition at line 615 of file [MRCC\\_config.h](#).

#### 7.20.2.50 SPI1EN

```
#define SPI1EN DISABLE
```

Definition at line 623 of file [MRCC\\_config.h](#).

#### 7.20.2.51 SDIOEN

```
#define SDIOEN DISABLE
```

Definition at line 631 of file [MRCC\\_config.h](#).

#### 7.20.2.52 ADC1EN

```
#define ADC1EN DISABLE
```

Definition at line 639 of file [MRCC\\_config.h](#).

#### 7.20.2.53 USART6EN

```
#define USART6EN DISABLE
```

Definition at line 647 of file [MRCC\\_config.h](#).

#### 7.20.2.54 USART1EN

```
#define USART1EN DISABLE
```

Definition at line 655 of file [MRCC\\_config.h](#).

#### 7.20.2.55 TIM1EN

```
#define TIM1EN DISABLE
```

Definition at line 663 of file [MRCC\\_config.h](#).

## 7.21 MRCC\_config.h

[Go to the documentation of this file.](#)

```

00001
00009 #ifndef MCAL_RCC_MRCC_CONFIG_H_
00010 #define MCAL_RCC_MRCC_CONFIG_H_
00011
00012
00013 /*****
00014  /*
00015  *****/
00016
00017 /*options:
00018 *ENABLE
00019 *DISABLE
00020 */
00021 #define PLLI2S_DISABLE
00022
00023 /*****
00024  /*
00025  *****/
00026
00027 /*options:
00028 *ENABLE
00029 *DISABLE
00030 */
00031 #define PLL_DISABLE
00032
00033 /*****
00034  /*
00035  *****/
00036
00037 #define CSS_ENABLE
00038
00039 /*****
00040  /*
00041  *****/
00042
00043 /*options:
00044 *BYBASED
00045 *NOTBYBASED
00046 */
00047 #define HSEBYP_NOTBYBASED
00048
00049 /*****
00050  /*
00051  *****/
00052
00053 #define HSE_EN_DISABLE
00054
00055 /*****
00056  /*
00057  *****/
00058
00059 /*options:
00060 *ENABLE
00061 *DISABLE
00062 */
00063 #define HSI_EN_ENABLE
00064
00065 /*****
00066  /*
00067  *****/
00068
00069 /*options:
00070 *Equal_2
00071 *Equal_3
00072 *Equal_4
00073 *Equal_5
00074 *Equal_6
00075 *Equal_7
00076 *Equal_8
00077 *Equal_9
00078 *Equal_10
00079 *Equal_11
00080 *Equal_12
00081 *Equal_13
00082 *Equal_14
00083 *Equal_15
00084 */
00085 #define PLLQ_Equal_2
00086
00087 /*****
00088  /*
00089  *****/
00090
00091 /*options:
00092 *HSI

```



```
00090 *HSE
00091 */
00092 #define PLLSRC HSE
00093
00094 /*****
00095
00096 */options:
00097 *Equal_2
00098 *Equal_4
00099 *Equal_6
00100 *Equal_8
00101 */
00102 #define PLLP PLLCLK
00103
00104 /*****
00105
00106 */options:
00107 *Equal_192
00108 *Equal_193
00109 *Equal_194
00110 *Equal_195
00111 *Equal_196
00112 *Equal_197
00113 *Equal_198
00114 *Equal_199
00115 *Equal_200
00116 *Equal_201
00117 *Equal_202
00118 *Equal_203
00119 *Equal_204
00120 *Equal_205
00121 *Equal_206
00122 *Equal_207
00123 *Equal_208
00124 *Equal_209
00125 *Equal_210
00126
00127 *Equal_300
00128 *Equal_301
00129 *Equal_302
00130 *Equal_303
00131 *Equal_304
00132 *Equal_305
00133 *Equal_306
00134 *Equal_307
00135 *Equal_308
00136 *Equal_309
00137 *Equal_310
00138
00139 *Equal_400
00140 *Equal_401
00141 *Equal_402
00142 *Equal_403
00143 *Equal_404
00144 *Equal_405
00145 *Equal_406
00146 *Equal_407
00147 *Equal_408
00148 *Equal_409
00149 *Equal_410
00150 *Equal_411
00151 *Equal_412
00152 *Equal_413
00153 *Equal_414
00154 *Equal_415
00155 *Equal_416
00156 *Equal_417
00157 *Equal_418
00158 *Equal_419
00159 *Equal_420
00160 *Equal_421
00161 *Equal_422
00162 *Equal_423
00163 *Equal_424
00164 *Equal_425
00165 *Equal_426
00166 *Equal_427
00167 *Equal_428
00168 *Equal_429
00169 *Equal_430
00170 *Equal_431
00171 *Equal_432
00172 */
00173 #define PLLN Equal_200
00174
00175 /*****
00176
```

```
00177 /*options:
00178 *Equal_2
00179 *Equal_4
00180 *Equal_5
00181 *Equal_6
00182 *Equal_7
00183 *Equal_8
00184 *Equal_9
00185 *Equal_10
00186 *Equal_11
00187 *Equal_12
00188 *Equal_13
00189 *Equal_14
00190 *Equal_15
00191 *Equal_16
00192 *Equal_17
00193 *Equal_18
00194 *Equal_19
00195 *Equal_20
00196 *Equal_21
00197 *Equal_22
00198 *Equal_23
00199 *Equal_24
00200 *Equal_25
00201 *Equal_26
00202 *Equal_27
00203 *Equal_28
00204 *Equal_29
00205 *Equal_30
00206 *Equal_31
00207 *Equal_32
00208 *Equal_33
00209 *Equal_34
00210 *Equal_35
00211 *Equal_36
00212 *Equal_37
00213 *Equal_38
00214 *Equal_39
00215 *Equal_40
00216 *Equal_41
00217 *Equal_42
00218 *Equal_43
00219 *Equal_44
00220 *Equal_45
00221 *Equal_46
00222 *Equal_47
00223 *Equal_48
00224 *Equal_49
00225 *Equal_50
00226 *Equal_51
00227 *Equal_52
00228 *Equal_53
00229 *Equal_54
00230 *Equal_55
00231 *Equal_56
00232 *Equal_57
00233 *Equal_58
00234 *Equal_59
00235 *Equal_60
00236 *Equal_61
00237 *Equal_62
00238 *Equal_63
00239 */
00240 #define PLLM Equal_2
00241
00242
00243 /*****
00244 // RCC_CFGR configurations //
00245 *****/
00246
00247 /*options:
00248 *SYSCLK
00249 *PLLI2SCLK
00250 *HSE
00251 *PLLCLK
00252 */
00253 #define MCO2 SYSCLK
00254
00255 /*****
00256
00257 /*options:
00258 *NoDivision
00259 *DivisionBy2
00260 *DivisionBy3
00261 *DivisionBy4
00262 *DivisionBy5
00263 */
```

```
00264 #define MCO2PRE NoDivision
00265
00266 /*****
00267
00268 */options:
00269 *NoDivision
00270 *DivisionBy2
00271 *DivisionBy3
00272 *DivisionBy4
00273 *DivisionBy5
00274 */
00275 #define MCO1PRE NoDivision
00276
00277 /*****
00278
00279 */options:
00280 *PLLI2SCLK
00281 *I2S_CKIN
00282 */
00283 #define I2SSRC PLLI2SCLK
00284
00285 /*****
00286
00287 */options:
00288 *HSI
00289 *LSE
00290 *HSE
00291 *PLLCLK
00292 */
00293 #define MCO1 HSE
00294
00295 /*****
00296
00297 */options:
00298 *NoCLK0
00299 *NoCLK1
00300 *HSEby2
00301 *HSEby3
00302 *HSEby4
00303 *HSEby5
00304 *HSEby6
00305 *HSEby7
00306 *HSEby8
00307 *HSEby9
00308 *HSEby10
00309 *HSEby11
00310 *HSEby12
00311 *HSEby13
00312 *HSEby14
00313 *HSEby15
00314 *HSEby16
00315 *HSEby17
00316 *HSEby18
00317 *HSEby19
00318 *HSEby20
00319 *HSEby21
00320 *HSEby22
00321 *HSEby22
00322 *HSEby23
00323 *HSEby24
00324 *HSEby25
00325 *HSEby26
00326 *HSEby27
00327 *HSEby28
00328 *HSEby29
00329 *HSEby30
00330 *HSEby31
00331 */
00332 #define RTCPRE HSEby2
00333
00334 /*****
00335
00336 */options:
00337 *NoDivision
00338 *AHBby2
00339 *AHBby4
00340 *AHBby8
00341 *AHBby16
00342 */
00343 #define PPRE2 NoDivision
00344
00345 /*****
00346
00347 */options:
00348 *NoDivision
00349 *AHBby2
00350 *AHBby4
```

```
00351 *AHBby8
00352 *AHBby16
00353 */
00354 #define PPRE1 NoDivision
00355
00356 /*****
00357
00358 */*options:
00359 *NoDivision
00360 *SYSCLKby2
00361 *SYSCLKby4
00362 *SYSCLKby8
00363 *SYSCLKby16
00364 *SYSCLKby64
00365 *SYSCLKby128
00366 *SYSCLKby256
00367 *SYSCLKby512
00368 */
00369 #define HPRE SYSCLKby2
00370
00371 /*****
00372
00373 */*options:
00374 *HSI
00375 *HSE
00376 *PLLCLK
00377 */
00378 #define SWS HSI
00379
00380 /*****
00381
00382 */*options:
00383 *HSI
00384 *HSE
00385 *PLLCLK
00386 */
00387 #define SW HSI
00388
00389
00390 /*****
00391                                     // RCC_AHB1ENR configurations //
00392 /*****
00393
00394 */*options:
00395 *ENABLE
00396 *DISABLE
00397 */
00398 #define DMA2CLK DISABLE
00399
00400 /*****
00401
00402 */*options:
00403 *ENABLE
00404 *DISABLE
00405 */
00406 #define DMA1CLK DISABLE
00407
00408 /*****
00409
00410 */*options:
00411 *ENABLE
00412 *DISABLE
00413 */
00414 #define CRCCLK DISABLE
00415
00416 /*****
00417
00418 */*options:
00419 *ENABLE
00420 *DISABLE
00421 */
00422 #define GPIOHCLK DISABLE
00423
00424 /*****
00425
00426 */*options:
00427 *ENABLE
00428 *DISABLE
00429 */
00430 #define GPIOECLK DISABLE
00431
00432 /*****
00433
00434 */*options:
00435 *ENABLE
00436 *DISABLE
00437 */
```

```
00438 #define GPIODCLK DISABLE
00439
00440 /*****
00441
00442 */options:
00443 *ENABLE
00444 *DISABLE
00445 */
00446 #define GPIOCCLK DISABLE
00447
00448 /*****
00449
00450 */options:
00451 *ENABLE
00452 *DISABLE
00453 */
00454 #define GPIOBCLK DISABLE
00455
00456 /*****
00457
00458 */options:
00459 *ENABLE
00460 *DISABLE
00461 */
00462 #define GPIOACLK DISABLE
00463
00464 /*****
00465 // RCC_AHB2ENR configurations //
00466 //
00467 //
00468 //
00469 */options:
00470 *ENABLE
00471 *DISABLE
00472 */
00473 #define OTGFS DISABLE
00474
00475 /*****
00476 // RCC_APB1ENR configurations //
00477 //
00478 //
00479 //
00480 */options:
00481 *ENABLE
00482 *DISABLE
00483 */
00484 #define PWREN DISABLE
00485
00486 /*****
00487
00488 */options:
00489 *ENABLE
00490 *DISABLE
00491 */
00492 #define I2C3EN DISABLE
00493
00494 /*****
00495
00496 */options:
00497 *ENABLE
00498 *DISABLE
00499 */
00500 #define I2C2EN DISABLE
00501
00502 /*****
00503
00504 */options:
00505 *ENABLE
00506 *DISABLE
00507 */
00508 #define I2C1EN DISABLE
00509
00510 /*****
00511
00512 */options:
00513 *ENABLE
00514 *DISABLE
00515 */
00516 #define USART2EN DISABLE
00517
00518 /*****
00519
00520 */options:
00521 *ENABLE
00522 *DISABLE
00523 */
00524 #define SPI3EN DISABLE
```

```
00525
00526 /*****
00527
00528 */options:
00529 *ENABLE
00530 *DISABLE
00531 */
00532 #define SPI2EN DISABLE
00533
00534 /*****
00535
00536 */options:
00537 *ENABLE
00538 *DISABLE
00539 */
00540 #define WWDGEN DISABLE
00541
00542 /*****
00543
00544 */options:
00545 *ENABLE
00546 *DISABLE
00547 */
00548 #define TIM5EN DISABLE
00549
00550 /*****
00551
00552 */options:
00553 *ENABLE
00554 *DISABLE
00555 */
00556 #define TIM4EN DISABLE
00557
00558 /*****
00559
00560 */options:
00561 *ENABLE
00562 *DISABLE
00563 */
00564 #define TIM3EN DISABLE
00565
00566 /*****
00567
00568 */options:
00569 *ENABLE
00570 *DISABLE
00571 */
00572 #define TIM2EN DISABLE
00573
00574
00575 /*****
00576 // RCC_APB2ENR configurations //
00577 /*****
00578
00579 */options:
00580 *ENABLE
00581 *DISABLE
00582 */
00583 #define TIM11EN DISABLE
00584
00585 /*****
00586
00587 */options:
00588 *ENABLE
00589 *DISABLE
00590 */
00591 #define TIM10EN DISABLE
00592
00593 /*****
00594
00595 */options:
00596 *ENABLE
00597 *DISABLE
00598 */
00599 #define TIM9EN DISABLE
00600
00601 /*****
00602
00603 */options:
00604 *ENABLE
00605 *DISABLE
00606 */
00607 #define SYSCFGEN DISABLE
00608
00609 /*****
00610
00611 */options:
```

```

00612 *ENABLE
00613 *DISABLE
00614 */
00615 #define SPI4EN_DISABLE
00616
00617 /*****
00618
00619 */options:
00620 *ENABLE
00621 *DISABLE
00622 */
00623 #define SPI1EN_DISABLE
00624
00625 /*****
00626
00627 */options:
00628 *ENABLE
00629 *DISABLE
00630 */
00631 #define SDIOEN_DISABLE
00632
00633 /*****
00634
00635 */options:
00636 *ENABLE
00637 *DISABLE
00638 */
00639 #define ADC1EN_DISABLE
00640
00641 /*****
00642
00643 */options:
00644 *ENABLE
00645 *DISABLE
00646 */
00647 #define USART6EN_DISABLE
00648
00649 /*****
00650
00651 */options:
00652 *ENABLE
00653 *DISABLE
00654 */
00655 #define USART1EN_DISABLE
00656
00657 /*****
00658
00659 */options:
00660 *ENABLE
00661 *DISABLE
00662 */
00663 #define TIM1EN_DISABLE
00664
00665 /*****
00666
00667
00668
00669
00670
00671
00672
00673
00674
00675
00676
00677
00678
00679
00680
00681
00682 #endif /* MCAL_RCC_MRCC_CONFIG_H_ */

```

## 7.22 src/COTS/MCAL/RCC/MRCC\_interface.h File Reference

This file contains the interfacing information for the RCC module.

### Macros

- #define [RCC\\_AHB1](#) 1

- #define [RCC\\_AHB2](#) 2
- #define [RCC\\_APB1](#) 3
- #define [RCC\\_APB2](#) 4
- #define [RCC\\_AHB1LPENR](#) 5
- #define [AHB1ENR\\_DMA2EN](#) 22
- #define [AHB1ENR\\_DMA1EN](#) 21
- #define [AHB1ENR\\_CRCEN](#) 12
- #define [AHB1ENR\\_GPIOHEN](#) 7
- #define [AHB1ENR\\_GPIOEEN](#) 4
- #define [AHB1ENR\\_GPIODEN](#) 3
- #define [AHB1ENR\\_GPIOCEN](#) 2
- #define [AHB1ENR\\_GPIOBEN](#) 1
- #define [AHB1ENR\\_GPIOAEN](#) 0
- #define [AHB2ENR\\_OTGFSEN](#) 7
- #define [APB1ENR\\_PWREN](#) 28
- #define [APB1ENR\\_I2C3EN](#) 23
- #define [APB1ENR\\_I2C2EN](#) 22
- #define [APB1ENR\\_I2C1EN](#) 21
- #define [APB1ENR\\_USART2EN](#) 17
- #define [APB1ENR\\_SPI3EN](#) 15
- #define [APB1ENR\\_SPI2EN](#) 14
- #define [APB1ENR\\_WWDGEN](#) 11
- #define [APB1ENR\\_TIM5EN](#) 3
- #define [APB1ENR\\_TIM4EN](#) 2
- #define [APB1ENR\\_TIM3EN](#) 1
- #define [APB1ENR\\_TIM2EN](#) 0
- #define [APB2ENR\\_TIM11EN](#) 18
- #define [APB2ENR\\_TIM10EN](#) 17
- #define [APB2ENR\\_TIM9EN](#) 16
- #define [APB2ENR\\_SYSCFGEN](#) 14
- #define [APB2ENR\\_SPI4EN](#) 13
- #define [APB2ENR\\_SPI1EN](#) 12
- #define [APB2ENR\\_SDIOEN](#) 11
- #define [APB2ENR\\_ADC1EN](#) 8
- #define [APB2ENR\\_USART6EN](#) 5
- #define [APB2ENR\\_USART1EN](#) 4
- #define [APB2ENR\\_TIM1EN](#) 0
- #define [AHB1LPENR\\_FLITFLPEN](#) 15

## Functions

- void [MRCC\\_vInit](#) (void)
- void [MRCC\\_vEnablePeriphralCLK](#) ([u32\\_t](#) A\_u32BusID, [u32\\_t](#) A\_u32PeriphralID)
- void [MRCC\\_vDisablePeriphralCLK](#) ([u32\\_t](#) A\_u32BusID, [u32\\_t](#) A\_u32PeriphralID)



### 7.22.1 Detailed Description

This file contains the interfacing information for the RCC module.

**Author**

Ali El Bana & Mo Alaa

**Version**

2.0

**Date**

11/9/2022

Definition in file [MRCC\\_interface.h](#).

### 7.22.2 Macro Definition Documentation

#### 7.22.2.1 RCC\_AHB1

```
#define RCC_AHB1 1
```

Definition at line 31 of file [MRCC\\_interface.h](#).

#### 7.22.2.2 RCC\_AHB2

```
#define RCC_AHB2 2
```

Definition at line 32 of file [MRCC\\_interface.h](#).

#### 7.22.2.3 RCC\_APB1

```
#define RCC_APB1 3
```

Definition at line 33 of file [MRCC\\_interface.h](#).

#### 7.22.2.4 RCC\_APB2

```
#define RCC_APB2 4
```

Definition at line 34 of file [MRCC\\_interface.h](#).

#### 7.22.2.5 RCC\_AHB1LPENR

```
#define RCC_AHB1LPENR 5
```

Definition at line 35 of file [MRCC\\_interface.h](#).

#### 7.22.2.6 AHB1ENR\_DMA2EN

```
#define AHB1ENR_DMA2EN 22
```

Definition at line 38 of file [MRCC\\_interface.h](#).

#### 7.22.2.7 AHB1ENR\_DMA1EN

```
#define AHB1ENR_DMA1EN 21
```

Definition at line 39 of file [MRCC\\_interface.h](#).

#### 7.22.2.8 AHB1ENR\_CRCEN

```
#define AHB1ENR_CRCEN 12
```

Definition at line 40 of file [MRCC\\_interface.h](#).

#### 7.22.2.9 AHB1ENR\_GPIOHEN

```
#define AHB1ENR_GPIOHEN 7
```

Definition at line 41 of file [MRCC\\_interface.h](#).

#### 7.22.2.10 AHB1ENR\_GPIOEEN

```
#define AHB1ENR_GPIOEEN 4
```

Definition at line 42 of file [MRCC\\_interface.h](#).

#### 7.22.2.11 AHB1ENR\_GPIODEN

```
#define AHB1ENR_GPIODEN 3
```

Definition at line 43 of file [MRCC\\_interface.h](#).

#### 7.22.2.12 AHB1ENR\_GPIOCEN

```
#define AHB1ENR_GPIOCEN 2
```

Definition at line 44 of file [MRCC\\_interface.h](#).

#### 7.22.2.13 AHB1ENR\_GPIOBEN

```
#define AHB1ENR_GPIOBEN 1
```

Definition at line 45 of file [MRCC\\_interface.h](#).

#### 7.22.2.14 AHB1ENR\_GPIOAEN

```
#define AHB1ENR_GPIOAEN 0
```

Definition at line 46 of file [MRCC\\_interface.h](#).

#### 7.22.2.15 AHB2ENR\_OTGFSEN

```
#define AHB2ENR_OTGFSEN 7
```

Definition at line 48 of file [MRCC\\_interface.h](#).

#### 7.22.2.16 APB1ENR\_PWREN

```
#define APB1ENR_PWREN 28
```

Definition at line 50 of file [MRCC\\_interface.h](#).

#### 7.22.2.17 APB1ENR\_I2C3EN

```
#define APB1ENR_I2C3EN 23
```

Definition at line 51 of file [MRCC\\_interface.h](#).

#### 7.22.2.18 APB1ENR\_I2C2EN

```
#define APB1ENR_I2C2EN 22
```

Definition at line 52 of file [MRCC\\_interface.h](#).

#### 7.22.2.19 APB1ENR\_I2C1EN

```
#define APB1ENR_I2C1EN 21
```

Definition at line 53 of file [MRCC\\_interface.h](#).

#### 7.22.2.20 APB1ENR\_USART2EN

```
#define APB1ENR_USART2EN 17
```

Definition at line 54 of file [MRCC\\_interface.h](#).

#### 7.22.2.21 APB1ENR\_SPI3EN

```
#define APB1ENR_SPI3EN 15
```

Definition at line 55 of file [MRCC\\_interface.h](#).

#### 7.22.2.22 APB1ENR\_SPI2EN

```
#define APB1ENR_SPI2EN 14
```

Definition at line 56 of file [MRCC\\_interface.h](#).

#### 7.22.2.23 APB1ENR\_WWDGEN

```
#define APB1ENR_WWDGEN 11
```

Definition at line 57 of file [MRCC\\_interface.h](#).

#### 7.22.2.24 APB1ENR\_TIM5EN

```
#define APB1ENR_TIM5EN 3
```

Definition at line 58 of file [MRCC\\_interface.h](#).

#### 7.22.2.25 APB1ENR\_TIM4EN

```
#define APB1ENR_TIM4EN 2
```

Definition at line 59 of file [MRCC\\_interface.h](#).

#### 7.22.2.26 APB1ENR\_TIM3EN

```
#define APB1ENR_TIM3EN 1
```

Definition at line 60 of file [MRCC\\_interface.h](#).

#### 7.22.2.27 APB1ENR\_TIM2EN

```
#define APB1ENR_TIM2EN 0
```

Definition at line 61 of file [MRCC\\_interface.h](#).

#### 7.22.2.28 APB2ENR\_TIM11EN

```
#define APB2ENR_TIM11EN 18
```

Definition at line 63 of file [MRCC\\_interface.h](#).

#### 7.22.2.29 APB2ENR\_TIM10EN

```
#define APB2ENR_TIM10EN 17
```

Definition at line 64 of file [MRCC\\_interface.h](#).

#### 7.22.2.30 APB2ENR\_TIM9EN

```
#define APB2ENR_TIM9EN 16
```

Definition at line 65 of file [MRCC\\_interface.h](#).

#### 7.22.2.31 APB2ENR\_SYSCFGEN

```
#define APB2ENR_SYSCFGEN 14
```

Definition at line 66 of file [MRCC\\_interface.h](#).

#### 7.22.2.32 APB2ENR\_SPI4EN

```
#define APB2ENR_SPI4EN 13
```

Definition at line 67 of file [MRCC\\_interface.h](#).

#### 7.22.2.33 APB2ENR\_SPI1EN

```
#define APB2ENR_SPI1EN 12
```

Definition at line 68 of file [MRCC\\_interface.h](#).

#### 7.22.2.34 APB2ENR\_SDIOEN

```
#define APB2ENR_SDIOEN 11
```

Definition at line 69 of file [MRCC\\_interface.h](#).

#### 7.22.2.35 APB2ENR\_ADC1EN

```
#define APB2ENR_ADC1EN 8
```

Definition at line 70 of file [MRCC\\_interface.h](#).

#### 7.22.2.36 APB2ENR\_USART6EN

```
#define APB2ENR_USART6EN 5
```

Definition at line 71 of file [MRCC\\_interface.h](#).

#### 7.22.2.37 APB2ENR\_USART1EN

```
#define APB2ENR_USART1EN 4
```

Definition at line 72 of file [MRCC\\_interface.h](#).

#### 7.22.2.38 APB2ENR\_TIM1EN

```
#define APB2ENR_TIM1EN 0
```

Definition at line 73 of file [MRCC\\_interface.h](#).

#### 7.22.2.39 AHB1LPENR\_FLITFLPEN

```
#define AHB1LPENR_FLITFLPEN 15
```

Definition at line 77 of file [MRCC\\_interface.h](#).

## 7.22.3 Function Documentation

### 7.22.3.1 MRCC\_vInit()

```
void MRCC_vInit (
    void )
```

Definition at line 29 of file [MRCC\\_program.c](#).

```
00030 {
00031
00032     // PLLI2S (ON/OFF).
00033     #if PLLI2S == ENABLE
00034
00035         SET_BIT( RCC->CR, RCC_CR_PLLI2SON ) ;
00036
00037     #elif PLLI2S == DISABLE
00038
00039         CLR_BIT( RCC->CR, RCC_CR_PLLI2SON ) ;
00040
00041     #endif
00042
00043
00044     // PLL (ON/OFF).
00045     #if PLL == ENABLE
00046
00047         SET_BIT( RCC->CR, RCC_CR_PLLON ) ;
00048
00049     #elif PLL == DISABLE
00050
00051         CLR_BIT( RCC->CR, RCC_CR_PLLON ) ;
00052
00053     #endif
00054
00055
00056     // CSS (ON/OFF).
00057     #if CSS == ENABLE
00058
00059         SET_BIT( RCC->CR, RCC_CR_CSSON ) ;
00060
00061     #elif CSS == DISABLE
00062
00063         CLR_BIT( RCC->CR, RCC_CR_CSSON ) ;
00064
00065     #endif
00066
00067
00068     // HSEBYP.
00069     #if HSEBYP == BYBASED
00070
00071         SET_BIT( RCC->CR, RCC_CR_HSEBYP ) ;
00072
00073     #elif HSEBYP == NOTBYBASED
00074
00075         CLR_BIT( RCC->CR, RCC_CR_HSEBYP ) ;
00076
00077     #endif
00078
00079
00080     // Select CLK switch (HSI/HSE/PLL).
00081     #if SW == HSI
00082
00083         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b0 ) ;
00084         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b1 ) ;
00085
00086     #elif SW == HSE
00087
00088         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b0 ) ;
00089         SET_BIT( RCC->CFGR, RCC_CFGR_SW_b1 ) ;
00090
00091     #elif SW == PLLCLK
00092
00093         SET_BIT( RCC->CFGR, RCC_CFGR_SW_b0 ) ;
00094         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b1 ) ;
00095
00096     #endif
00097
00098 }
```



```
00099      // MCO2 selection:
00100 #if MCO2 == SYSCLK
00101
00102     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2_b0 ) ;
00103     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2_b1 ) ;
00104
00105 #elif MCO2 == PLLI2SCLK
00106
00107     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2_b0 ) ;
00108     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b1 ) ;
00109
00110 #elif MCO2 == HSE
00111
00112     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b0 ) ;
00113     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2_b1 ) ;
00114
00115 #elif MCO2 == PLLCLK
00116
00117     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b0 ) ;
00118     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b1 ) ;
00119
00120 #endif
00121
00122      // MCO2 prescaler:
00123 #if MCO2PRE == NoDivision
00124
00125     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00126     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00127     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00128
00129 #elif MCO2PRE == DivisionBy2
00130
00131     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00132     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00133     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00134
00135 #elif MCO2PRE == DivisionBy3
00136
00137     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00138     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00139     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00140
00141 #elif MCO2PRE == DivisionBy4
00142
00143     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00144     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00145     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00146
00147 #elif MCO2PRE == DivisionBy5
00148
00149     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00150     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00151     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00152
00153 #endif
00154
00155      // MCO1 selection:
00156 #if MCO1 == HSI
00157
00158     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00159     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00160
00161 #elif MCO1 == LSE
00162
00163     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00164     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00165
00166 #elif MCO1 == HSE
00167
00168     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00169     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00170
00171 #elif MCO1 == PLLCLK
00172
00173     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00174     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00175
00176 #endif
00177
00178      // MCO1 prescaler:
00179 #if MCO1PRE == NoDivision
00180
00181     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00182     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
```

```

00186         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00187
00188 #elif MCO1PRE == DivisionBy2
00189
00190         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00191         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00192         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00193
00194 #elif MCO1PRE == DivisionBy3
00195
00196         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00197         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00198         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00199
00200 #elif MCO1PRE == DivisionBy4
00201
00202         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00203         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00204         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00205
00206 #elif MCO1PRE == DivisionBy5
00207
00208         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00209         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00210         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00211
00212 #endif
00213
00214
00215 // AHB prescalers:
00216 #if HPRE == NoDivision
00217
00218         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00219         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00220         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00221         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00222
00223 #elif HPRE == SYSCLKby2
00224
00225         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00226         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00227         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00228         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00229
00230 #elif HPRE == SYSCLKby4
00231
00232         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00233         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00234         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00235         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00236
00237 #elif HPRE == SYSCLKby8
00238
00239         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00240         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00241         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00242         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00243
00244 #elif HPRE == SYSCLKby16
00245
00246         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00247         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00248         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00249         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00250
00251 #elif HPRE == SYSCLKby64
00252
00253         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00254         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00255         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00256         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00257
00258 #elif HPRE == SYSCLKby128
00259
00260         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00261         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00262         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00263         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00264
00265 #elif HPRE == SYSCLKby256
00266
00267         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00268         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00269         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00270         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00271
00272 #elif HPRE == SYSCLKby512

```

```

00273
00274     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00275     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00276     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00277     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00278
00279 #endif
00280
00281
00282     // APB1 prescalers:
00283     #if PPRE1 == NoDivision
00284
00285         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00286         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00287         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00288
00289     #elif PPRE1 == AHBby2
00290
00291         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00292         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00293         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00294
00295     #elif PPRE1 == AHBby4
00296
00297         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00298         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00299         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00300
00301     #elif PPRE1 == AHBby8
00302
00303         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00304         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00305         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00306
00307     #elif PPRE1 == AHBby16
00308
00309         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00310         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00311         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00312
00313 #endif
00314
00315     // APB2 prescalers:
00316     #if PPRE2 == NoDivision
00317
00318         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00319         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00320         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00321
00322     #elif PPRE2 == AHBby2
00323
00324         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00325         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00326         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00327
00328     #elif PPRE2 == AHBby4
00329
00330         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00331         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00332         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00333
00334     #elif PPRE2 == AHBby8
00335
00336         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00337         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00338         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00339
00340     #elif PPRE2 == AHBby16
00341
00342         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00343         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00344         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00345
00346 #endif
00347
00348
00349     // PLL configurations:
00350
00351
00352
00353     // Enable the selected CLK (HSI ON/HSE ON/PLL ON): //
00354     // HSE (ON/OFF).
00355     #if HSE_EN == ENABLE
00356
00357         SET_BIT( RCC->CR, RCC_CR_HSEON ) ;
00358
00359     #elif HSE_EN == DISABLE

```

```

00360
00361         CLR_BIT( RCC->CR, RCC_CR_HSEON ) ;
00362
00363 #endif
00364
00365 // HSI (ON/OFF).
00366 #if HSI_EN == ENABLE
00367
00368         SET_BIT( RCC->CR, RCC_CR_HSION ) ;
00369
00370 #elif HSI_EN == DISABLE
00371
00372         CLR_BIT( RCC->CR, RCC_CR_HSION ) ;
00373
00374 #endif
00375 #endif
00376
00377 }

```

References [CLR\\_BIT](#), [RCC](#), [RCC\\_CFGR\\_HPRE\\_b0](#), [RCC\\_CFGR\\_HPRE\\_b1](#), [RCC\\_CFGR\\_HPRE\\_b2](#), [RCC\\_CFGR\\_HPRE\\_b3](#), [RCC\\_CFGR\\_MOC1\\_b0](#), [RCC\\_CFGR\\_MOC1\\_b1](#), [RCC\\_CFGR\\_MOC1PRE\\_b0](#), [RCC\\_CFGR\\_MOC1PRE\\_b1](#), [RCC\\_CFGR\\_MOC1PRE\\_b2](#), [RCC\\_CFGR\\_MOC2\\_b0](#), [RCC\\_CFGR\\_MOC2\\_b1](#), [RCC\\_CFGR\\_MOC2PRE\\_b0](#), [RCC\\_CFGR\\_MOC2PRE\\_b1](#), [RCC\\_CFGR\\_MOC2PRE\\_b2](#), [RCC\\_CFGR\\_PPRE1\\_b0](#), [RCC\\_CFGR\\_PPRE1\\_b1](#), [RCC\\_CFGR\\_PPRE1\\_b2](#), [RCC\\_CFGR\\_PPRE2\\_b0](#), [RCC\\_CFGR\\_PPRE2\\_b1](#), [RCC\\_CFGR\\_PPRE2\\_b2](#), [RCC\\_CFGR\\_SW\\_b0](#), [RCC\\_CFGR\\_SW\\_b1](#), [RCC\\_CR\\_CSSON](#), [RCC\\_CR\\_HSEBYP](#), [RCC\\_CR\\_HSEON](#), [RCC\\_CR\\_HSION](#), [RCC\\_CR\\_PLLI2SON](#), [RCC\\_CR\\_PLLON](#), and [SET\\_BIT](#).

### 7.22.3.2 MRCC\_vEnablePeriphralCLK()

```

void MRCC_vEnablePeriphralCLK (
    u32_t A_u32BusID,
    u32_t A_u32PeriphralID )

```

Definition at line 382 of file [MRCC\\_program.c](#).

```

00383 {
00384
00385     switch( A_u32BusID )
00386     {
00387
00388         case RCC_AHB1 :
00389
00390             SET_BIT( RCC->AHB1ENR, A_u32PeriphralID ) ;
00391
00392             break ;
00393
00394         case RCC_AHB2 :
00395
00396             SET_BIT( RCC->AHB2ENR, A_u32PeriphralID ) ;
00397
00398             break ;
00399
00400         case RCC_APB1 :
00401
00402             SET_BIT( RCC->APB1ENR, A_u32PeriphralID ) ;
00403
00404             break ;
00405
00406         case RCC_APB2 :
00407
00408             SET_BIT( RCC->APB2ENR, A_u32PeriphralID ) ;
00409
00410             break ;
00411
00412         case RCC_AHB1LPENR:
00413
00414             SET_BIT( RCC->AHB1LPENR, A_u32PeriphralID ) ;
00415
00416             break ;
00417
00418         default:
00419
00420             break ;
00421
00422     }

```

```

00423
00424         // Error wrong Bus ID
00425
00426         break ;
00427
00428     }
00429
00430 }
```

References [RCC](#), [RCC\\_AHB1](#), [RCC\\_AHB1LPENR](#), [RCC\\_AHB2](#), [RCC\\_APB1](#), [RCC\\_APB2](#), and [SET\\_BIT](#).

### 7.22.3.3 MRCC\_vDisablePeripheralCLK()

```

void MRCC_vDisablePeripheralCLK (
    u32_t A_u32BusID,
    u32_t A_u32PeripheralID )
```

Definition at line 435 of file [MRCC\\_program.c](#).

```

00436 {
00437
00438     switch( A_u32BusID )
00439     {
00440
00441         case RCC_AHB1 :
00442
00443             CLR_BIT( RCC->AHB1ENR, A_u32PeripheralID ) ;
00444
00445             break ;
00446
00447         case RCC_AHB2 :
00448
00449             CLR_BIT( RCC->AHB2ENR, A_u32PeripheralID ) ;
00450
00451             break ;
00452
00453             break ;
00454
00455         case RCC_APB1 :
00456
00457             CLR_BIT( RCC->APB1ENR, A_u32PeripheralID ) ;
00458
00459             break ;
00460
00461         case RCC_APB2 :
00462
00463             CLR_BIT( RCC->APB2ENR, A_u32PeripheralID ) ;
00464
00465             break ;
00466
00467         case RCC_AHB1LPENR:
00468
00469             CLR_BIT( RCC->AHB1LPENR, A_u32PeripheralID ) ;
00470
00471             break ;
00472
00473             break ;
00474
00475         default:
00476
00477             // Error wrong Bus ID
00478
00479             break ;
00480
00481     }
00482
00483 }
```

References [CLR\\_BIT](#), [RCC](#), [RCC\\_AHB1](#), [RCC\\_AHB1LPENR](#), [RCC\\_AHB2](#), [RCC\\_APB1](#), and [RCC\\_APB2](#).

## 7.23 MRCC\_interface.h

[Go to the documentation of this file.](#)

```

00001
00010 #ifndef MCAL_RCC_MRCC_INTERFACE_H_
00011 #define MCAL_RCC_MRCC_INTERFACE_H_
00012
00013
00014
00015 /*****
00016 /*                      Functions prototypes                      */
00017 *****/
00018
00019 void MRCC_vInit( void ) ;
00020
00021 void MRCC_vEnablePeripheralCLK( u32_t A_u32BusID, u32_t A_u32PeripheralID ) ;
00022
00023 void MRCC_vDisablePeripheralCLK( u32_t A_u32BusID, u32_t A_u32PeripheralID ) ;
00024
00025
00026 /*****
00027 /*                      Interfacing macros                      */
00028 *****/
00029
00030 // A_u32BusID options:
00031 #define RCC_AHB1          1
00032 #define RCC_AHB2          2
00033 #define RCC_APB1          3
00034 #define RCC_APB2          4
00035 #define RCC_AHB1LPENR    5
00036
00037 // A_u32PeripheralID options:
00038 #define AHB1ENR_DMA2EN    22
00039 #define AHB1ENR_DMA1EN    21
00040 #define AHB1ENR_CRCEN     12
00041 #define AHB1ENR_GPIOHEN   7
00042 #define AHB1ENR_GPIOEEN   4
00043 #define AHB1ENR_GPIODEN   3
00044 #define AHB1ENR_GPIOCEN   2
00045 #define AHB1ENR_GPIOBEN   1
00046 #define AHB1ENR_GPIOAEN   0
00047
00048 #define AHB2ENR_OTGFSEN    7
00049
00050 #define APB1ENR_PWREN      28
00051 #define APB1ENR_I2C3EN     23
00052 #define APB1ENR_I2C2EN     22
00053 #define APB1ENR_I2C1EN     21
00054 #define APB1ENR_USART2EN   17
00055 #define APB1ENR_SPI3EN     15
00056 #define APB1ENR_SPI2EN     14
00057 #define APB1ENR_WWDGEN     11
00058 #define APB1ENR_TIM5EN     3
00059 #define APB1ENR_TIM4EN     2
00060 #define APB1ENR_TIM3EN     1
00061 #define APB1ENR_TIM2EN     0
00062
00063 #define APB2ENR_TIM11EN    18
00064 #define APB2ENR_TIM10EN    17
00065 #define APB2ENR_TIM9EN     16
00066 #define APB2ENR_SYSCFGEN   14
00067 #define APB2ENR_SPI4EN     13
00068 #define APB2ENR_SPI1EN     12
00069 #define APB2ENR_SDI0EN     11
00070 #define APB2ENR_ADC1EN     8
00071 #define APB2ENR_USART6EN   5
00072 #define APB2ENR_USART1EN   4
00073 #define APB2ENR_TIM1EN     0
00074
00075
00076
00077 #define AHB1LPENR_FLITFLEN 15
00078
00079
00080
00081
00082 #endif /* MCAL_RCC_MRCC_INTERFACE_H_ */

```

## 7.24 src/COTS/MCAL/RCC/MRCC\_private.h File Reference

This file contains the registers information and addresses for the RCC module.

## Data Structures

- struct [RCC\\_MemoryMapType](#)

## Macros

- #define [RCC\\_BASE\\_ADDRESS](#) 0x40023800
- #define [RCC](#) ( (volatile [P2VAR\(RCC\\_MemoryMapType\)](#) ) ([RCC\\_BASE\\_ADDRESS](#)) )
- #define [RCC\\_CR\\_PLLI2SRDY](#) 27
- #define [RCC\\_CR\\_PLLI2SON](#) 26
- #define [RCC\\_CR\\_PLLRDY](#) 25
- #define [RCC\\_CR\\_PLLON](#) 24
- #define [RCC\\_CR\\_CSSON](#) 19
- #define [RCC\\_CR\\_HSEBYP](#) 18
- #define [RCC\\_CR\\_HSERDY](#) 17
- #define [RCC\\_CR\\_HSEON](#) 16
- #define [RCC\\_CR\\_HSIRDY](#) 1
- #define [RCC\\_CR\\_HSION](#) 0
- #define [RCC\\_PLLCFGR\\_PLLQ\\_b0](#) 24
- #define [RCC\\_PLLCFGR\\_PLLQ\\_b1](#) 25
- #define [RCC\\_PLLCFGR\\_PLLQ\\_b2](#) 26
- #define [RCC\\_PLLCFGR\\_PLLQ\\_b3](#) 27
- #define [RCC\\_PLLCFGR\\_PLLSRC](#) 22
- #define [RCC\\_PLLCFGR\\_PLLP\\_b0](#) 16
- #define [RCC\\_PLLCFGR\\_PLLP\\_b1](#) 17
- #define [RCC\\_PLLCFGR\\_PLLN\\_b0](#) 6
- #define [RCC\\_PLLCFGR\\_PLLN\\_b1](#) 7
- #define [RCC\\_PLLCFGR\\_PLLN\\_b2](#) 8
- #define [RCC\\_PLLCFGR\\_PLLN\\_b3](#) 9
- #define [RCC\\_PLLCFGR\\_PLLN\\_b4](#) 10
- #define [RCC\\_PLLCFGR\\_PLLN\\_b5](#) 11
- #define [RCC\\_PLLCFGR\\_PLLN\\_b6](#) 12
- #define [RCC\\_PLLCFGR\\_PLLN\\_b7](#) 13
- #define [RCC\\_PLLCFGR\\_PLLN\\_b8](#) 14
- #define [RCC\\_PLLCFGR\\_PLLM\\_b0](#) 0
- #define [RCC\\_PLLCFGR\\_PLLM\\_b1](#) 1
- #define [RCC\\_PLLCFGR\\_PLLM\\_b2](#) 2
- #define [RCC\\_PLLCFGR\\_PLLM\\_b3](#) 3
- #define [RCC\\_PLLCFGR\\_PLLM\\_b4](#) 4
- #define [RCC\\_PLLCFGR\\_PLLM\\_b5](#) 5
- #define [RCC\\_CFGR\\_MOC2\\_b0](#) 30
- #define [RCC\\_CFGR\\_MOC2\\_b1](#) 31
- #define [RCC\\_CFGR\\_MOC2PRE\\_b0](#) 27
- #define [RCC\\_CFGR\\_MOC2PRE\\_b1](#) 28
- #define [RCC\\_CFGR\\_MOC2PRE\\_b2](#) 29
- #define [RCC\\_CFGR\\_MOC1PRE\\_b0](#) 24
- #define [RCC\\_CFGR\\_MOC1PRE\\_b1](#) 25
- #define [RCC\\_CFGR\\_MOC1PRE\\_b2](#) 26
- #define [RCC\\_CFGR\\_I2SSRC](#) 23
- #define [RCC\\_CFGR\\_MOC1\\_b0](#) 21
- #define [RCC\\_CFGR\\_MOC1\\_b1](#) 22
- #define [RCC\\_CFGR\\_RTCPRE\\_b0](#) 16
- #define [RCC\\_CFGR\\_RTCPRE\\_b1](#) 17
- #define [RCC\\_CFGR\\_RTCPRE\\_b2](#) 18

- #define [RCC\\_CFGR\\_RTCPRE\\_b3](#) 19
- #define [RCC\\_CFGR\\_RTCPRE\\_b4](#) 20
- #define [RCC\\_CFGR\\_PPRE2\\_b0](#) 13
- #define [RCC\\_CFGR\\_PPRE2\\_b1](#) 14
- #define [RCC\\_CFGR\\_PPRE2\\_b2](#) 15
- #define [RCC\\_CFGR\\_PPRE1\\_b0](#) 10
- #define [RCC\\_CFGR\\_PPRE1\\_b1](#) 11
- #define [RCC\\_CFGR\\_PPRE1\\_b2](#) 12
- #define [RCC\\_CFGR\\_HPRE\\_b0](#) 4
- #define [RCC\\_CFGR\\_HPRE\\_b1](#) 5
- #define [RCC\\_CFGR\\_HPRE\\_b2](#) 6
- #define [RCC\\_CFGR\\_HPRE\\_b3](#) 7
- #define [RCC\\_CFGR\\_SWS\\_b0](#) 2
- #define [RCC\\_CFGR\\_SWS\\_b1](#) 3
- #define [RCC\\_CFGR\\_SW\\_b0](#) 0
- #define [RCC\\_CFGR\\_SW\\_b1](#) 1
- #define [RCC\\_AHB1ENR\\_DMA2EN](#) 22
- #define [RCC\\_AHB1ENR\\_DMA1EN](#) 21
- #define [RCC\\_AHB1ENR\\_CRCEN](#) 12
- #define [RCC\\_AHB1ENR\\_GPIOHEN](#) 7
- #define [RCC\\_AHB1ENR\\_GPIOEEN](#) 4
- #define [RCC\\_AHB1ENR\\_GPIODEN](#) 3
- #define [RCC\\_AHB1ENR\\_GPIOCEN](#) 2
- #define [RCC\\_AHB1ENR\\_GPIOBEN](#) 1
- #define [RCC\\_AHB1ENR\\_GPIOAEN](#) 0
- #define [RCC\\_AHB2ENR\\_OTGFSEN](#) 7
- #define [RCC\\_APB1ENR\\_PWREN](#) 28
- #define [RCC\\_APB1ENR\\_I2C3EN](#) 23
- #define [RCC\\_APB1ENR\\_I2C2EN](#) 22
- #define [RCC\\_APB1ENR\\_I2C1EN](#) 21
- #define [RCC\\_APB1ENR\\_USART2EN](#) 17
- #define [RCC\\_APB1ENR\\_SPI3EN](#) 15
- #define [RCC\\_APB1ENR\\_SPI2EN](#) 14
- #define [RCC\\_APB1ENR\\_WWDGEN](#) 11
- #define [RCC\\_APB1ENR\\_TIM5EN](#) 3
- #define [RCC\\_APB1ENR\\_TIM4EN](#) 2
- #define [RCC\\_APB1ENR\\_TIM3EN](#) 1
- #define [RCC\\_APB1ENR\\_TIM2EN](#) 0
- #define [RCC\\_APB2ENR\\_TIM11EN](#) 18
- #define [RCC\\_APB2ENR\\_TIM10EN](#) 17
- #define [RCC\\_APB2ENR\\_TIM9EN](#) 16
- #define [RCC\\_APB2ENR\\_SYSCFGEN](#) 14
- #define [RCC\\_APB2ENR\\_SPI4EN](#) 13
- #define [RCC\\_APB2ENR\\_SPI1EN](#) 12
- #define [RCC\\_APB2ENR\\_SDIOEN](#) 11
- #define [RCC\\_APB2ENR\\_ADC1EN](#) 8
- #define [RCC\\_APB2ENR\\_USART6EN](#) 5
- #define [RCC\\_APB2ENR\\_USART1EN](#) 4
- #define [RCC\\_APB2ENR\\_TIM1EN](#) 0
- #define [RCC\\_AHB1LPENR\\_FLITFLPEN](#) 15
- #define [ENABLE](#) 1
- #define [DISABLE](#) 2
- #define [BYBASED](#) 1
- #define [NOTBYBASED](#) 2
- #define [SYSCLK](#) 1



- #define [PLLI2SCLK](#) 5
- #define [HSE](#) 3
- #define [PLLCLK](#) 4
- #define [NoDivision](#) 1
- #define [DivisionBy2](#) 2
- #define [DivisionBy3](#) 3
- #define [DivisionBy4](#) 4
- #define [DivisionBy5](#) 5
- #define [I2S\\_CKIN](#) 2
- #define [HSI](#) 1
- #define [LSE](#) 2
- #define [PLLCLK](#) 4
- #define [SYSCLKby2](#) 2
- #define [SYSCLKby4](#) 3
- #define [SYSCLKby8](#) 4
- #define [SYSCLKby16](#) 5
- #define [SYSCLKby64](#) 6
- #define [SYSCLKby128](#) 7
- #define [SYSCLKby256](#) 8
- #define [SYSCLKby512](#) 9
- #define [NoCLK0](#) 1
- #define [NoCLK1](#) 2
- #define [HSEby2](#) 3
- #define [HSEby3](#) 4
- #define [HSEby4](#) 5
- #define [HSEby5](#) 6
- #define [HSEby6](#) 7
- #define [HSEby7](#) 8
- #define [HSEby8](#) 9
- #define [HSEby9](#) 10
- #define [HSEby10](#) 11
- #define [HSEby11](#) 12
- #define [HSEby12](#) 13
- #define [HSEby13](#) 14
- #define [HSEby14](#) 15
- #define [HSEby15](#) 16
- #define [HSEby16](#) 17
- #define [HSEby17](#) 18
- #define [HSEby18](#) 19
- #define [HSEby19](#) 20
- #define [HSEby20](#) 21
- #define [HSEby21](#) 22
- #define [HSEby22](#) 23
- #define [HSEby23](#) 24
- #define [HSEby24](#) 25
- #define [HSEby25](#) 26
- #define [HSEby26](#) 27
- #define [HSEby27](#) 28
- #define [HSEby28](#) 39
- #define [HSEby29](#) 30
- #define [HSEby30](#) 31
- #define [HSEby31](#) 32
- #define [AHBby2](#) 1
- #define [AHBby4](#) 2
- #define [AHBby8](#) 3

- #define [AHBby16](#) 4
- #define [Equal\\_0](#) 0
- #define [Equal\\_1](#) 1
- #define [Equal\\_2](#) 2
- #define [Equal\\_3](#) 3
- #define [Equal\\_4](#) 4
- #define [Equal\\_5](#) 5
- #define [Equal\\_6](#) 6
- #define [Equal\\_7](#) 7
- #define [Equal\\_8](#) 8
- #define [Equal\\_9](#) 9
- #define [Equal\\_10](#) 10
- #define [Equal\\_11](#) 11
- #define [Equal\\_12](#) 12
- #define [Equal\\_13](#) 13
- #define [Equal\\_14](#) 14
- #define [Equal\\_15](#) 15
- #define [Equal\\_16](#) 16
- #define [Equal\\_17](#) 17
- #define [Equal\\_18](#) 18
- #define [Equal\\_19](#) 19
- #define [Equal\\_20](#) 20
- #define [Equal\\_21](#) 21
- #define [Equal\\_22](#) 22
- #define [Equal\\_23](#) 23
- #define [Equal\\_24](#) 24
- #define [Equal\\_25](#) 25
- #define [Equal\\_26](#) 26
- #define [Equal\\_27](#) 27
- #define [Equal\\_28](#) 28
- #define [Equal\\_29](#) 29
- #define [Equal\\_30](#) 30
- #define [Equal\\_31](#) 31
- #define [Equal\\_32](#) 32
- #define [Equal\\_33](#) 33
- #define [Equal\\_34](#) 34
- #define [Equal\\_35](#) 35
- #define [Equal\\_36](#) 36
- #define [Equal\\_37](#) 37
- #define [Equal\\_38](#) 38
- #define [Equal\\_39](#) 39
- #define [Equal\\_40](#) 40
- #define [Equal\\_41](#) 41
- #define [Equal\\_42](#) 42
- #define [Equal\\_43](#) 43
- #define [Equal\\_44](#) 44
- #define [Equal\\_45](#) 45
- #define [Equal\\_46](#) 46
- #define [Equal\\_47](#) 47
- #define [Equal\\_48](#) 48
- #define [Equal\\_49](#) 49
- #define [Equal\\_50](#) 50
- #define [Equal\\_51](#) 51
- #define [Equal\\_52](#) 52
- #define [Equal\\_53](#) 53

- `#define Equal_54` 54
- `#define Equal_55` 55
- `#define Equal_56` 56
- `#define Equal_57` 57
- `#define Equal_58` 58
- `#define Equal_59` 59
- `#define Equal_60` 60
- `#define Equal_61` 61
- `#define Equal_62` 62
- `#define Equal_63` 63

### 7.24.1 Detailed Description

This file contains the registers information and addresses for the RCC module.

#### Author

Ali El Bana & Mo Alaa

#### Version

2.0

#### Date

11/9/2022

Definition in file [MRCC\\_private.h](#).

### 7.24.2 Macro Definition Documentation

#### 7.24.2.1 RCC\_BASE\_ADDRESS

```
#define RCC_BASE_ADDRESS 0x40023800
```

Definition at line 76 of file [MRCC\\_private.h](#).

#### 7.24.2.2 RCC

```
#define RCC ( (volatile P2VAR(RCC_MemoryMapType) ) (RCC_BASE_ADDRESS) )
```

Definition at line 78 of file [MRCC\\_private.h](#).

#### 7.24.2.3 RCC\_CR\_PLLI2SRDY

```
#define RCC_CR_PLLI2SRDY 27
```

Definition at line 85 of file [MRCC\\_private.h](#).

#### 7.24.2.4 RCC\_CR\_PLLI2SON

```
#define RCC_CR_PLLI2SON 26
```

Definition at line 86 of file [MRCC\\_private.h](#).

#### 7.24.2.5 RCC\_CR\_PLLRDY

```
#define RCC_CR_PLLRDY 25
```

Definition at line 87 of file [MRCC\\_private.h](#).

#### 7.24.2.6 RCC\_CR\_PLLON

```
#define RCC_CR_PLLON 24
```

Definition at line 88 of file [MRCC\\_private.h](#).

#### 7.24.2.7 RCC\_CR\_CSSON

```
#define RCC_CR_CSSON 19
```

Definition at line 89 of file [MRCC\\_private.h](#).

#### 7.24.2.8 RCC\_CR\_HSEBYP

```
#define RCC_CR_HSEBYP 18
```

Definition at line 90 of file [MRCC\\_private.h](#).

#### 7.24.2.9 RCC\_CR\_HSERDY

```
#define RCC_CR_HSERDY 17
```

Definition at line 91 of file [MRCC\\_private.h](#).

#### 7.24.2.10 RCC\_CR\_HSEON

```
#define RCC_CR_HSEON 16
```

Definition at line 92 of file [MRCC\\_private.h](#).

#### 7.24.2.11 RCC\_CR\_HSIRDY

```
#define RCC_CR_HSIRDY 1
```

Definition at line 93 of file [MRCC\\_private.h](#).

#### 7.24.2.12 RCC\_CR\_HSION

```
#define RCC_CR_HSION 0
```

Definition at line 94 of file [MRCC\\_private.h](#).

#### 7.24.2.13 RCC\_PLLCFGR\_PLLQ\_b0

```
#define RCC_PLLCFGR_PLLQ_b0 24
```

Definition at line 97 of file [MRCC\\_private.h](#).

#### 7.24.2.14 RCC\_PLLCFGR\_PLLQ\_b1

```
#define RCC_PLLCFGR_PLLQ_b1 25
```

Definition at line 98 of file [MRCC\\_private.h](#).

#### 7.24.2.15 RCC\_PLLCFGR\_PLLQ\_b2

```
#define RCC_PLLCFGR_PLLQ_b2 26
```

Definition at line 99 of file [MRCC\\_private.h](#).

#### 7.24.2.16 RCC\_PLLCFGR\_PLLQ\_b3

```
#define RCC_PLLCFGR_PLLQ_b3 27
```

Definition at line 100 of file [MRCC\\_private.h](#).

#### 7.24.2.17 RCC\_PLLCFGR\_PLLSRC

```
#define RCC_PLLCFGR_PLLSRC 22
```

Definition at line 101 of file [MRCC\\_private.h](#).

#### 7.24.2.18 RCC\_PLLCFGR\_PLLP\_b0

```
#define RCC_PLLCFGR_PLLP_b0 16
```

Definition at line 102 of file [MRCC\\_private.h](#).

#### 7.24.2.19 RCC\_PLLCFGR\_PLLP\_b1

```
#define RCC_PLLCFGR_PLLP_b1 17
```

Definition at line 103 of file [MRCC\\_private.h](#).

#### 7.24.2.20 RCC\_PLLCFGR\_PLLN\_b0

```
#define RCC_PLLCFGR_PLLN_b0 6
```

Definition at line 104 of file [MRCC\\_private.h](#).

#### 7.24.2.21 RCC\_PLLCFGR\_PLLN\_b1

```
#define RCC_PLLCFGR_PLLN_b1 7
```

Definition at line 105 of file [MRCC\\_private.h](#).

#### 7.24.2.22 RCC\_PLLCFGR\_PLLN\_b2

```
#define RCC_PLLCFGR_PLLN_b2 8
```

Definition at line 106 of file [MRCC\\_private.h](#).

#### 7.24.2.23 RCC\_PLLCFGR\_PLLN\_b3

```
#define RCC_PLLCFGR_PLLN_b3 9
```

Definition at line 107 of file [MRCC\\_private.h](#).

#### 7.24.2.24 RCC\_PLLCFGR\_PLLN\_b4

```
#define RCC_PLLCFGR_PLLN_b4 10
```

Definition at line 108 of file [MRCC\\_private.h](#).

#### 7.24.2.25 RCC\_PLLCFGR\_PLLN\_b5

```
#define RCC_PLLCFGR_PLLN_b5 11
```

Definition at line 109 of file [MRCC\\_private.h](#).

#### 7.24.2.26 RCC\_PLLCFGR\_PLLN\_b6

```
#define RCC_PLLCFGR_PLLN_b6 12
```

Definition at line 110 of file [MRCC\\_private.h](#).

#### 7.24.2.27 RCC\_PLLCFGR\_PLLN\_b7

```
#define RCC_PLLCFGR_PLLN_b7 13
```

Definition at line 111 of file [MRCC\\_private.h](#).

#### 7.24.2.28 RCC\_PLLCFGR\_PLLN\_b8

```
#define RCC_PLLCFGR_PLLN_b8 14
```

Definition at line 112 of file [MRCC\\_private.h](#).

#### 7.24.2.29 RCC\_PLLCFGR\_PLLM\_b0

```
#define RCC_PLLCFGR_PLLM_b0 0
```

Definition at line 113 of file [MRCC\\_private.h](#).

#### 7.24.2.30 RCC\_PLLCFGR\_PLLM\_b1

```
#define RCC_PLLCFGR_PLLM_b1 1
```

Definition at line 114 of file [MRCC\\_private.h](#).

#### 7.24.2.31 RCC\_PLLCFGR\_PLLM\_b2

```
#define RCC_PLLCFGR_PLLM_b2 2
```

Definition at line 115 of file [MRCC\\_private.h](#).

#### 7.24.2.32 RCC\_PLLCFGR\_PLLM\_b3

```
#define RCC_PLLCFGR_PLLM_b3 3
```

Definition at line 116 of file [MRCC\\_private.h](#).



#### 7.24.2.33 RCC\_PLLCFGR\_PLLM\_b4

```
#define RCC_PLLCFGR_PLLM_b4 4
```

Definition at line 117 of file [MRCC\\_private.h](#).

#### 7.24.2.34 RCC\_PLLCFGR\_PLLM\_b5

```
#define RCC_PLLCFGR_PLLM_b5 5
```

Definition at line 118 of file [MRCC\\_private.h](#).

#### 7.24.2.35 RCC\_CFGR\_MOC2\_b0

```
#define RCC_CFGR_MOC2_b0 30
```

Definition at line 121 of file [MRCC\\_private.h](#).

#### 7.24.2.36 RCC\_CFGR\_MOC2\_b1

```
#define RCC_CFGR_MOC2_b1 31
```

Definition at line 122 of file [MRCC\\_private.h](#).

#### 7.24.2.37 RCC\_CFGR\_MOC2PRE\_b0

```
#define RCC_CFGR_MOC2PRE_b0 27
```

Definition at line 123 of file [MRCC\\_private.h](#).

#### 7.24.2.38 RCC\_CFGR\_MOC2PRE\_b1

```
#define RCC_CFGR_MOC2PRE_b1 28
```

Definition at line 124 of file [MRCC\\_private.h](#).

#### 7.24.2.39 RCC\_CFGR\_MOC2PRE\_b2

```
#define RCC_CFGR_MOC2PRE_b2 29
```

Definition at line 125 of file [MRCC\\_private.h](#).

#### 7.24.2.40 RCC\_CFGR\_MOC1PRE\_b0

```
#define RCC_CFGR_MOC1PRE_b0 24
```

Definition at line 126 of file [MRCC\\_private.h](#).

#### 7.24.2.41 RCC\_CFGR\_MOC1PRE\_b1

```
#define RCC_CFGR_MOC1PRE_b1 25
```

Definition at line 127 of file [MRCC\\_private.h](#).

#### 7.24.2.42 RCC\_CFGR\_MOC1PRE\_b2

```
#define RCC_CFGR_MOC1PRE_b2 26
```

Definition at line 128 of file [MRCC\\_private.h](#).

#### 7.24.2.43 RCC\_CFGR\_I2SSRC

```
#define RCC_CFGR_I2SSRC 23
```

Definition at line 129 of file [MRCC\\_private.h](#).

#### 7.24.2.44 RCC\_CFGR\_MOC1\_b0

```
#define RCC_CFGR_MOC1_b0 21
```

Definition at line 130 of file [MRCC\\_private.h](#).

#### 7.24.2.45 RCC\_CFGR\_MOC1\_b1

```
#define RCC_CFGR_MOC1_b1 22
```

Definition at line 131 of file [MRCC\\_private.h](#).

#### 7.24.2.46 RCC\_CFGR\_RTCPRE\_b0

```
#define RCC_CFGR_RTCPRE_b0 16
```

Definition at line 132 of file [MRCC\\_private.h](#).

#### 7.24.2.47 RCC\_CFGR\_RTCPRE\_b1

```
#define RCC_CFGR_RTCPRE_b1 17
```

Definition at line 133 of file [MRCC\\_private.h](#).

#### 7.24.2.48 RCC\_CFGR\_RTCPRE\_b2

```
#define RCC_CFGR_RTCPRE_b2 18
```

Definition at line 134 of file [MRCC\\_private.h](#).

#### 7.24.2.49 RCC\_CFGR\_RTCPRE\_b3

```
#define RCC_CFGR_RTCPRE_b3 19
```

Definition at line 135 of file [MRCC\\_private.h](#).

#### 7.24.2.50 RCC\_CFGR\_RTCPRE\_b4

```
#define RCC_CFGR_RTCPRE_b4 20
```

Definition at line 136 of file [MRCC\\_private.h](#).

#### 7.24.2.51 RCC\_CFGR\_PPRE2\_b0

```
#define RCC_CFGR_PPRE2_b0 13
```

Definition at line 137 of file [MRCC\\_private.h](#).

#### 7.24.2.52 RCC\_CFGR\_PPRE2\_b1

```
#define RCC_CFGR_PPRE2_b1 14
```

Definition at line 138 of file [MRCC\\_private.h](#).

#### 7.24.2.53 RCC\_CFGR\_PPRE2\_b2

```
#define RCC_CFGR_PPRE2_b2 15
```

Definition at line 139 of file [MRCC\\_private.h](#).

#### 7.24.2.54 RCC\_CFGR\_PPRE1\_b0

```
#define RCC_CFGR_PPRE1_b0 10
```

Definition at line 140 of file [MRCC\\_private.h](#).

#### 7.24.2.55 RCC\_CFGR\_PPRE1\_b1

```
#define RCC_CFGR_PPRE1_b1 11
```

Definition at line 141 of file [MRCC\\_private.h](#).

#### 7.24.2.56 RCC\_CFGR\_PPRE1\_b2

```
#define RCC_CFGR_PPRE1_b2 12
```

Definition at line 142 of file [MRCC\\_private.h](#).

#### 7.24.2.57 RCC\_CFGR\_HPRE\_b0

```
#define RCC_CFGR_HPRE_b0 4
```

Definition at line 143 of file [MRCC\\_private.h](#).

#### 7.24.2.58 RCC\_CFGR\_HPRE\_b1

```
#define RCC_CFGR_HPRE_b1 5
```

Definition at line 144 of file [MRCC\\_private.h](#).

#### 7.24.2.59 RCC\_CFGR\_HPRE\_b2

```
#define RCC_CFGR_HPRE_b2 6
```

Definition at line 145 of file [MRCC\\_private.h](#).

#### 7.24.2.60 RCC\_CFGR\_HPRE\_b3

```
#define RCC_CFGR_HPRE_b3 7
```

Definition at line 146 of file [MRCC\\_private.h](#).

#### 7.24.2.61 RCC\_CFGR\_SWS\_b0

```
#define RCC_CFGR_SWS_b0 2
```

Definition at line 147 of file [MRCC\\_private.h](#).

#### 7.24.2.62 RCC\_CFGR\_SWS\_b1

```
#define RCC_CFGR_SWS_b1 3
```

Definition at line 148 of file [MRCC\\_private.h](#).

#### 7.24.2.63 RCC\_CFGR\_SW\_b0

```
#define RCC_CFGR_SW_b0 0
```

Definition at line 149 of file [MRCC\\_private.h](#).

#### 7.24.2.64 RCC\_CFGR\_SW\_b1

```
#define RCC_CFGR_SW_b1 1
```

Definition at line 150 of file [MRCC\\_private.h](#).

#### 7.24.2.65 RCC\_AHB1ENR\_DMA2EN

```
#define RCC_AHB1ENR_DMA2EN 22
```

Definition at line 153 of file [MRCC\\_private.h](#).

#### 7.24.2.66 RCC\_AHB1ENR\_DMA1EN

```
#define RCC_AHB1ENR_DMA1EN 21
```

Definition at line 154 of file [MRCC\\_private.h](#).

#### 7.24.2.67 RCC\_AHB1ENR\_CRCEN

```
#define RCC_AHB1ENR_CRCEN 12
```

Definition at line 155 of file [MRCC\\_private.h](#).

#### 7.24.2.68 RCC\_AHB1ENR\_GPIOHEN

```
#define RCC_AHB1ENR_GPIOHEN 7
```

Definition at line 156 of file [MRCC\\_private.h](#).

**7.24.2.69 RCC\_AHB1ENR\_GPIOEEN**

```
#define RCC_AHB1ENR_GPIOEEN 4
```

Definition at line 157 of file [MRCC\\_private.h](#).

**7.24.2.70 RCC\_AHB1ENR\_GPIODEN**

```
#define RCC_AHB1ENR_GPIODEN 3
```

Definition at line 158 of file [MRCC\\_private.h](#).

**7.24.2.71 RCC\_AHB1ENR\_GPIOCEN**

```
#define RCC_AHB1ENR_GPIOCEN 2
```

Definition at line 159 of file [MRCC\\_private.h](#).

**7.24.2.72 RCC\_AHB1ENR\_GPIOBEN**

```
#define RCC_AHB1ENR_GPIOBEN 1
```

Definition at line 160 of file [MRCC\\_private.h](#).

**7.24.2.73 RCC\_AHB1ENR\_GPIOAEN**

```
#define RCC_AHB1ENR_GPIOAEN 0
```

Definition at line 161 of file [MRCC\\_private.h](#).

**7.24.2.74 RCC\_AHB2ENR\_OTGFSEN**

```
#define RCC_AHB2ENR_OTGFSEN 7
```

Definition at line 164 of file [MRCC\\_private.h](#).

#### 7.24.2.75 RCC\_APB1ENR\_PWREN

```
#define RCC_APB1ENR_PWREN 28
```

Definition at line 167 of file [MRCC\\_private.h](#).

#### 7.24.2.76 RCC\_APB1ENR\_I2C3EN

```
#define RCC_APB1ENR_I2C3EN 23
```

Definition at line 168 of file [MRCC\\_private.h](#).

#### 7.24.2.77 RCC\_APB1ENR\_I2C2EN

```
#define RCC_APB1ENR_I2C2EN 22
```

Definition at line 169 of file [MRCC\\_private.h](#).

#### 7.24.2.78 RCC\_APB1ENR\_I2C1EN

```
#define RCC_APB1ENR_I2C1EN 21
```

Definition at line 170 of file [MRCC\\_private.h](#).

#### 7.24.2.79 RCC\_APB1ENR\_USART2EN

```
#define RCC_APB1ENR_USART2EN 17
```

Definition at line 171 of file [MRCC\\_private.h](#).

#### 7.24.2.80 RCC\_APB1ENR\_SPI3EN

```
#define RCC_APB1ENR_SPI3EN 15
```

Definition at line 172 of file [MRCC\\_private.h](#).



**7.24.2.81 RCC\_APB1ENR\_SPI2EN**

```
#define RCC_APB1ENR_SPI2EN 14
```

Definition at line 173 of file [MRCC\\_private.h](#).

**7.24.2.82 RCC\_APB1ENR\_WWDGEN**

```
#define RCC_APB1ENR_WWDGEN 11
```

Definition at line 174 of file [MRCC\\_private.h](#).

**7.24.2.83 RCC\_APB1ENR\_TIM5EN**

```
#define RCC_APB1ENR_TIM5EN 3
```

Definition at line 175 of file [MRCC\\_private.h](#).

**7.24.2.84 RCC\_APB1ENR\_TIM4EN**

```
#define RCC_APB1ENR_TIM4EN 2
```

Definition at line 176 of file [MRCC\\_private.h](#).

**7.24.2.85 RCC\_APB1ENR\_TIM3EN**

```
#define RCC_APB1ENR_TIM3EN 1
```

Definition at line 177 of file [MRCC\\_private.h](#).

**7.24.2.86 RCC\_APB1ENR\_TIM2EN**

```
#define RCC_APB1ENR_TIM2EN 0
```

Definition at line 178 of file [MRCC\\_private.h](#).

#### 7.24.2.87 RCC\_APB2ENR\_TIM11EN

```
#define RCC_APB2ENR_TIM11EN 18
```

Definition at line 181 of file [MRCC\\_private.h](#).

#### 7.24.2.88 RCC\_APB2ENR\_TIM10EN

```
#define RCC_APB2ENR_TIM10EN 17
```

Definition at line 182 of file [MRCC\\_private.h](#).

#### 7.24.2.89 RCC\_APB2ENR\_TIM9EN

```
#define RCC_APB2ENR_TIM9EN 16
```

Definition at line 183 of file [MRCC\\_private.h](#).

#### 7.24.2.90 RCC\_APB2ENR\_SYSCFGEN

```
#define RCC_APB2ENR_SYSCFGEN 14
```

Definition at line 184 of file [MRCC\\_private.h](#).

#### 7.24.2.91 RCC\_APB2ENR\_SPI4EN

```
#define RCC_APB2ENR_SPI4EN 13
```

Definition at line 185 of file [MRCC\\_private.h](#).

#### 7.24.2.92 RCC\_APB2ENR\_SPI1EN

```
#define RCC_APB2ENR_SPI1EN 12
```

Definition at line 186 of file [MRCC\\_private.h](#).

**7.24.2.93 RCC\_APB2ENR\_SDIOEN**

```
#define RCC_APB2ENR_SDIOEN 11
```

Definition at line 187 of file [MRCC\\_private.h](#).

**7.24.2.94 RCC\_APB2ENR\_ADC1EN**

```
#define RCC_APB2ENR_ADC1EN 8
```

Definition at line 188 of file [MRCC\\_private.h](#).

**7.24.2.95 RCC\_APB2ENR\_USART6EN**

```
#define RCC_APB2ENR_USART6EN 5
```

Definition at line 189 of file [MRCC\\_private.h](#).

**7.24.2.96 RCC\_APB2ENR\_USART1EN**

```
#define RCC_APB2ENR_USART1EN 4
```

Definition at line 190 of file [MRCC\\_private.h](#).

**7.24.2.97 RCC\_APB2ENR\_TIM1EN**

```
#define RCC_APB2ENR_TIM1EN 0
```

Definition at line 191 of file [MRCC\\_private.h](#).

**7.24.2.98 RCC\_AHB1LPENR\_FLITFLPEN**

```
#define RCC_AHB1LPENR_FLITFLPEN 15
```

Definition at line 193 of file [MRCC\\_private.h](#).

**7.24.2.99 ENABLE**

```
#define ENABLE 1
```

Definition at line 200 of file [MRCC\\_private.h](#).

**7.24.2.100 DISABLE**

```
#define DISABLE 2
```

Definition at line 201 of file [MRCC\\_private.h](#).

**7.24.2.101 BYBASED**

```
#define BYBASED 1
```

Definition at line 203 of file [MRCC\\_private.h](#).

**7.24.2.102 NOTBYBASED**

```
#define NOTBYBASED 2
```

Definition at line 204 of file [MRCC\\_private.h](#).

**7.24.2.103 SYSCLK**

```
#define SYSCLK 1
```

Definition at line 206 of file [MRCC\\_private.h](#).

**7.24.2.104 PLLI2SCLK**

```
#define PLLI2SCLK 5
```

Definition at line 207 of file [MRCC\\_private.h](#).

**7.24.2.105 HSE**

```
#define HSE 3
```

Definition at line 208 of file [MRCC\\_private.h](#).

**7.24.2.106 PLLCLK [1/2]**

```
#define PLLCLK 4
```

Definition at line 222 of file [MRCC\\_private.h](#).

**7.24.2.107 NoDivision**

```
#define NoDivision 1
```

Definition at line 211 of file [MRCC\\_private.h](#).

**7.24.2.108 DivisionBy2**

```
#define DivisionBy2 2
```

Definition at line 212 of file [MRCC\\_private.h](#).

**7.24.2.109 DivisionBy3**

```
#define DivisionBy3 3
```

Definition at line 213 of file [MRCC\\_private.h](#).

**7.24.2.110 DivisionBy4**

```
#define DivisionBy4 4
```

Definition at line 214 of file [MRCC\\_private.h](#).

**7.24.2.111 DivisionBy5**

```
#define DivisionBy5 5
```

Definition at line 215 of file [MRCC\\_private.h](#).

**7.24.2.112 I2S\_CKIN**

```
#define I2S_CKIN 2
```

Definition at line 218 of file [MRCC\\_private.h](#).

**7.24.2.113 HSI**

```
#define HSI 1
```

Definition at line 220 of file [MRCC\\_private.h](#).

**7.24.2.114 LSE**

```
#define LSE 2
```

Definition at line 221 of file [MRCC\\_private.h](#).

**7.24.2.115 PLLCLK [2/2]**

```
#define PLLCLK 4
```

Definition at line 222 of file [MRCC\\_private.h](#).

**7.24.2.116 SYSCLKby2**

```
#define SYSCLKby2 2
```

Definition at line 225 of file [MRCC\\_private.h](#).

**7.24.2.117 SYSCLKby4**

```
#define SYSCLKby4 3
```

Definition at line 226 of file [MRCC\\_private.h](#).

**7.24.2.118 SYSCLKby8**

```
#define SYSCLKby8 4
```

Definition at line 227 of file [MRCC\\_private.h](#).

**7.24.2.119 SYSCLKby16**

```
#define SYSCLKby16 5
```

Definition at line 228 of file [MRCC\\_private.h](#).

**7.24.2.120 SYSCLKby64**

```
#define SYSCLKby64 6
```

Definition at line 229 of file [MRCC\\_private.h](#).

**7.24.2.121 SYSCLKby128**

```
#define SYSCLKby128 7
```

Definition at line 230 of file [MRCC\\_private.h](#).

**7.24.2.122 SYSCLKby256**

```
#define SYSCLKby256 8
```

Definition at line 231 of file [MRCC\\_private.h](#).

**7.24.2.123 SYSCLKby512**

```
#define SYSCLKby512 9
```

Definition at line 232 of file [MRCC\\_private.h](#).

**7.24.2.124 NoCLK0**

```
#define NoCLK0 1
```

Definition at line 235 of file [MRCC\\_private.h](#).

**7.24.2.125 NoCLK1**

```
#define NoCLK1 2
```

Definition at line 236 of file [MRCC\\_private.h](#).

**7.24.2.126 HSEby2**

```
#define HSEby2 3
```

Definition at line 237 of file [MRCC\\_private.h](#).

**7.24.2.127 HSEby3**

```
#define HSEby3 4
```

Definition at line 238 of file [MRCC\\_private.h](#).

**7.24.2.128 HSEby4**

```
#define HSEby4 5
```

Definition at line 239 of file [MRCC\\_private.h](#).



**7.24.2.129 HSEby5**

```
#define HSEby5 6
```

Definition at line 240 of file [MRCC\\_private.h](#).

**7.24.2.130 HSEby6**

```
#define HSEby6 7
```

Definition at line 241 of file [MRCC\\_private.h](#).

**7.24.2.131 HSEby7**

```
#define HSEby7 8
```

Definition at line 242 of file [MRCC\\_private.h](#).

**7.24.2.132 HSEby8**

```
#define HSEby8 9
```

Definition at line 243 of file [MRCC\\_private.h](#).

**7.24.2.133 HSEby9**

```
#define HSEby9 10
```

Definition at line 244 of file [MRCC\\_private.h](#).

**7.24.2.134 HSEby10**

```
#define HSEby10 11
```

Definition at line 245 of file [MRCC\\_private.h](#).

**7.24.2.135 HSEby11**

```
#define HSEby11 12
```

Definition at line [246](#) of file [MRCC\\_private.h](#).

**7.24.2.136 HSEby12**

```
#define HSEby12 13
```

Definition at line [247](#) of file [MRCC\\_private.h](#).

**7.24.2.137 HSEby13**

```
#define HSEby13 14
```

Definition at line [248](#) of file [MRCC\\_private.h](#).

**7.24.2.138 HSEby14**

```
#define HSEby14 15
```

Definition at line [249](#) of file [MRCC\\_private.h](#).

**7.24.2.139 HSEby15**

```
#define HSEby15 16
```

Definition at line [250](#) of file [MRCC\\_private.h](#).

**7.24.2.140 HSEby16**

```
#define HSEby16 17
```

Definition at line [251](#) of file [MRCC\\_private.h](#).

**7.24.2.141 HSEby17**

```
#define HSEby17 18
```

Definition at line 252 of file [MRCC\\_private.h](#).

**7.24.2.142 HSEby18**

```
#define HSEby18 19
```

Definition at line 253 of file [MRCC\\_private.h](#).

**7.24.2.143 HSEby19**

```
#define HSEby19 20
```

Definition at line 254 of file [MRCC\\_private.h](#).

**7.24.2.144 HSEby20**

```
#define HSEby20 21
```

Definition at line 255 of file [MRCC\\_private.h](#).

**7.24.2.145 HSEby21**

```
#define HSEby21 22
```

Definition at line 256 of file [MRCC\\_private.h](#).

**7.24.2.146 HSEby22**

```
#define HSEby22 23
```

Definition at line 257 of file [MRCC\\_private.h](#).

**7.24.2.147 HSEby23**

```
#define HSEby23 24
```

Definition at line 258 of file [MRCC\\_private.h](#).

**7.24.2.148 HSEby24**

```
#define HSEby24 25
```

Definition at line 259 of file [MRCC\\_private.h](#).

**7.24.2.149 HSEby25**

```
#define HSEby25 26
```

Definition at line 260 of file [MRCC\\_private.h](#).

**7.24.2.150 HSEby26**

```
#define HSEby26 27
```

Definition at line 261 of file [MRCC\\_private.h](#).

**7.24.2.151 HSEby27**

```
#define HSEby27 28
```

Definition at line 262 of file [MRCC\\_private.h](#).

**7.24.2.152 HSEby28**

```
#define HSEby28 39
```

Definition at line 263 of file [MRCC\\_private.h](#).

**7.24.2.153 HSEby29**

```
#define HSEby29 30
```

Definition at line 264 of file [MRCC\\_private.h](#).

**7.24.2.154 HSEby30**

```
#define HSEby30 31
```

Definition at line 265 of file [MRCC\\_private.h](#).

**7.24.2.155 HSEby31**

```
#define HSEby31 32
```

Definition at line 266 of file [MRCC\\_private.h](#).

**7.24.2.156 AHBby2**

```
#define AHBby2 1
```

Definition at line 269 of file [MRCC\\_private.h](#).

**7.24.2.157 AHBby4**

```
#define AHBby4 2
```

Definition at line 270 of file [MRCC\\_private.h](#).

**7.24.2.158 AHBby8**

```
#define AHBby8 3
```

Definition at line 271 of file [MRCC\\_private.h](#).

**7.24.2.159 AHBby16**

```
#define AHBby16 4
```

Definition at line 272 of file [MRCC\\_private.h](#).

**7.24.2.160 Equal\_0**

```
#define Equal_0 0
```

Definition at line 275 of file [MRCC\\_private.h](#).

**7.24.2.161 Equal\_1**

```
#define Equal_1 1
```

Definition at line 276 of file [MRCC\\_private.h](#).

**7.24.2.162 Equal\_2**

```
#define Equal_2 2
```

Definition at line 277 of file [MRCC\\_private.h](#).

**7.24.2.163 Equal\_3**

```
#define Equal_3 3
```

Definition at line 278 of file [MRCC\\_private.h](#).

**7.24.2.164 Equal\_4**

```
#define Equal_4 4
```

Definition at line 279 of file [MRCC\\_private.h](#).

**7.24.2.165 Equal\_5**

```
#define Equal_5 5
```

Definition at line 280 of file [MRCC\\_private.h](#).

**7.24.2.166 Equal\_6**

```
#define Equal_6 6
```

Definition at line 281 of file [MRCC\\_private.h](#).

**7.24.2.167 Equal\_7**

```
#define Equal_7 7
```

Definition at line 282 of file [MRCC\\_private.h](#).

**7.24.2.168 Equal\_8**

```
#define Equal_8 8
```

Definition at line 283 of file [MRCC\\_private.h](#).

**7.24.2.169 Equal\_9**

```
#define Equal_9 9
```

Definition at line 284 of file [MRCC\\_private.h](#).

**7.24.2.170 Equal\_10**

```
#define Equal_10 10
```

Definition at line 285 of file [MRCC\\_private.h](#).

**7.24.2.171 Equal\_11**

```
#define Equal_11 11
```

Definition at line 286 of file [MRCC\\_private.h](#).

**7.24.2.172 Equal\_12**

```
#define Equal_12 12
```

Definition at line 287 of file [MRCC\\_private.h](#).

**7.24.2.173 Equal\_13**

```
#define Equal_13 13
```

Definition at line 288 of file [MRCC\\_private.h](#).

**7.24.2.174 Equal\_14**

```
#define Equal_14 14
```

Definition at line 289 of file [MRCC\\_private.h](#).

**7.24.2.175 Equal\_15**

```
#define Equal_15 15
```

Definition at line 290 of file [MRCC\\_private.h](#).

**7.24.2.176 Equal\_16**

```
#define Equal_16 16
```

Definition at line 291 of file [MRCC\\_private.h](#).



**7.24.2.177 Equal\_17**

```
#define Equal_17 17
```

Definition at line 292 of file [MRCC\\_private.h](#).

**7.24.2.178 Equal\_18**

```
#define Equal_18 18
```

Definition at line 293 of file [MRCC\\_private.h](#).

**7.24.2.179 Equal\_19**

```
#define Equal_19 19
```

Definition at line 294 of file [MRCC\\_private.h](#).

**7.24.2.180 Equal\_20**

```
#define Equal_20 20
```

Definition at line 295 of file [MRCC\\_private.h](#).

**7.24.2.181 Equal\_21**

```
#define Equal_21 21
```

Definition at line 296 of file [MRCC\\_private.h](#).

**7.24.2.182 Equal\_22**

```
#define Equal_22 22
```

Definition at line 297 of file [MRCC\\_private.h](#).

**7.24.2.183 Equal\_23**

```
#define Equal_23 23
```

Definition at line 298 of file [MRCC\\_private.h](#).

**7.24.2.184 Equal\_24**

```
#define Equal_24 24
```

Definition at line 299 of file [MRCC\\_private.h](#).

**7.24.2.185 Equal\_25**

```
#define Equal_25 25
```

Definition at line 300 of file [MRCC\\_private.h](#).

**7.24.2.186 Equal\_26**

```
#define Equal_26 26
```

Definition at line 301 of file [MRCC\\_private.h](#).

**7.24.2.187 Equal\_27**

```
#define Equal_27 27
```

Definition at line 302 of file [MRCC\\_private.h](#).

**7.24.2.188 Equal\_28**

```
#define Equal_28 28
```

Definition at line 303 of file [MRCC\\_private.h](#).

**7.24.2.189 Equal\_29**

```
#define Equal_29 29
```

Definition at line 304 of file [MRCC\\_private.h](#).

**7.24.2.190 Equal\_30**

```
#define Equal_30 30
```

Definition at line 305 of file [MRCC\\_private.h](#).

**7.24.2.191 Equal\_31**

```
#define Equal_31 31
```

Definition at line 306 of file [MRCC\\_private.h](#).

**7.24.2.192 Equal\_32**

```
#define Equal_32 32
```

Definition at line 307 of file [MRCC\\_private.h](#).

**7.24.2.193 Equal\_33**

```
#define Equal_33 33
```

Definition at line 308 of file [MRCC\\_private.h](#).

**7.24.2.194 Equal\_34**

```
#define Equal_34 34
```

Definition at line 309 of file [MRCC\\_private.h](#).

**7.24.2.195 Equal\_35**

```
#define Equal_35 35
```

Definition at line 310 of file [MRCC\\_private.h](#).

**7.24.2.196 Equal\_36**

```
#define Equal_36 36
```

Definition at line 311 of file [MRCC\\_private.h](#).

**7.24.2.197 Equal\_37**

```
#define Equal_37 37
```

Definition at line 312 of file [MRCC\\_private.h](#).

**7.24.2.198 Equal\_38**

```
#define Equal_38 38
```

Definition at line 313 of file [MRCC\\_private.h](#).

**7.24.2.199 Equal\_39**

```
#define Equal_39 39
```

Definition at line 314 of file [MRCC\\_private.h](#).

**7.24.2.200 Equal\_40**

```
#define Equal_40 40
```

Definition at line 315 of file [MRCC\\_private.h](#).

**7.24.2.201 Equal\_41**

```
#define Equal_41 41
```

Definition at line 316 of file [MRCC\\_private.h](#).

**7.24.2.202 Equal\_42**

```
#define Equal_42 42
```

Definition at line 317 of file [MRCC\\_private.h](#).

**7.24.2.203 Equal\_43**

```
#define Equal_43 43
```

Definition at line 318 of file [MRCC\\_private.h](#).

**7.24.2.204 Equal\_44**

```
#define Equal_44 44
```

Definition at line 319 of file [MRCC\\_private.h](#).

**7.24.2.205 Equal\_45**

```
#define Equal_45 45
```

Definition at line 320 of file [MRCC\\_private.h](#).

**7.24.2.206 Equal\_46**

```
#define Equal_46 46
```

Definition at line 321 of file [MRCC\\_private.h](#).

**7.24.2.207 Equal\_47**

```
#define Equal_47 47
```

Definition at line 322 of file [MRCC\\_private.h](#).

**7.24.2.208 Equal\_48**

```
#define Equal_48 48
```

Definition at line 323 of file [MRCC\\_private.h](#).

**7.24.2.209 Equal\_49**

```
#define Equal_49 49
```

Definition at line 324 of file [MRCC\\_private.h](#).

**7.24.2.210 Equal\_50**

```
#define Equal_50 50
```

Definition at line 325 of file [MRCC\\_private.h](#).

**7.24.2.211 Equal\_51**

```
#define Equal_51 51
```

Definition at line 326 of file [MRCC\\_private.h](#).

**7.24.2.212 Equal\_52**

```
#define Equal_52 52
```

Definition at line 327 of file [MRCC\\_private.h](#).

**7.24.2.213 Equal\_53**

```
#define Equal_53 53
```

Definition at line 328 of file [MRCC\\_private.h](#).

**7.24.2.214 Equal\_54**

```
#define Equal_54 54
```

Definition at line 329 of file [MRCC\\_private.h](#).

**7.24.2.215 Equal\_55**

```
#define Equal_55 55
```

Definition at line 330 of file [MRCC\\_private.h](#).

**7.24.2.216 Equal\_56**

```
#define Equal_56 56
```

Definition at line 331 of file [MRCC\\_private.h](#).

**7.24.2.217 Equal\_57**

```
#define Equal_57 57
```

Definition at line 332 of file [MRCC\\_private.h](#).

**7.24.2.218 Equal\_58**

```
#define Equal_58 58
```

Definition at line 333 of file [MRCC\\_private.h](#).

**7.24.2.219 Equal\_59**

```
#define Equal_59 59
```

Definition at line 334 of file [MRCC\\_private.h](#).

**7.24.2.220 Equal\_60**

```
#define Equal_60 60
```

Definition at line 335 of file [MRCC\\_private.h](#).

**7.24.2.221 Equal\_61**

```
#define Equal_61 61
```

Definition at line 336 of file [MRCC\\_private.h](#).

**7.24.2.222 Equal\_62**

```
#define Equal_62 62
```

Definition at line 337 of file [MRCC\\_private.h](#).

**7.24.2.223 Equal\_63**

```
#define Equal_63 63
```

Definition at line 338 of file [MRCC\\_private.h](#).



## 7.25 MRCC\_private.h

[Go to the documentation of this file.](#)

```

00001
00010 #ifndef MCAL_RCC_MRCC_PRIVATE_H_
00011 #define MCAL_RCC_MRCC_PRIVATE_H_
00012
00013
00014
00015 /*****
00016  /*                      Peripherals declaration                      */
00017  *****/
00018
00019 typedef struct
00020 {
00021
00022     u32_t CR                ;
00023     u32_t PLLCFGR           ;
00024     u32_t CFGR              ;
00025     u32_t CIR               ;
00026     u32_t AHB1RSTR          ;
00027     u32_t AHB2RSTR          ;
00028
00029     u32_t Reserved1         ;
00030     u32_t Reserved2         ;
00031
00032     u32_t APB1RSTR          ;
00033     u32_t APB2RSTR          ;
00034
00035     u32_t Reserved3         ;
00036     u32_t Reserved4         ;
00037
00038     u32_t AHB1ENR           ;
00039     u32_t AHB2ENR           ;
00040
00041     u32_t Reserved5         ;
00042     u32_t Reserved6         ;
00043
00044     u32_t APB1ENR           ;
00045     u32_t APB2ENR           ;
00046
00047     u32_t Reserved7         ;
00048     u32_t Reserved8         ;
00049
00050     u32_t AHB1LPENR         ;
00051     u32_t AHB2LPENR         ;
00052
00053     u32_t Reserved9         ;
00054     u32_t Reserved10        ;
00055
00056     u32_t APB1LPENR         ;
00057     u32_t APB2LPENR         ;
00058
00059     u32_t Reserved11        ;
00060     u32_t Reserved12        ;
00061
00062     u32_t BDCR              ;
00063     u32_t CSR               ;
00064
00065     u32_t Reserved13        ;
00066     u32_t Reserved14        ;
00067
00068     u32_t SSCGR             ;
00069     u32_t PLLI2SCFGR        ;
00070     u32_t DCKCFGR           ;
00071
00072
00073 } RCC_MemoryMapType ;
00074
00075
00076 #define RCC_BASE_ADDRESS 0x40023800
00077
00078 #define RCC ( (volatile P2VAR(RCC_MemoryMapType) ) (RCC_BASE_ADDRESS) ) // Is a pointer to the struct.
00079
00080
00081 /*****
00082  /*                      Peripheral registers                      */
00083  *****/
00084
00085 #define RCC_CR_PLLI2SRDY    27
00086 #define RCC_CR_PLLI2SON    26
00087 #define RCC_CR_PLLRDY      25
00088 #define RCC_CR_PLLON       24
00089 #define RCC_CR_CSSON       19
00090 #define RCC_CR_HSEBYP      18

```

```

00091 #define RCC_CR_HSERDY      17
00092 #define RCC_CR_HSEON        16
00093 #define RCC_CR_HSIRDY        1
00094 #define RCC_CR_HSION         0
00095
00096
00097 #define RCC_PLLCFGR_PLLQ_b0  24
00098 #define RCC_PLLCFGR_PLLQ_b1  25
00099 #define RCC_PLLCFGR_PLLQ_b2  26
00100 #define RCC_PLLCFGR_PLLQ_b3  27
00101 #define RCC_PLLCFGR_PLLSRC    22
00102 #define RCC_PLLCFGR_PLLP_b0  16
00103 #define RCC_PLLCFGR_PLLP_b1  17
00104 #define RCC_PLLCFGR_PLLN_b0    6
00105 #define RCC_PLLCFGR_PLLN_b1    7
00106 #define RCC_PLLCFGR_PLLN_b2    8
00107 #define RCC_PLLCFGR_PLLN_b3    9
00108 #define RCC_PLLCFGR_PLLN_b4   10
00109 #define RCC_PLLCFGR_PLLN_b5   11
00110 #define RCC_PLLCFGR_PLLN_b6   12
00111 #define RCC_PLLCFGR_PLLN_b7   13
00112 #define RCC_PLLCFGR_PLLN_b8   14
00113 #define RCC_PLLCFGR_PLLM_b0    0
00114 #define RCC_PLLCFGR_PLLM_b1    1
00115 #define RCC_PLLCFGR_PLLM_b2    2
00116 #define RCC_PLLCFGR_PLLM_b3    3
00117 #define RCC_PLLCFGR_PLLM_b4    4
00118 #define RCC_PLLCFGR_PLLM_b5    5
00119
00120
00121 #define RCC_CFGR_MOC2_b0       30
00122 #define RCC_CFGR_MOC2_b1       31
00123 #define RCC_CFGR_MOC2PRE_b0    27
00124 #define RCC_CFGR_MOC2PRE_b1    28
00125 #define RCC_CFGR_MOC2PRE_b2    29
00126 #define RCC_CFGR_MOC1PRE_b0    24
00127 #define RCC_CFGR_MOC1PRE_b1    25
00128 #define RCC_CFGR_MOC1PRE_b2    26
00129 #define RCC_CFGR_I2SSRC        23
00130 #define RCC_CFGR_MOC1_b0       21
00131 #define RCC_CFGR_MOC1_b1       22
00132 #define RCC_CFGR_RTCPRE_b0     16
00133 #define RCC_CFGR_RTCPRE_b1     17
00134 #define RCC_CFGR_RTCPRE_b2     18
00135 #define RCC_CFGR_RTCPRE_b3     19
00136 #define RCC_CFGR_RTCPRE_b4     20
00137 #define RCC_CFGR_PPRE2_b0      13
00138 #define RCC_CFGR_PPRE2_b1      14
00139 #define RCC_CFGR_PPRE2_b2      15
00140 #define RCC_CFGR_PPRE1_b0      10
00141 #define RCC_CFGR_PPRE1_b1      11
00142 #define RCC_CFGR_PPRE1_b2      12
00143 #define RCC_CFGR_HPRE_b0        4
00144 #define RCC_CFGR_HPRE_b1        5
00145 #define RCC_CFGR_HPRE_b2        6
00146 #define RCC_CFGR_HPRE_b3        7
00147 #define RCC_CFGR_SWS_b0         2
00148 #define RCC_CFGR_SWS_b1         3
00149 #define RCC_CFGR_SW_b0          0
00150 #define RCC_CFGR_SW_b1          1
00151
00152
00153 #define RCC_AHB1ENR_DMA2EN      22
00154 #define RCC_AHB1ENR_DMA1EN      21
00155 #define RCC_AHB1ENR_CRCEN       12
00156 #define RCC_AHB1ENR_GPIOHEN     7
00157 #define RCC_AHB1ENR_GPIOEEN     4
00158 #define RCC_AHB1ENR_GPIODEN     3
00159 #define RCC_AHB1ENR_GPIOCEN     2
00160 #define RCC_AHB1ENR_GPIOBEN     1
00161 #define RCC_AHB1ENR_GPIOAEN     0
00162
00163
00164 #define RCC_AHB2ENR_OTGFSEN      7
00165
00166
00167 #define RCC_APB1ENR_PWREN        28
00168 #define RCC_APB1ENR_I2C3EN       23
00169 #define RCC_APB1ENR_I2C2EN       22
00170 #define RCC_APB1ENR_I2C1EN       21
00171 #define RCC_APB1ENR_USART2EN     17
00172 #define RCC_APB1ENR_SPI3EN       15
00173 #define RCC_APB1ENR_SPI2EN       14
00174 #define RCC_APB1ENR_WWDGEN       11
00175 #define RCC_APB1ENR_TIM5EN        3
00176 #define RCC_APB1ENR_TIM4EN        2
00177 #define RCC_APB1ENR_TIM3EN        1

```

```
00178 #define RCC_APB1ENR_TIM2EN      0
00179
00180
00181 #define RCC_APB2ENR_TIM11EN      18
00182 #define RCC_APB2ENR_TIM10EN      17
00183 #define RCC_APB2ENR_TIM9EN       16
00184 #define RCC_APB2ENR_SYSCFGEN     14
00185 #define RCC_APB2ENR_SPI4EN       13
00186 #define RCC_APB2ENR_SPI1EN       12
00187 #define RCC_APB2ENR_SDIOEN        11
00188 #define RCC_APB2ENR_ADC1EN        8
00189 #define RCC_APB2ENR_USART6EN      5
00190 #define RCC_APB2ENR_USART1EN      4
00191 #define RCC_APB2ENR_TIM1EN        0
00192
00193 #define RCC_AHB1LPENR_FLITFLEN    15
00194
00195
00196 /*****
00197 /*                               Config.h Macros                               */
00198 *****/
00199
00200 #define ENABLE      1
00201 #define DISABLE     2
00202
00203 #define BYBASED     1
00204 #define NOTBYBASED  2
00205
00206 #define SYSCLK      1
00207 #define PLLI2SCLK   5
00208 #define HSE         3
00209 #define PLLCLK      4
00210
00211 #define NoDivision   1
00212 #define DivisionBy2  2
00213 #define DivisionBy3  3
00214 #define DivisionBy4  4
00215 #define DivisionBy5  5
00216
00217
00218 #define I2S_CKIN     2
00219
00220 #define HSI          1
00221 #define LSE          2
00222 #define PLLCLK      4
00223
00224 // AHB prescalers:
00225 #define SYSCLKby2    2    //(0b1000)
00226 #define SYSCLKby4    3    //(0b1001)
00227 #define SYSCLKby8    4    //(0b1010)
00228 #define SYSCLKby16   5    //(0b1011)
00229 #define SYSCLKby64   6    //(0b1100)
00230 #define SYSCLKby128  7    //(0b1101)
00231 #define SYSCLKby256  8    //(0b1110)
00232 #define SYSCLKby512  9    //(0b1111)
00233
00234
00235 #define NoCLK0       1
00236 #define NoCLK1       2
00237 #define HSEby2       3
00238 #define HSEby3       4
00239 #define HSEby4       5
00240 #define HSEby5       6
00241 #define HSEby6       7
00242 #define HSEby7       8
00243 #define HSEby8       9
00244 #define HSEby9      10
00245 #define HSEby10     11
00246 #define HSEby11     12
00247 #define HSEby12     13
00248 #define HSEby13     14
00249 #define HSEby14     15
00250 #define HSEby15     16
00251 #define HSEby16     17
00252 #define HSEby17     18
00253 #define HSEby18     19
00254 #define HSEby19     20
00255 #define HSEby20     21
00256 #define HSEby21     22
00257 #define HSEby22     23
00258 #define HSEby23     24
00259 #define HSEby24     25
00260 #define HSEby25     26
00261 #define HSEby26     27
00262 #define HSEby27     28
00263 #define HSEby28     29
00264 #define HSEby29     30
```

```
00265 #define HSEby30 31
00266 #define HSEby31 32
00267
00268
00269 #define AHBby2 1
00270 #define AHBby4 2
00271 #define AHBby8 3
00272 #define AHBby16 4
00273
00274
00275 #define Equal_0 0
00276 #define Equal_1 1
00277 #define Equal_2 2
00278 #define Equal_3 3
00279 #define Equal_4 4
00280 #define Equal_5 5
00281 #define Equal_6 6
00282 #define Equal_7 7
00283 #define Equal_8 8
00284 #define Equal_9 9
00285 #define Equal_10 10
00286 #define Equal_11 11
00287 #define Equal_12 12
00288 #define Equal_13 13
00289 #define Equal_14 14
00290 #define Equal_15 15
00291 #define Equal_16 16
00292 #define Equal_17 17
00293 #define Equal_18 18
00294 #define Equal_19 19
00295 #define Equal_20 20
00296 #define Equal_21 21
00297 #define Equal_22 22
00298 #define Equal_23 23
00299 #define Equal_24 24
00300 #define Equal_25 25
00301 #define Equal_26 26
00302 #define Equal_27 27
00303 #define Equal_28 28
00304 #define Equal_29 29
00305 #define Equal_30 30
00306 #define Equal_31 31
00307 #define Equal_32 32
00308 #define Equal_33 33
00309 #define Equal_34 34
00310 #define Equal_35 35
00311 #define Equal_36 36
00312 #define Equal_37 37
00313 #define Equal_38 38
00314 #define Equal_39 39
00315 #define Equal_40 40
00316 #define Equal_41 41
00317 #define Equal_42 42
00318 #define Equal_43 43
00319 #define Equal_44 44
00320 #define Equal_45 45
00321 #define Equal_46 46
00322 #define Equal_47 47
00323 #define Equal_48 48
00324 #define Equal_49 49
00325 #define Equal_50 50
00326 #define Equal_51 51
00327 #define Equal_52 52
00328 #define Equal_53 53
00329 #define Equal_54 54
00330 #define Equal_55 55
00331 #define Equal_56 56
00332 #define Equal_57 57
00333 #define Equal_58 58
00334 #define Equal_59 59
00335 #define Equal_60 60
00336 #define Equal_61 61
00337 #define Equal_62 62
00338 #define Equal_63 63
00339
00340
00341
00342
00343
00344 #endif /* MCAL_RCC_MRCC_PRIVATE_H_ */
```

## 7.26 src/COTS/MCAL/RCC/MRCC\_program.c File Reference

This file contains the source code of the interfacing for the RCC module.

```
#include "../..../LIB/LSTD_TYPES.h"
#include "../..../LIB/LSTD_COMPILER.h"
#include "../..../LIB/LSTD_VALUES.h"
#include "../..../LIB/LSTD_BITMATH.h"
#include "MRCC_private.h"
#include "MRCC_interface.h"
#include "MRCC_config.h"
```

### Functions

- void [MRCC\\_vInit](#) (void)
- void [MRCC\\_vEnablePeriphralCLK](#) ([u32\\_t](#) A\_u32BusID, [u32\\_t](#) A\_u32PeriphralID)
- void [MRCC\\_vDisablePeriphralCLK](#) ([u32\\_t](#) A\_u32BusID, [u32\\_t](#) A\_u32PeriphralID)

### 7.26.1 Detailed Description

This file contains the source code of the interfacing for the RCC module.

#### Author

Ali El Bana & Mo Alaa

#### Version

2.0

#### Date

11/9/2022

Definition in file [MRCC\\_program.c](#).

### 7.26.2 Function Documentation

### 7.26.2.1 MRCC\_vInit()

```
void MRCC_vInit (
    void )
```

Definition at line 29 of file [MRCC\\_program.c](#).

```
00030 {
00031
00032     // PLLI2S (ON/OFF).
00033     #if PLLI2S == ENABLE
00034
00035         SET_BIT( RCC->CR, RCC_CR_PLLI2SON ) ;
00036
00037     #elif PLLI2S == DISABLE
00038
00039         CLR_BIT( RCC->CR, RCC_CR_PLLI2SON ) ;
00040
00041     #endif
00042
00043
00044     // PLL (ON/OFF).
00045     #if PLL == ENABLE
00046
00047         SET_BIT( RCC->CR, RCC_CR_PLLON ) ;
00048
00049     #elif PLL == DISABLE
00050
00051         CLR_BIT( RCC->CR, RCC_CR_PLLON ) ;
00052
00053     #endif
00054
00055
00056     // CSS (ON/OFF).
00057     #if CSS == ENABLE
00058
00059         SET_BIT( RCC->CR, RCC_CR_CSSON ) ;
00060
00061     #elif CSS == DISABLE
00062
00063         CLR_BIT( RCC->CR, RCC_CR_CSSON ) ;
00064
00065     #endif
00066
00067
00068     // HSEBYP.
00069     #if HSEBYP == BYBASED
00070
00071         SET_BIT( RCC->CR, RCC_CR_HSEBYP ) ;
00072
00073     #elif HSEBYP == NOTBYBASED
00074
00075         CLR_BIT( RCC->CR, RCC_CR_HSEBYP ) ;
00076
00077     #endif
00078
00079
00080     // Select CLK switch (HSI/HSE/PLL).
00081     #if SW == HSI
00082
00083         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b0 ) ;
00084         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b1 ) ;
00085
00086     #elif SW == HSE
00087
00088         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b0 ) ;
00089         SET_BIT( RCC->CFGR, RCC_CFGR_SW_b1 ) ;
00090
00091     #elif SW == PLLCLK
00092
00093         SET_BIT( RCC->CFGR, RCC_CFGR_SW_b0 ) ;
00094         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b1 ) ;
00095
00096     #endif
00097
00098
00099     // MCO2 selection:
00100     #if MCO2 == SYSCLK
00101
00102         CLR_BIT( RCC->CFGR, RCC_CFGR_MCO2_b0 ) ;
00103         CLR_BIT( RCC->CFGR, RCC_CFGR_MCO2_b1 ) ;
00104
00105     #elif MCO2 == PLLI2SCLK
00106
00107         CLR_BIT( RCC->CFGR, RCC_CFGR_MCO2_b0 ) ;
```

```
00108         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b1 ) ;
00109
00110 #elif MCO2 == HSE
00111
00112         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b0 ) ;
00113         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2_b1 ) ;
00114
00115 #elif MCO2 == PLLCLK
00116
00117         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b0 ) ;
00118         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b1 ) ;
00119
00120 #endif
00121
00122
00123 // MCO2 prescaler:
00124 #if MCO2PRE == NoDivision
00125
00126         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00127         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00128         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00129
00130 #elif MCO2PRE == DivisionBy2
00131
00132         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00133         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00134         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00135
00136 #elif MCO2PRE == DivisionBy3
00137
00138         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00139         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00140         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00141
00142 #elif MCO2PRE == DivisionBy4
00143
00144         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00145         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00146         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00147
00148 #elif MCO2PRE == DivisionBy5
00149
00150         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00151         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00152         SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00153
00154 #endif
00155
00156
00157 // MCO1 selection:
00158 #if MCO1 == HSI
00159
00160         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00161         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00162
00163 #elif MCO1 == LSE
00164
00165         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00166         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00167
00168 #elif MCO1 == HSE
00169
00170         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00171         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00172
00173 #elif MCO1 == PLLCLK
00174
00175         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00176         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00177
00178 #endif
00179
00180
00181 // MCO1 prescaler:
00182 #if MCO1PRE == NoDivision
00183
00184         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00185         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00186         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00187
00188 #elif MCO1PRE == DivisionBy2
00189
00190         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00191         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00192         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00193
00194 #elif MCO1PRE == DivisionBy3
```

```

00195
00196     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00197     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00198     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00199
00200 #elif MOC1PRE == DivisionBy4
00201
00202     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00203     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00204     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00205
00206 #elif MOC1PRE == DivisionBy5
00207
00208     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00209     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00210     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00211
00212 #endif
00213
00214 // AHB prescalers:
00215 #if HPRE == NoDivision
00216
00217
00218     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00219     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00220     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00221     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00222
00223 #elif HPRE == SYSCLKby2
00224
00225     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00226     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00227     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00228     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00229
00230 #elif HPRE == SYSCLKby4
00231
00232     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00233     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00234     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00235     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00236
00237 #elif HPRE == SYSCLKby8
00238
00239     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00240     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00241     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00242     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00243
00244 #elif HPRE == SYSCLKby16
00245
00246     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00247     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00248     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00249     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00250
00251 #elif HPRE == SYSCLKby64
00252
00253     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00254     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00255     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00256     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00257
00258 #elif HPRE == SYSCLKby128
00259
00260     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00261     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00262     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00263     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00264
00265 #elif HPRE == SYSCLKby256
00266
00267     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00268     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00269     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00270     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00271
00272 #elif HPRE == SYSCLKby512
00273
00274     SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00275     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00276     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00277     CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00278
00279 #endif
00280
00281

```



```

00282     // APB1 prescalers:
00283     #if PPRE1 == NoDivision
00284
00285         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 );
00286         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 );
00287         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 );
00288
00289     #elif PPRE1 == AHBby2
00290
00291         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 );
00292         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 );
00293         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 );
00294
00295     #elif PPRE1 == AHBby4
00296
00297         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 );
00298         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 );
00299         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 );
00300
00301     #elif PPRE1 == AHBby8
00302
00303         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 );
00304         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 );
00305         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 );
00306
00307     #elif PPRE1 == AHBby16
00308
00309         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 );
00310         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 );
00311         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 );
00312
00313     #endif
00314
00315     // APB2 prescalers:
00316     #if PPRE2 == NoDivision
00317
00318         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 );
00319         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 );
00320         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 );
00321
00322     #elif PPRE2 == AHBby2
00323
00324         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 );
00325         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 );
00326         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 );
00327
00328     #elif PPRE2 == AHBby4
00329
00330         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 );
00331         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 );
00332         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 );
00333
00334     #elif PPRE2 == AHBby8
00335
00336         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 );
00337         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 );
00338         CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 );
00339
00340     #elif PPRE2 == AHBby16
00341
00342         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 );
00343         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 );
00344         SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 );
00345
00346     #endif
00347
00348
00349     // PLL configurations:
00350
00351
00352
00353     // Enable the selected CLK (HSI ON/HSE ON/PLL ON): //
00354     // HSE (ON/OFF).
00355     #if HSE_EN == ENABLE
00356
00357         SET_BIT( RCC->CR, RCC_CR_HSEON );
00358
00359     #elif HSE_EN == DISABLE
00360
00361         CLR_BIT( RCC->CR, RCC_CR_HSEON );
00362
00363     #endif
00364
00365
00366     // HSI (ON/OFF).
00367     #if HSI_EN == ENABLE
00368

```

```

00369         SET_BIT( RCC->CR, RCC_CR_HSION ) ;
00370
00371 #elif HSI_EN == DISABLE
00372
00373         CLR_BIT( RCC->CR, RCC_CR_HSION ) ;
00374
00375 #endif
00376
00377 }

```

References [CLR\\_BIT](#), [RCC](#), [RCC\\_CFGR\\_HPRE\\_b0](#), [RCC\\_CFGR\\_HPRE\\_b1](#), [RCC\\_CFGR\\_HPRE\\_b2](#), [RCC\\_CFGR\\_HPRE\\_b3](#), [RCC\\_CFGR\\_MOC1\\_b0](#), [RCC\\_CFGR\\_MOC1\\_b1](#), [RCC\\_CFGR\\_MOC1PRE\\_b0](#), [RCC\\_CFGR\\_MOC1PRE\\_b1](#), [RCC\\_CFGR\\_MOC1PRE\\_b2](#), [RCC\\_CFGR\\_MOC2\\_b0](#), [RCC\\_CFGR\\_MOC2\\_b1](#), [RCC\\_CFGR\\_MOC2PRE\\_b0](#), [RCC\\_CFGR\\_MOC2PRE\\_b1](#), [RCC\\_CFGR\\_MOC2PRE\\_b2](#), [RCC\\_CFGR\\_PPRE1\\_b0](#), [RCC\\_CFGR\\_PPRE1\\_b1](#), [RCC\\_CFGR\\_PPRE1\\_b2](#), [RCC\\_CFGR\\_PPRE2\\_b0](#), [RCC\\_CFGR\\_PPRE2\\_b1](#), [RCC\\_CFGR\\_PPRE2\\_b2](#), [RCC\\_CFGR\\_SW\\_b0](#), [RCC\\_CFGR\\_SW\\_b1](#), [RCC\\_CR\\_CSSON](#), [RCC\\_CR\\_HSEBYP](#), [RCC\\_CR\\_HSEON](#), [RCC\\_CR\\_HSION](#), [RCC\\_CR\\_PLLI2SON](#), [RCC\\_CR\\_PLLON](#), and [SET\\_BIT](#).

### 7.26.2.2 MRCC\_vEnablePeripheralCLK()

```

void MRCC_vEnablePeripheralCLK (
    u32_t A_u32BusID,
    u32_t A_u32PeripheralID )

```

Definition at line 382 of file [MRCC\\_program.c](#).

```

00383 {
00384
00385     switch( A_u32BusID )
00386     {
00387
00388         case RCC_AHB1 :
00389
00390             SET_BIT( RCC->AHB1ENR, A_u32PeripheralID ) ;
00391
00392             break ;
00393
00394         case RCC_AHB2 :
00395
00396             SET_BIT( RCC->AHB2ENR, A_u32PeripheralID ) ;
00397
00398             break ;
00399
00400         case RCC_APB1 :
00401
00402             SET_BIT( RCC->APB1ENR, A_u32PeripheralID ) ;
00403
00404             break ;
00405
00406         case RCC_APB2 :
00407
00408             SET_BIT( RCC->APB2ENR, A_u32PeripheralID ) ;
00409
00410             break ;
00411
00412         case RCC_AHB1LPENR:
00413
00414             SET_BIT( RCC->AHB1LPENR, A_u32PeripheralID ) ;
00415
00416             break ;
00417
00418         default:
00419
00420             // Error wrong Bus ID
00421
00422             break ;
00423
00424     }
00425 }
00426
00427 }
00428
00429 }
00430 }

```

References [RCC](#), [RCC\\_AHB1](#), [RCC\\_AHB1LPENR](#), [RCC\\_AHB2](#), [RCC\\_APB1](#), [RCC\\_APB2](#), and [SET\\_BIT](#).

## 7.26.2.3 MRCC\_vDisablePeripheralCLK()

```
void MRCC_vDisablePeripheralCLK (
    u32_t A_u32BusID,
    u32_t A_u32PeripheralID )
```

Definition at line 435 of file [MRCC\\_program.c](#).

```
00436 {
00437
00438     switch( A_u32BusID )
00439     {
00440
00441         case RCC_AHB1 :
00442
00443             CLR_BIT( RCC->AHB1ENR, A_u32PeripheralID ) ;
00444
00445             break ;
00446
00447         case RCC_AHB2 :
00448
00449             CLR_BIT( RCC->AHB2ENR, A_u32PeripheralID ) ;
00450
00451             break ;
00452
00453         case RCC_APB1 :
00454
00455             CLR_BIT( RCC->APB1ENR, A_u32PeripheralID ) ;
00456
00457             break ;
00458
00459         case RCC_APB2 :
00460
00461             CLR_BIT( RCC->APB2ENR, A_u32PeripheralID ) ;
00462
00463             break ;
00464
00465         case RCC_AHB1LPENR:
00466
00467             CLR_BIT( RCC->AHB1LPENR, A_u32PeripheralID ) ;
00468
00469             break ;
00470
00471         default:
00472             // Error wrong Bus ID
00473
00474             break ;
00475
00476     }
00477
00478     break ;
00479
00480     }
00481
00482     }
00483
00484     }
00485 }
```

References [CLR\\_BIT](#), [RCC](#), [RCC\\_AHB1](#), [RCC\\_AHB1LPENR](#), [RCC\\_AHB2](#), [RCC\\_APB1](#), and [RCC\\_APB2](#).

## 7.27 MRCC\_program.c

[Go to the documentation of this file.](#)

```
00001
00011 /*****
00012 /*          Include headers          */
00013 /*****
00014 #include "../LIB/LSTD_TYPES.h"
00015 #include "../LIB/LSTD_COMPILER.h"
00016 #include "../LIB/LSTD_VALUES.h"
00017 #include "../LIB/LSTD_BITMATH.h"
00018
00019 #include "MRCC_private.h"
00020 #include "MRCC_interface.h"
00021 #include "MRCC_config.h"
00022
```

```

00023
00024
00025 /*****
00026 /*          Functions implementations          */
00027 /*****
00028
00029 FUNC(void) MRCC_vInit( void )
00030 {
00031
00032     // PLLI2S (ON/OFF).
00033     #if PLLI2S == ENABLE
00034
00035         SET_BIT( RCC->CR, RCC_CR_PLLI2SON ) ;
00036
00037     #elif PLLI2S == DISABLE
00038
00039         CLR_BIT( RCC->CR, RCC_CR_PLLI2SON ) ;
00040
00041     #endif
00042
00043
00044     // PLL (ON/OFF).
00045     #if PLL == ENABLE
00046
00047         SET_BIT( RCC->CR, RCC_CR_PLLON ) ;
00048
00049     #elif PLL == DISABLE
00050
00051         CLR_BIT( RCC->CR, RCC_CR_PLLON ) ;
00052
00053     #endif
00054
00055
00056     // CSS (ON/OFF).
00057     #if CSS == ENABLE
00058
00059         SET_BIT( RCC->CR, RCC_CR_CSSON ) ;
00060
00061     #elif CSS == DISABLE
00062
00063         CLR_BIT( RCC->CR, RCC_CR_CSSON ) ;
00064
00065     #endif
00066
00067
00068     // HSEBYP.
00069     #if HSEBYP == BYBASED
00070
00071         SET_BIT( RCC->CR, RCC_CR_HSEBYP ) ;
00072
00073     #elif HSEBYP == NOTBYBASED
00074
00075         CLR_BIT( RCC->CR, RCC_CR_HSEBYP ) ;
00076
00077     #endif
00078
00079
00080     // Select CLK switch (HSI/HSE/PLL).
00081     #if SW == HSI
00082
00083         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b0 ) ;
00084         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b1 ) ;
00085
00086     #elif SW == HSE
00087
00088         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b0 ) ;
00089         SET_BIT( RCC->CFGR, RCC_CFGR_SW_b1 ) ;
00090
00091     #elif SW == PLLCLK
00092
00093         SET_BIT( RCC->CFGR, RCC_CFGR_SW_b0 ) ;
00094         CLR_BIT( RCC->CFGR, RCC_CFGR_SW_b1 ) ;
00095
00096     #endif
00097
00098
00099     // MCO2 selection:
00100     #if MCO2 == SYSCLK
00101
00102         CLR_BIT( RCC->CFGR, RCC_CFGR_MCO2_b0 ) ;
00103         CLR_BIT( RCC->CFGR, RCC_CFGR_MCO2_b1 ) ;
00104
00105     #elif MCO2 == PLLI2SCLK
00106
00107         CLR_BIT( RCC->CFGR, RCC_CFGR_MCO2_b0 ) ;
00108         SET_BIT( RCC->CFGR, RCC_CFGR_MCO2_b1 ) ;
00109

```

```

00110 #elif MCO2 == HSE
00111
00112     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b0 ) ;
00113     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2_b1 ) ;
00114
00115 #elif MCO2 == PLLCLK
00116
00117     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b0 ) ;
00118     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2_b1 ) ;
00119
00120 #endif
00121
00122
00123 // MCO2 prescaler:
00124 #if MCO2PRE == NoDivision
00125
00126     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00127     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00128     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00129
00130 #elif MCO2PRE == DivisionBy2
00131
00132     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00133     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00134     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00135
00136 #elif MCO2PRE == DivisionBy3
00137
00138     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00139     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00140     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00141
00142 #elif MCO2PRE == DivisionBy4
00143
00144     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00145     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00146     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00147
00148 #elif MCO2PRE == DivisionBy5
00149
00150     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b0 ) ;
00151     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b1 ) ;
00152     SET_BIT( RCC->CFGR, RCC_CFGR_MOC2PRE_b2 ) ;
00153
00154 #endif
00155
00156 // MCO1 selection:
00157 #if MCO1 == HSI
00158
00159     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00160     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00161
00162 #elif MCO1 == LSE
00163
00164     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00165     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00166
00167 #elif MCO1 == HSE
00168
00169     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00170     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00171
00172 #elif MCO1 == PLLCLK
00173
00174     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b0 ) ;
00175     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1_b1 ) ;
00176
00177 #endif
00178
00179
00180 // MCO1 prescaler:
00181 #if MCO1PRE == NoDivision
00182
00183     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00184     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00185     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00186
00187 #elif MCO1PRE == DivisionBy2
00188
00189     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00190     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00191     CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00192
00193 #elif MCO1PRE == DivisionBy3
00194
00195     SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00196

```

```

00197         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00198         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00199
00200 #elif MCO1PRE == DivisionBy4
00201
00202         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00203         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00204         CLR_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00205
00206 #elif MCO1PRE == DivisionBy5
00207
00208         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b0 ) ;
00209         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b1 ) ;
00210         SET_BIT( RCC->CFGR, RCC_CFGR_MOC1PRE_b2 ) ;
00211
00212 #endif
00213
00214
00215 // AHB prescalers:
00216 #if HPRE == NoDivision
00217
00218         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00219         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00220         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00221         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00222
00223 #elif HPRE == SYSCLKby2
00224
00225         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00226         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00227         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00228         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00229
00230 #elif HPRE == SYSCLKby4
00231
00232         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00233         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00234         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00235         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00236
00237 #elif HPRE == SYSCLKby8
00238
00239         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00240         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00241         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00242         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00243
00244 #elif HPRE == SYSCLKby16
00245
00246         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00247         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00248         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00249         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00250
00251 #elif HPRE == SYSCLKby64
00252
00253         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00254         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00255         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00256         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00257
00258 #elif HPRE == SYSCLKby128
00259
00260         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00261         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00262         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00263         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00264
00265 #elif HPRE == SYSCLKby256
00266
00267         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00268         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00269         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00270         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00271
00272 #elif HPRE == SYSCLKby512
00273
00274         SET_BIT( RCC->CFGR, RCC_CFGR_HPRE_b0 ) ;
00275         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b1 ) ;
00276         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b2 ) ;
00277         CLR_BIT( RCC->CFGR, RCC_CFGR_HPRE_b3 ) ;
00278
00279 #endif
00280
00281
00282 // APB1 prescalers:
00283 #if PPRES1 == NoDivision

```

```

00284
00285     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00286     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00287     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00288
00289 #elif PPRE1 == AHBby2
00290
00291     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00292     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00293     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00294
00295 #elif PPRE1 == AHBby4
00296
00297     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00298     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00299     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00300
00301 #elif PPRE1 == AHBby8
00302
00303     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00304     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00305     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00306
00307 #elif PPRE1 == AHBby16
00308
00309     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b0 ) ;
00310     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b1 ) ;
00311     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE1_b2 ) ;
00312
00313 #endif
00314
00315     // APB2 prescalers:
00316 #if PPRE2 == NoDivision
00317
00318     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00319     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00320     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00321
00322 #elif PPRE2 == AHBby2
00323
00324     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00325     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00326     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00327
00328 #elif PPRE2 == AHBby4
00329
00330     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00331     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00332     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00333
00334 #elif PPRE2 == AHBby8
00335
00336     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00337     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00338     CLR_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00339
00340 #elif PPRE2 == AHBby16
00341
00342     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b0 ) ;
00343     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b1 ) ;
00344     SET_BIT( RCC->CFGR, RCC_CFGR_PPRE2_b2 ) ;
00345
00346 #endif
00347
00348
00349     // PLL configurations:
00350
00351
00352
00353     // Enable the selected CLK (HSI ON/HSE ON/PLL ON): //
00354     // HSE (ON/OFF).
00355 #if HSE_EN == ENABLE
00356
00357     SET_BIT( RCC->CR, RCC_CR_HSEON ) ;
00358
00359 #elif HSE_EN == DISABLE
00360
00361     CLR_BIT( RCC->CR, RCC_CR_HSEON ) ;
00362
00363 #endif
00364
00365
00366     // HSI (ON/OFF).
00367 #if HSI_EN == ENABLE
00368
00369     SET_BIT( RCC->CR, RCC_CR_HSION ) ;
00370

```

```

00371 #elif HSI_EN == DISABLE
00372
00373     CLR_BIT( RCC->CR, RCC_CR_HSION ) ;
00374
00375 #endif
00376
00377 }
00378
00379 /*****/
00380 /*****/
00381
00382 FUNC(void) MRCC_vEnablePeriphralCLK( VAR(u32_t) A_u32BusID, VAR(u32_t) A_u32PeriphralID )
00383 {
00384     switch( A_u32BusID )
00385     {
00386     case RCC_AHB1 :
00387         SET_BIT( RCC->AHB1ENR, A_u32PeriphralID ) ;
00388
00389         break ;
00390
00391     case RCC_AHB2 :
00392         SET_BIT( RCC->AHB2ENR, A_u32PeriphralID ) ;
00393
00394         break ;
00395
00396     case RCC_APB1 :
00397         SET_BIT( RCC->APB1ENR, A_u32PeriphralID ) ;
00398
00399         break ;
00400
00401     case RCC_APB2 :
00402         SET_BIT( RCC->APB2ENR, A_u32PeriphralID ) ;
00403
00404         break ;
00405
00406     case RCC_AHB1LPENR:
00407         SET_BIT( RCC->AHB1LPENR, A_u32PeriphralID ) ;
00408
00409         break ;
00410
00411     default:
00412         // Error wrong Bus ID
00413
00414         break ;
00415     }
00416 }
00417
00418
00419 /*****/
00420 /*****/
00421
00422 FUNC(void) MRCC_vDisablePeriphralCLK( VAR(u32_t) A_u32BusID, VAR(u32_t) A_u32PeriphralID )
00423 {
00424     switch( A_u32BusID )
00425     {
00426     case RCC_AHB1 :
00427         CLR_BIT( RCC->AHB1ENR, A_u32PeriphralID ) ;
00428
00429         break ;
00430
00431     case RCC_AHB2 :
00432         CLR_BIT( RCC->AHB2ENR, A_u32PeriphralID ) ;
00433
00434         break ;
00435
00436     case RCC_APB1 :

```



```
00458         CLR_BIT( RCC->APB1ENR, A_u32PeripheralID ) ;
00459
00460     break ;
00461
00462
00463     case RCC_APB2 :
00464
00465         CLR_BIT( RCC->APB2ENR, A_u32PeripheralID ) ;
00466
00467     break ;
00468
00469
00470     case RCC_AHB1LPENR:
00471
00472         CLR_BIT( RCC->AHB1LPENR, A_u32PeripheralID ) ;
00473
00474     break ;
00475
00476
00477     default:
00478
00479         // Error wrong Bus ID
00480
00481     break ;
00482
00483 }
00484
00485 }
00486
00487 /*****/
00488 /*****/
00489
00490
00491
00492
00493
00494
00495
00496
00497
```



# Index

ADC1EN  
    MRCC\_config.h, 93

AF\_Type  
    MGPIOx\_ConfigType, 44

AFRHx  
    GPIOx\_MemoryMapType, 41

AFRLx  
    GPIOx\_MemoryMapType, 41

AHB1ENR  
    RCC\_MemoryMapType, 47

AHB1ENR\_CRCEN  
    MRCC\_interface.h, 104

AHB1ENR\_DMA1EN  
    MRCC\_interface.h, 104

AHB1ENR\_DMA2EN  
    MRCC\_interface.h, 104

AHB1ENR\_GPIOAEN  
    MRCC\_interface.h, 105

AHB1ENR\_GPIOBEN  
    MRCC\_interface.h, 105

AHB1ENR\_GPIOCEN  
    MRCC\_interface.h, 105

AHB1ENR\_GPIODEN  
    MRCC\_interface.h, 105

AHB1ENR\_GPIOEEN  
    MRCC\_interface.h, 104

AHB1ENR\_GPIOHEN  
    MRCC\_interface.h, 104

AHB1LPENR  
    RCC\_MemoryMapType, 48

AHB1LPENR\_FLITFLPEN  
    MRCC\_interface.h, 109

AHB1RSTR  
    RCC\_MemoryMapType, 46

AHB2ENR  
    RCC\_MemoryMapType, 47

AHB2ENR\_OTGFSEN  
    MRCC\_interface.h, 105

AHB2LPENR  
    RCC\_MemoryMapType, 48

AHB2RSTR  
    RCC\_MemoryMapType, 46

AHBby16  
    MRCC\_private.h, 147

AHBby2  
    MRCC\_private.h, 147

AHBby4  
    MRCC\_private.h, 147

AHBby8  
    MRCC\_private.h, 147

APB1ENR  
    RCC\_MemoryMapType, 48

APB1ENR\_I2C1EN  
    MRCC\_interface.h, 106

APB1ENR\_I2C2EN  
    MRCC\_interface.h, 106

APB1ENR\_I2C3EN  
    MRCC\_interface.h, 106

APB1ENR\_PWREN  
    MRCC\_interface.h, 105

APB1ENR\_SPI2EN  
    MRCC\_interface.h, 106

APB1ENR\_SPI3EN  
    MRCC\_interface.h, 106

APB1ENR\_TIM2EN  
    MRCC\_interface.h, 107

APB1ENR\_TIM3EN  
    MRCC\_interface.h, 107

APB1ENR\_TIM4EN  
    MRCC\_interface.h, 107

APB1ENR\_TIM5EN  
    MRCC\_interface.h, 107

APB1ENR\_USART2EN  
    MRCC\_interface.h, 106

APB1ENR\_WWDGEN  
    MRCC\_interface.h, 107

APB1LPENR  
    RCC\_MemoryMapType, 49

APB1RSTR  
    RCC\_MemoryMapType, 46

APB2ENR  
    RCC\_MemoryMapType, 48

APB2ENR\_ADC1EN  
    MRCC\_interface.h, 109

APB2ENR\_SDIOEN  
    MRCC\_interface.h, 108

APB2ENR\_SPI1EN  
    MRCC\_interface.h, 108

APB2ENR\_SPI4EN  
    MRCC\_interface.h, 108

APB2ENR\_SYSCFGEN  
    MRCC\_interface.h, 108

APB2ENR\_TIM10EN  
    MRCC\_interface.h, 108

APB2ENR\_TIM11EN  
    MRCC\_interface.h, 107

APB2ENR\_TIM1EN  
    MRCC\_interface.h, 109

- APB2ENR\_TIM9EN
  - MRCC\_interface.h, [108](#)
- APB2ENR\_USART1EN
  - MRCC\_interface.h, [109](#)
- APB2ENR\_USART6EN
  - MRCC\_interface.h, [109](#)
- APB2LPENR
  - RCC\_MemoryMapType, [49](#)
- APB2RSTR
  - RCC\_MemoryMapType, [46](#)
- BDCR
  - RCC\_MemoryMapType, [50](#)
- Bit Group Manipulation Math Macros, [10](#)
  - CLR\_BITs, [11](#)
  - GET\_BITs, [11](#)
  - SET\_BITs, [11](#)
  - TOGGLE\_BITs, [11](#)
- Bit Manipulation Math Macros, [9](#)
  - CLR\_BIT, [9](#)
  - GET\_BIT, [10](#)
  - SET\_BIT, [9](#)
  - TOGGLE\_BIT, [10](#)
- bool\_t
  - Standard types, [15](#)
- BSRRx
  - GPIOx\_MemoryMapType, [41](#)
- BYBASED
  - MRCC\_private.h, [138](#)
- CFGR
  - RCC\_MemoryMapType, [45](#)
- CIR
  - RCC\_MemoryMapType, [45](#)
- CLR\_BIT
  - Bit Manipulation Math Macros, [9](#)
- CLR\_BITs
  - Bit Group Manipulation Math Macros, [11](#)
- Compiler standard macros, [12](#)
  - CONST, [14](#)
  - CONSTP2CONST, [13](#)
  - CONSTP2VAR, [13](#)
  - FUNC, [12](#)
  - P2CONST, [13](#)
  - P2FUNC, [14](#)
  - P2VAR, [13](#)
  - STATIC, [14](#)
  - VAR, [12](#)
- CONST
  - Compiler standard macros, [14](#)
- CONSTP2CONST
  - Compiler standard macros, [13](#)
- CONSTP2VAR
  - Compiler standard macros, [13](#)
- CR
  - RCC\_MemoryMapType, [45](#)
- CRCCLK
  - MRCC\_config.h, [88](#)
- CSR
  - RCC\_MemoryMapType, [50](#)
- CSS
  - MRCC\_config.h, [85](#)
- DCKCFGR
  - RCC\_MemoryMapType, [50](#)
- DISABLE
  - MRCC\_private.h, [138](#)
- DivisionBy2
  - MRCC\_private.h, [139](#)
- DivisionBy3
  - MRCC\_private.h, [139](#)
- DivisionBy4
  - MRCC\_private.h, [139](#)
- DivisionBy5
  - MRCC\_private.h, [139](#)
- DMA1CLK
  - MRCC\_config.h, [88](#)
- DMA2CLK
  - MRCC\_config.h, [88](#)
- ENABLE
  - MRCC\_private.h, [137](#)
- Equal\_0
  - MRCC\_private.h, [148](#)
- Equal\_1
  - MRCC\_private.h, [148](#)
- Equal\_10
  - MRCC\_private.h, [149](#)
- Equal\_11
  - MRCC\_private.h, [149](#)
- Equal\_12
  - MRCC\_private.h, [150](#)
- Equal\_13
  - MRCC\_private.h, [150](#)
- Equal\_14
  - MRCC\_private.h, [150](#)
- Equal\_15
  - MRCC\_private.h, [150](#)
- Equal\_16
  - MRCC\_private.h, [150](#)
- Equal\_17
  - MRCC\_private.h, [150](#)
- Equal\_18
  - MRCC\_private.h, [151](#)
- Equal\_19
  - MRCC\_private.h, [151](#)
- Equal\_2
  - MRCC\_private.h, [148](#)
- Equal\_20
  - MRCC\_private.h, [151](#)
- Equal\_21
  - MRCC\_private.h, [151](#)
- Equal\_22
  - MRCC\_private.h, [151](#)
- Equal\_23
  - MRCC\_private.h, [151](#)
- Equal\_24
  - MRCC\_private.h, [152](#)

- Equal\_25
  - MRCC\_private.h, [152](#)
- Equal\_26
  - MRCC\_private.h, [152](#)
- Equal\_27
  - MRCC\_private.h, [152](#)
- Equal\_28
  - MRCC\_private.h, [152](#)
- Equal\_29
  - MRCC\_private.h, [152](#)
- Equal\_3
  - MRCC\_private.h, [148](#)
- Equal\_30
  - MRCC\_private.h, [153](#)
- Equal\_31
  - MRCC\_private.h, [153](#)
- Equal\_32
  - MRCC\_private.h, [153](#)
- Equal\_33
  - MRCC\_private.h, [153](#)
- Equal\_34
  - MRCC\_private.h, [153](#)
- Equal\_35
  - MRCC\_private.h, [153](#)
- Equal\_36
  - MRCC\_private.h, [154](#)
- Equal\_37
  - MRCC\_private.h, [154](#)
- Equal\_38
  - MRCC\_private.h, [154](#)
- Equal\_39
  - MRCC\_private.h, [154](#)
- Equal\_4
  - MRCC\_private.h, [148](#)
- Equal\_40
  - MRCC\_private.h, [154](#)
- Equal\_41
  - MRCC\_private.h, [154](#)
- Equal\_42
  - MRCC\_private.h, [155](#)
- Equal\_43
  - MRCC\_private.h, [155](#)
- Equal\_44
  - MRCC\_private.h, [155](#)
- Equal\_45
  - MRCC\_private.h, [155](#)
- Equal\_46
  - MRCC\_private.h, [155](#)
- Equal\_47
  - MRCC\_private.h, [155](#)
- Equal\_48
  - MRCC\_private.h, [156](#)
- Equal\_49
  - MRCC\_private.h, [156](#)
- Equal\_5
  - MRCC\_private.h, [148](#)
- Equal\_50
  - MRCC\_private.h, [156](#)
- Equal\_51
  - MRCC\_private.h, [156](#)
- Equal\_52
  - MRCC\_private.h, [156](#)
- Equal\_53
  - MRCC\_private.h, [156](#)
- Equal\_54
  - MRCC\_private.h, [157](#)
- Equal\_55
  - MRCC\_private.h, [157](#)
- Equal\_56
  - MRCC\_private.h, [157](#)
- Equal\_57
  - MRCC\_private.h, [157](#)
- Equal\_58
  - MRCC\_private.h, [157](#)
- Equal\_59
  - MRCC\_private.h, [157](#)
- Equal\_6
  - MRCC\_private.h, [149](#)
- Equal\_60
  - MRCC\_private.h, [158](#)
- Equal\_61
  - MRCC\_private.h, [158](#)
- Equal\_62
  - MRCC\_private.h, [158](#)
- Equal\_63
  - MRCC\_private.h, [158](#)
- Equal\_7
  - MRCC\_private.h, [149](#)
- Equal\_8
  - MRCC\_private.h, [149](#)
- Equal\_9
  - MRCC\_private.h, [149](#)
- FALSE
  - Standard values, [18](#)
- fl32\_t
  - Standard types, [17](#)
- fl64\_t
  - Standard types, [17](#)
- FLAG\_CLEARED
  - Standard values, [19](#)
- FLAG\_SET
  - Standard values, [19](#)
- FUNC
  - Compiler standard macros, [12](#)
- GET\_BIT
  - Bit Manipulation Math Macros, [10](#)
- GET\_BITs
  - Bit Group Manipulation Math Macros, [11](#)
- GPIO Addresses, [35](#)
  - GPIOA\_BASE\_ADDRESS, [35](#)
  - GPIOB\_BASE\_ADDRESS, [35](#)
  - GPIOC\_BASE\_ADDRESS, [36](#)
- GPIO Alternate Functions, [30](#)
  - GPIOx\_AF0, [31](#)
  - GPIOx\_AF1, [31](#)

- GPIOx\_AF10, [33](#)
- GPIOx\_AF11, [34](#)
- GPIOx\_AF12, [34](#)
- GPIOx\_AF13, [34](#)
- GPIOx\_AF14, [34](#)
- GPIOx\_AF15, [35](#)
- GPIOx\_AF2, [31](#)
- GPIOx\_AF3, [32](#)
- GPIOx\_AF4, [32](#)
- GPIOx\_AF5, [32](#)
- GPIOx\_AF6, [32](#)
- GPIOx\_AF7, [33](#)
- GPIOx\_AF8, [33](#)
- GPIOx\_AF9, [33](#)
- GPIO Modes, [20](#)
  - GPIOx\_MODE\_AF, [20](#)
  - GPIOx\_MODE\_ANALOG, [20](#)
  - GPIOx\_MODE\_INPUT, [20](#)
  - GPIOx\_MODE\_OUTPUT, [20](#)
- GPIO Output PINs, [26](#)
  - GPIOx\_PIN0, [26](#)
  - GPIOx\_PIN1, [26](#)
  - GPIOx\_PIN10, [29](#)
  - GPIOx\_PIN11, [29](#)
  - GPIOx\_PIN12, [29](#)
  - GPIOx\_PIN13, [29](#)
  - GPIOx\_PIN14, [30](#)
  - GPIOx\_PIN15, [30](#)
  - GPIOx\_PIN2, [27](#)
  - GPIOx\_PIN3, [27](#)
  - GPIOx\_PIN4, [27](#)
  - GPIOx\_PIN5, [27](#)
  - GPIOx\_PIN6, [28](#)
  - GPIOx\_PIN7, [28](#)
  - GPIOx\_PIN8, [28](#)
  - GPIOx\_PIN9, [28](#)
- GPIO Output Types, [22](#)
  - GPIOx\_OPENDRAIN, [22](#)
  - GPIOx\_PUSH\_PULL, [22](#)
- GPIO Output Values, [25](#)
  - GPIOx\_HIGH, [25](#)
  - GPIOx\_LOW, [25](#)
- GPIO PIN Speed, [23](#)
  - GPIOx\_HighSpeed, [23](#)
  - GPIOx\_LowSpeed, [23](#)
  - GPIOx\_MediumSpeed, [23](#)
  - GPIOx\_VeryHighSpeed, [23](#)
- GPIO Ports, [21](#)
  - GPIO\_PORTA, [21](#)
  - GPIO\_PORTB, [21](#)
  - GPIO\_PORTC, [21](#)
- GPIO Pull Types, [24](#)
  - GPIOx\_NoPull, [24](#)
  - GPIOx\_PullDown, [24](#)
  - GPIOx\_PullUp, [24](#)
- GPIO Registers, [36](#)
  - GPIOA, [36](#)
  - GPIOB, [36](#)
- GPIOC, [37](#)
- GPIO\_config.h
  - GPIOA\_PIN\_POS, [57](#)
  - GPIOB\_PIN\_POS, [57](#)
  - LCKK\_BIT\_POS, [57](#)
- GPIO\_interface.h
  - MGPIOx\_u8GetPinValue, [64](#)
  - MGPIOx\_vInit, [65](#)
  - MGPIOx\_vLockedPins, [61](#)
  - MGPIOx\_vSetAlternateFunctionON, [65](#)
  - MGPIOx\_vSetPinInputPullType, [62](#)
  - MGPIOx\_vSetPinMode, [61](#)
  - MGPIOx\_vSetPinOutputSpeed, [62](#)
  - MGPIOx\_vSetPinOutputType, [62](#)
  - MGPIOx\_vSetPinValue, [64](#)
  - MGPIOx\_vSetPortConfigLock, [65](#)
  - MGPIOx\_vSetResetAtomic, [64](#)
  - MGPIOx\_vTogglePinValue, [66](#)
- GPIO\_PORTA
  - GPIO Ports, [21](#)
- GPIO\_PORTB
  - GPIO Ports, [21](#)
- GPIO\_PORTC
  - GPIO Ports, [21](#)
- GPIO\_program.c
  - MGPIOx\_u8GetPinValue, [73](#)
  - MGPIOx\_vInit, [76](#)
  - MGPIOx\_vLockedPins, [70](#)
  - MGPIOx\_vSetAlternateFunctionON, [75](#)
  - MGPIOx\_vSetPinInputPullType, [72](#)
  - MGPIOx\_vSetPinMode, [71](#)
  - MGPIOx\_vSetPinOutputSpeed, [72](#)
  - MGPIOx\_vSetPinOutputType, [71](#)
  - MGPIOx\_vSetPinValue, [73](#)
  - MGPIOx\_vSetResetAtomic, [74](#)
  - MGPIOx\_vTogglePinValue, [77](#)
- GPIOA
  - GPIO Registers, [36](#)
- GPIOA\_BASE\_ADDRESS
  - GPIO Addresses, [35](#)
- GPIOA\_PIN\_POS
  - GPIO\_config.h, [57](#)
- GPIOCLK
  - MRCC\_config.h, [89](#)
- GPIOB
  - GPIO Registers, [36](#)
- GPIOB\_BASE\_ADDRESS
  - GPIO Addresses, [35](#)
- GPIOB\_PIN\_POS
  - GPIO\_config.h, [57](#)
- GPIOBCLK
  - MRCC\_config.h, [89](#)
- GPIOC
  - GPIO Registers, [37](#)
- GPIOC\_BASE\_ADDRESS
  - GPIO Addresses, [36](#)
- GPIOCCLK
  - MRCC\_config.h, [89](#)

- GPIODCLK
  - MRCC\_config.h, [89](#)
- GPIOECLK
  - MRCC\_config.h, [89](#)
- GPIOHCLK
  - MRCC\_config.h, [88](#)
- GPIOx\_AF0
  - GPIO Alternate Functions, [31](#)
- GPIOx\_AF1
  - GPIO Alternate Functions, [31](#)
- GPIOx\_AF10
  - GPIO Alternate Functions, [33](#)
- GPIOx\_AF11
  - GPIO Alternate Functions, [34](#)
- GPIOx\_AF12
  - GPIO Alternate Functions, [34](#)
- GPIOx\_AF13
  - GPIO Alternate Functions, [34](#)
- GPIOx\_AF14
  - GPIO Alternate Functions, [34](#)
- GPIOx\_AF15
  - GPIO Alternate Functions, [35](#)
- GPIOx\_AF2
  - GPIO Alternate Functions, [31](#)
- GPIOx\_AF3
  - GPIO Alternate Functions, [32](#)
- GPIOx\_AF4
  - GPIO Alternate Functions, [32](#)
- GPIOx\_AF5
  - GPIO Alternate Functions, [32](#)
- GPIOx\_AF6
  - GPIO Alternate Functions, [32](#)
- GPIOx\_AF7
  - GPIO Alternate Functions, [33](#)
- GPIOx\_AF8
  - GPIO Alternate Functions, [33](#)
- GPIOx\_AF9
  - GPIO Alternate Functions, [33](#)
- GPIOx\_HIGH
  - GPIO Output Values, [25](#)
- GPIOx\_HighSpeed
  - GPIO PIN Speed, [23](#)
- GPIOx\_LOW
  - GPIO Output Values, [25](#)
- GPIOx\_LowSpeed
  - GPIO PIN Speed, [23](#)
- GPIOx\_MediumSpeed
  - GPIO PIN Speed, [23](#)
- GPIOx\_MemoryMapType, [39](#)
  - AFRHx, [41](#)
  - AFRLx, [41](#)
  - BSRRx, [41](#)
  - IDRx, [40](#)
  - LCKRx, [41](#)
  - MODERx, [40](#)
  - ODRx, [40](#)
  - OSPEEDRx, [40](#)
  - OTYPERx, [40](#)
  - PUPDRx, [40](#)
- GPIOx\_MODE\_AF
  - GPIO Modes, [20](#)
- GPIOx\_MODE\_ANALOG
  - GPIO Modes, [20](#)
- GPIOx\_MODE\_INPUT
  - GPIO Modes, [20](#)
- GPIOx\_MODE\_OUTPUT
  - GPIO Modes, [20](#)
- GPIOx\_NoPull
  - GPIO Pull Types, [24](#)
- GPIOx\_OPENDRAIN
  - GPIO Output Types, [22](#)
- GPIOx\_PIN0
  - GPIO Output PINs, [26](#)
- GPIOx\_PIN1
  - GPIO Output PINs, [26](#)
- GPIOx\_PIN10
  - GPIO Output PINs, [29](#)
- GPIOx\_PIN11
  - GPIO Output PINs, [29](#)
- GPIOx\_PIN12
  - GPIO Output PINs, [29](#)
- GPIOx\_PIN13
  - GPIO Output PINs, [29](#)
- GPIOx\_PIN14
  - GPIO Output PINs, [30](#)
- GPIOx\_PIN15
  - GPIO Output PINs, [30](#)
- GPIOx\_PIN2
  - GPIO Output PINs, [27](#)
- GPIOx\_PIN3
  - GPIO Output PINs, [27](#)
- GPIOx\_PIN4
  - GPIO Output PINs, [27](#)
- GPIOx\_PIN5
  - GPIO Output PINs, [27](#)
- GPIOx\_PIN6
  - GPIO Output PINs, [28](#)
- GPIOx\_PIN7
  - GPIO Output PINs, [28](#)
- GPIOx\_PIN8
  - GPIO Output PINs, [28](#)
- GPIOx\_PIN9
  - GPIO Output PINs, [28](#)
- GPIOx\_PullDown
  - GPIO Pull Types, [24](#)
- GPIOx\_PullUp
  - GPIO Pull Types, [24](#)
- GPIOx\_PUSH\_PULL
  - GPIO Output Types, [22](#)
- GPIOx\_VeryHighSpeed
  - GPIO PIN Speed, [23](#)
- HPRE
  - MRCC\_config.h, [87](#)
- HSE
  - MRCC\_private.h, [138](#)
- HSE\_EN

- MRCC\_config.h, [85](#)
- HSEby10
  - MRCC\_private.h, [143](#)
- HSEby11
  - MRCC\_private.h, [143](#)
- HSEby12
  - MRCC\_private.h, [144](#)
- HSEby13
  - MRCC\_private.h, [144](#)
- HSEby14
  - MRCC\_private.h, [144](#)
- HSEby15
  - MRCC\_private.h, [144](#)
- HSEby16
  - MRCC\_private.h, [144](#)
- HSEby17
  - MRCC\_private.h, [144](#)
- HSEby18
  - MRCC\_private.h, [145](#)
- HSEby19
  - MRCC\_private.h, [145](#)
- HSEby2
  - MRCC\_private.h, [142](#)
- HSEby20
  - MRCC\_private.h, [145](#)
- HSEby21
  - MRCC\_private.h, [145](#)
- HSEby22
  - MRCC\_private.h, [145](#)
- HSEby23
  - MRCC\_private.h, [145](#)
- HSEby24
  - MRCC\_private.h, [146](#)
- HSEby25
  - MRCC\_private.h, [146](#)
- HSEby26
  - MRCC\_private.h, [146](#)
- HSEby27
  - MRCC\_private.h, [146](#)
- HSEby28
  - MRCC\_private.h, [146](#)
- HSEby29
  - MRCC\_private.h, [146](#)
- HSEby3
  - MRCC\_private.h, [142](#)
- HSEby30
  - MRCC\_private.h, [147](#)
- HSEby31
  - MRCC\_private.h, [147](#)
- HSEby4
  - MRCC\_private.h, [142](#)
- HSEby5
  - MRCC\_private.h, [142](#)
- HSEby6
  - MRCC\_private.h, [143](#)
- HSEby7
  - MRCC\_private.h, [143](#)
- HSEby8
  - MRCC\_private.h, [143](#)
- HSEby9
  - MRCC\_private.h, [143](#)
- HSEBYP
  - MRCC\_config.h, [85](#)
- HSI
  - MRCC\_private.h, [140](#)
- HSI\_EN
  - MRCC\_config.h, [85](#)
- I2C1EN
  - MRCC\_config.h, [90](#)
- I2C2EN
  - MRCC\_config.h, [90](#)
- I2C3EN
  - MRCC\_config.h, [90](#)
- I2S\_CKIN
  - MRCC\_private.h, [140](#)
- I2SSRC
  - MRCC\_config.h, [87](#)
- IDRx
  - GPIOx\_MemoryMapType, [40](#)
- INITIAL\_ZERO
  - Standard values, [19](#)
- InputType
  - MGPIOx\_ConfigType, [43](#)
- LCKK\_BIT\_POS
  - GPIO\_config.h, [57](#)
- LCKRx
  - GPIOx\_MemoryMapType, [41](#)
- LSE
  - MRCC\_private.h, [140](#)
- MCO1
  - MRCC\_config.h, [87](#)
- MCO1PRE
  - MRCC\_config.h, [86](#)
- MCO2
  - MRCC\_config.h, [86](#)
- MCO2PRE
  - MRCC\_config.h, [86](#)
- MGPIOx\_ConfigType, [42](#)
  - AF\_Type, [44](#)
  - InputType, [43](#)
  - Mode, [43](#)
  - OutputSpeed, [43](#)
  - OutputType, [43](#)
  - Pin, [42](#)
  - Port, [42](#)
- MGPIOx\_u8GetPinValue
  - GPIO\_interface.h, [64](#)
  - GPIO\_program.c, [73](#)
- MGPIOx\_vInit
  - GPIO\_interface.h, [65](#)
  - GPIO\_program.c, [76](#)
- MGPIOx\_vLockedPins
  - GPIO\_interface.h, [61](#)
  - GPIO\_program.c, [70](#)



MGPIOR\_vSetAlternateFunctionON  
     GPIO\_interface.h, 65  
     GPIO\_program.c, 75  
 MGPIOR\_vSetPinInputPullType  
     GPIO\_interface.h, 62  
     GPIO\_program.c, 72  
 MGPIOR\_vSetPinMode  
     GPIO\_interface.h, 61  
     GPIO\_program.c, 71  
 MGPIOR\_vSetPinOutputSpeed  
     GPIO\_interface.h, 62  
     GPIO\_program.c, 72  
 MGPIOR\_vSetPinOutputType  
     GPIO\_interface.h, 62  
     GPIO\_program.c, 71  
 MGPIOR\_vSetPinValue  
     GPIO\_interface.h, 64  
     GPIO\_program.c, 73  
 MGPIOR\_vSetPortConfigLock  
     GPIO\_interface.h, 65  
 MGPIOR\_vSetResetAtomic  
     GPIO\_interface.h, 64  
     GPIO\_program.c, 74  
 MGPIOR\_vTogglePinValue  
     GPIO\_interface.h, 66  
     GPIO\_program.c, 77  
 Mode  
     MGPIOR\_ConfigType, 43  
 MODERx  
     GPIOx\_MemoryMapType, 40  
 MRCC\_config.h  
     ADC1EN, 93  
     CRCCLK, 88  
     CSS, 85  
     DMA1CLK, 88  
     DMA2CLK, 88  
     GPIOACLK, 89  
     GPIOBCLK, 89  
     GPIOCCLK, 89  
     GPIODCLK, 89  
     GPIOECLK, 89  
     GPIOHCLK, 88  
     HPRE, 87  
     HSE\_EN, 85  
     HSEBYP, 85  
     HSI\_EN, 85  
     I2C1EN, 90  
     I2C2EN, 90  
     I2C3EN, 90  
     I2SSRC, 87  
     MCO1, 87  
     MCO1PRE, 86  
     MCO2, 86  
     MCO2PRE, 86  
     OTGFS, 89  
     PLL, 84  
     PLLI2S, 84  
     PLLM, 86  
     PLLN, 86  
     PLL, 86  
     PLLQ, 85  
     PLLSRC, 85  
     PPRE1, 87  
     PPRE2, 87  
     PWREN, 90  
     RTCPRE, 87  
     SDIOEN, 93  
     SPI1EN, 92  
     SPI2EN, 91  
     SPI3EN, 90  
     SPI4EN, 92  
     SW, 88  
     SWS, 88  
     SYSCFGEN, 92  
     TIM10EN, 92  
     TIM11EN, 92  
     TIM1EN, 93  
     TIM2EN, 91  
     TIM3EN, 91  
     TIM4EN, 91  
     TIM5EN, 91  
     TIM9EN, 92  
     USART1EN, 93  
     USART2EN, 90  
     USART6EN, 93  
     WWDGEN, 91  
 MRCC\_interface.h  
     AHB1ENR\_CRCEN, 104  
     AHB1ENR\_DMA1EN, 104  
     AHB1ENR\_DMA2EN, 104  
     AHB1ENR\_GPIOAEN, 105  
     AHB1ENR\_GPIOBEN, 105  
     AHB1ENR\_GPIOCEN, 105  
     AHB1ENR\_GPIODEN, 105  
     AHB1ENR\_GPIOEEN, 104  
     AHB1ENR\_GPIOHEN, 104  
     AHB1LPENR\_FLITFLPEN, 109  
     AHB2ENR\_OTGFSEN, 105  
     APB1ENR\_I2C1EN, 106  
     APB1ENR\_I2C2EN, 106  
     APB1ENR\_I2C3EN, 106  
     APB1ENR\_PWREN, 105  
     APB1ENR\_SPI2EN, 106  
     APB1ENR\_SPI3EN, 106  
     APB1ENR\_TIM2EN, 107  
     APB1ENR\_TIM3EN, 107  
     APB1ENR\_TIM4EN, 107  
     APB1ENR\_TIM5EN, 107  
     APB1ENR\_USART2EN, 106  
     APB1ENR\_WWDGEN, 107  
     APB2ENR\_ADC1EN, 109  
     APB2ENR\_SDIOEN, 108  
     APB2ENR\_SPI1EN, 108  
     APB2ENR\_SPI4EN, 108  
     APB2ENR\_SYSCFGEN, 108  
     APB2ENR\_TIM10EN, 108

APB2ENR\_TIM11EN, [107](#)  
 APB2ENR\_TIM1EN, [109](#)  
 APB2ENR\_TIM9EN, [108](#)  
 APB2ENR\_USART1EN, [109](#)  
 APB2ENR\_USART6EN, [109](#)  
 MRCC\_vDisablePeripheralCLK, [115](#)  
 MRCC\_vEnablePeripheralCLK, [114](#)  
 MRCC\_vInit, [110](#)  
 RCC\_AHB1, [103](#)  
 RCC\_AHB1LPENR, [104](#)  
 RCC\_AHB2, [103](#)  
 RCC\_APB1, [103](#)  
 RCC\_APB2, [103](#)  
 MRCC\_private.h  
   AHBby16, [147](#)  
   AHBby2, [147](#)  
   AHBby4, [147](#)  
   AHBby8, [147](#)  
   BYBASED, [138](#)  
   DISABLE, [138](#)  
   DivisionBy2, [139](#)  
   DivisionBy3, [139](#)  
   DivisionBy4, [139](#)  
   DivisionBy5, [139](#)  
   ENABLE, [137](#)  
   Equal\_0, [148](#)  
   Equal\_1, [148](#)  
   Equal\_10, [149](#)  
   Equal\_11, [149](#)  
   Equal\_12, [150](#)  
   Equal\_13, [150](#)  
   Equal\_14, [150](#)  
   Equal\_15, [150](#)  
   Equal\_16, [150](#)  
   Equal\_17, [150](#)  
   Equal\_18, [151](#)  
   Equal\_19, [151](#)  
   Equal\_2, [148](#)  
   Equal\_20, [151](#)  
   Equal\_21, [151](#)  
   Equal\_22, [151](#)  
   Equal\_23, [151](#)  
   Equal\_24, [152](#)  
   Equal\_25, [152](#)  
   Equal\_26, [152](#)  
   Equal\_27, [152](#)  
   Equal\_28, [152](#)  
   Equal\_29, [152](#)  
   Equal\_3, [148](#)  
   Equal\_30, [153](#)  
   Equal\_31, [153](#)  
   Equal\_32, [153](#)  
   Equal\_33, [153](#)  
   Equal\_34, [153](#)  
   Equal\_35, [153](#)  
   Equal\_36, [154](#)  
   Equal\_37, [154](#)  
   Equal\_38, [154](#)  
   Equal\_39, [154](#)  
   Equal\_4, [148](#)  
   Equal\_40, [154](#)  
   Equal\_41, [154](#)  
   Equal\_42, [155](#)  
   Equal\_43, [155](#)  
   Equal\_44, [155](#)  
   Equal\_45, [155](#)  
   Equal\_46, [155](#)  
   Equal\_47, [155](#)  
   Equal\_48, [156](#)  
   Equal\_49, [156](#)  
   Equal\_5, [148](#)  
   Equal\_50, [156](#)  
   Equal\_51, [156](#)  
   Equal\_52, [156](#)  
   Equal\_53, [156](#)  
   Equal\_54, [157](#)  
   Equal\_55, [157](#)  
   Equal\_56, [157](#)  
   Equal\_57, [157](#)  
   Equal\_58, [157](#)  
   Equal\_59, [157](#)  
   Equal\_6, [149](#)  
   Equal\_60, [158](#)  
   Equal\_61, [158](#)  
   Equal\_62, [158](#)  
   Equal\_63, [158](#)  
   Equal\_7, [149](#)  
   Equal\_8, [149](#)  
   Equal\_9, [149](#)  
   HSE, [138](#)  
   HSEby10, [143](#)  
   HSEby11, [143](#)  
   HSEby12, [144](#)  
   HSEby13, [144](#)  
   HSEby14, [144](#)  
   HSEby15, [144](#)  
   HSEby16, [144](#)  
   HSEby17, [144](#)  
   HSEby18, [145](#)  
   HSEby19, [145](#)  
   HSEby2, [142](#)  
   HSEby20, [145](#)  
   HSEby21, [145](#)  
   HSEby22, [145](#)  
   HSEby23, [145](#)  
   HSEby24, [146](#)  
   HSEby25, [146](#)  
   HSEby26, [146](#)  
   HSEby27, [146](#)  
   HSEby28, [146](#)  
   HSEby29, [146](#)  
   HSEby3, [142](#)  
   HSEby30, [147](#)  
   HSEby31, [147](#)  
   HSEby4, [142](#)  
   HSEby5, [142](#)

HSEby6, [143](#)  
HSEby7, [143](#)  
HSEby8, [143](#)  
HSEby9, [143](#)  
HSI, [140](#)  
I2S\_CKIN, [140](#)  
LSE, [140](#)  
NoCLK0, [142](#)  
NoCLK1, [142](#)  
NoDivision, [139](#)  
NOTBYBASED, [138](#)  
PLLCLK, [139](#), [140](#)  
PLLI2SCLK, [138](#)  
RCC, [121](#)  
RCC\_AHB1ENR\_CRCEN, [132](#)  
RCC\_AHB1ENR\_DMA1EN, [132](#)  
RCC\_AHB1ENR\_DMA2EN, [132](#)  
RCC\_AHB1ENR\_GPIOAEN, [133](#)  
RCC\_AHB1ENR\_GPIOBEN, [133](#)  
RCC\_AHB1ENR\_GPIOCEN, [133](#)  
RCC\_AHB1ENR\_GPIODEN, [133](#)  
RCC\_AHB1ENR\_GPIOEEN, [132](#)  
RCC\_AHB1ENR\_GPIOHEN, [132](#)  
RCC\_AHB1LPENR\_FLITFLPEN, [137](#)  
RCC\_AHB2ENR\_OTGFSEN, [133](#)  
RCC\_APB1ENR\_I2C1EN, [134](#)  
RCC\_APB1ENR\_I2C2EN, [134](#)  
RCC\_APB1ENR\_I2C3EN, [134](#)  
RCC\_APB1ENR\_PWREN, [133](#)  
RCC\_APB1ENR\_SPI2EN, [134](#)  
RCC\_APB1ENR\_SPI3EN, [134](#)  
RCC\_APB1ENR\_TIM2EN, [135](#)  
RCC\_APB1ENR\_TIM3EN, [135](#)  
RCC\_APB1ENR\_TIM4EN, [135](#)  
RCC\_APB1ENR\_TIM5EN, [135](#)  
RCC\_APB1ENR\_USART2EN, [134](#)  
RCC\_APB1ENR\_WWDGEN, [135](#)  
RCC\_APB2ENR\_ADC1EN, [137](#)  
RCC\_APB2ENR\_SDIOEN, [136](#)  
RCC\_APB2ENR\_SPI1EN, [136](#)  
RCC\_APB2ENR\_SPI4EN, [136](#)  
RCC\_APB2ENR\_SYSCFGEN, [136](#)  
RCC\_APB2ENR\_TIM10EN, [136](#)  
RCC\_APB2ENR\_TIM11EN, [135](#)  
RCC\_APB2ENR\_TIM1EN, [137](#)  
RCC\_APB2ENR\_TIM9EN, [136](#)  
RCC\_APB2ENR\_USART1EN, [137](#)  
RCC\_APB2ENR\_USART6EN, [137](#)  
RCC\_BASE\_ADDRESS, [121](#)  
RCC\_CFGR\_HPRE\_b0, [130](#)  
RCC\_CFGR\_HPRE\_b1, [131](#)  
RCC\_CFGR\_HPRE\_b2, [131](#)  
RCC\_CFGR\_HPRE\_b3, [131](#)  
RCC\_CFGR\_I2SSRC, [128](#)  
RCC\_CFGR\_MOC1\_b0, [128](#)  
RCC\_CFGR\_MOC1\_b1, [128](#)  
RCC\_CFGR\_MOC1PRE\_b0, [128](#)  
RCC\_CFGR\_MOC1PRE\_b1, [128](#)  
RCC\_CFGR\_MOC1PRE\_b2, [128](#)  
RCC\_CFGR\_MOC2\_b0, [127](#)  
RCC\_CFGR\_MOC2\_b1, [127](#)  
RCC\_CFGR\_MOC2PRE\_b0, [127](#)  
RCC\_CFGR\_MOC2PRE\_b1, [127](#)  
RCC\_CFGR\_MOC2PRE\_b2, [127](#)  
RCC\_CFGR\_PPRE1\_b0, [130](#)  
RCC\_CFGR\_PPRE1\_b1, [130](#)  
RCC\_CFGR\_PPRE1\_b2, [130](#)  
RCC\_CFGR\_PPRE2\_b0, [129](#)  
RCC\_CFGR\_PPRE2\_b1, [130](#)  
RCC\_CFGR\_PPRE2\_b2, [130](#)  
RCC\_CFGR\_RTCPRE\_b0, [129](#)  
RCC\_CFGR\_RTCPRE\_b1, [129](#)  
RCC\_CFGR\_RTCPRE\_b2, [129](#)  
RCC\_CFGR\_RTCPRE\_b3, [129](#)  
RCC\_CFGR\_RTCPRE\_b4, [129](#)  
RCC\_CFGR\_SW\_b0, [131](#)  
RCC\_CFGR\_SW\_b1, [132](#)  
RCC\_CFGR\_SWS\_b0, [131](#)  
RCC\_CFGR\_SWS\_b1, [131](#)  
RCC\_CR\_CSSON, [122](#)  
RCC\_CR\_HSEBYP, [122](#)  
RCC\_CR\_HSEON, [123](#)  
RCC\_CR\_HSERDY, [122](#)  
RCC\_CR\_HSION, [123](#)  
RCC\_CR\_HSIRDY, [123](#)  
RCC\_CR\_PLLI2SON, [122](#)  
RCC\_CR\_PLLI2SRDY, [121](#)  
RCC\_CR\_PLLON, [122](#)  
RCC\_CR\_PLLRDY, [122](#)  
RCC\_PLLCFGR\_PLLM\_b0, [126](#)  
RCC\_PLLCFGR\_PLLM\_b1, [126](#)  
RCC\_PLLCFGR\_PLLM\_b2, [126](#)  
RCC\_PLLCFGR\_PLLM\_b3, [126](#)  
RCC\_PLLCFGR\_PLLM\_b4, [126](#)  
RCC\_PLLCFGR\_PLLM\_b5, [127](#)  
RCC\_PLLCFGR\_PLLN\_b0, [124](#)  
RCC\_PLLCFGR\_PLLN\_b1, [124](#)  
RCC\_PLLCFGR\_PLLN\_b2, [125](#)  
RCC\_PLLCFGR\_PLLN\_b3, [125](#)  
RCC\_PLLCFGR\_PLLN\_b4, [125](#)  
RCC\_PLLCFGR\_PLLN\_b5, [125](#)  
RCC\_PLLCFGR\_PLLN\_b6, [125](#)  
RCC\_PLLCFGR\_PLLN\_b7, [125](#)  
RCC\_PLLCFGR\_PLLN\_b8, [126](#)  
RCC\_PLLCFGR\_PLLP\_b0, [124](#)  
RCC\_PLLCFGR\_PLLP\_b1, [124](#)  
RCC\_PLLCFGR\_PLLQ\_b0, [123](#)  
RCC\_PLLCFGR\_PLLQ\_b1, [123](#)  
RCC\_PLLCFGR\_PLLQ\_b2, [123](#)  
RCC\_PLLCFGR\_PLLQ\_b3, [124](#)  
RCC\_PLLCFGR\_PLLSRC, [124](#)  
SYSCLK, [138](#)  
SYSCLKby128, [141](#)  
SYSCLKby16, [141](#)  
SYSCLKby2, [140](#)  
SYSCLKby256, [141](#)

- SYSCLKby4, [140](#)
  - SYSCLKby512, [141](#)
  - SYSCLKby64, [141](#)
  - SYSCLKby8, [141](#)
- MRCC\_program.c
  - MRCC\_vDisablePeripheralCLK, [168](#)
  - MRCC\_vEnablePeripheralCLK, [168](#)
  - MRCC\_vInit, [163](#)
- MRCC\_vDisablePeripheralCLK
  - MRCC\_interface.h, [115](#)
  - MRCC\_program.c, [168](#)
- MRCC\_vEnablePeripheralCLK
  - MRCC\_interface.h, [114](#)
  - MRCC\_program.c, [168](#)
- MRCC\_vInit
  - MRCC\_interface.h, [110](#)
  - MRCC\_program.c, [163](#)
- MY\_MS\_DELAY
  - Utilities macros, [15](#)
- NoCLK0
  - MRCC\_private.h, [142](#)
- NoCLK1
  - MRCC\_private.h, [142](#)
- NoDivision
  - MRCC\_private.h, [139](#)
- NOTBYBASED
  - MRCC\_private.h, [138](#)
- NULL
  - Standard values, [18](#)
- ODRx
  - GPIOx\_MemoryMapType, [40](#)
- OSPEEDRx
  - GPIOx\_MemoryMapType, [40](#)
- OTGFS
  - MRCC\_config.h, [89](#)
- OTYPERx
  - GPIOx\_MemoryMapType, [40](#)
- OutputSpeed
  - MGPIOx\_ConfigType, [43](#)
- OutputType
  - MGPIOx\_ConfigType, [43](#)
- P2CONST
  - Compiler standard macros, [13](#)
- P2FUNC
  - Compiler standard macros, [14](#)
- P2VAR
  - Compiler standard macros, [13](#)
- Pin
  - MGPIOx\_ConfigType, [42](#)
- PLL
  - MRCC\_config.h, [84](#)
- PLLCFGR
  - RCC\_MemoryMapType, [45](#)
- PLLCLK
  - MRCC\_private.h, [139](#), [140](#)
- PLLI2S
  - MRCC\_config.h, [84](#)
- PLLI2SCFGR
  - RCC\_MemoryMapType, [50](#)
- PLLI2SCLK
  - MRCC\_private.h, [138](#)
- PLLM
  - MRCC\_config.h, [86](#)
- PLLN
  - MRCC\_config.h, [86](#)
- PLLQ
  - MRCC\_config.h, [86](#)
- PLLQ
  - MRCC\_config.h, [85](#)
- PLLSRC
  - MRCC\_config.h, [85](#)
- Port
  - MGPIOx\_ConfigType, [42](#)
- PPRE1
  - MRCC\_config.h, [87](#)
- PPRE2
  - MRCC\_config.h, [87](#)
- PUPDRx
  - GPIOx\_MemoryMapType, [40](#)
- PWREN
  - MRCC\_config.h, [90](#)
- RCC
  - MRCC\_private.h, [121](#)
- RCC\_AHB1
  - MRCC\_interface.h, [103](#)
- RCC\_AHB1ENR\_CRCEN
  - MRCC\_private.h, [132](#)
- RCC\_AHB1ENR\_DMA1EN
  - MRCC\_private.h, [132](#)
- RCC\_AHB1ENR\_DMA2EN
  - MRCC\_private.h, [132](#)
- RCC\_AHB1ENR\_GPIOAEN
  - MRCC\_private.h, [133](#)
- RCC\_AHB1ENR\_GPIOBEN
  - MRCC\_private.h, [133](#)
- RCC\_AHB1ENR\_GPIOCEN
  - MRCC\_private.h, [133](#)
- RCC\_AHB1ENR\_GPIODEN
  - MRCC\_private.h, [133](#)
- RCC\_AHB1ENR\_GPIOEEN
  - MRCC\_private.h, [132](#)
- RCC\_AHB1ENR\_GPIOHEN
  - MRCC\_private.h, [132](#)
- RCC\_AHB1LPENR
  - MRCC\_interface.h, [104](#)
- RCC\_AHB1LPENR\_FLITFLPEN
  - MRCC\_private.h, [137](#)
- RCC\_AHB2
  - MRCC\_interface.h, [103](#)
- RCC\_AHB2ENR\_OTGFSEN
  - MRCC\_private.h, [133](#)
- RCC\_APB1
  - MRCC\_interface.h, [103](#)
- RCC\_APB1ENR\_I2C1EN

MRCC\_private.h, 134  
RCC\_APB1ENR\_I2C2EN  
MRCC\_private.h, 134  
RCC\_APB1ENR\_I2C3EN  
MRCC\_private.h, 134  
RCC\_APB1ENR\_PWREN  
MRCC\_private.h, 133  
RCC\_APB1ENR\_SPI2EN  
MRCC\_private.h, 134  
RCC\_APB1ENR\_SPI3EN  
MRCC\_private.h, 134  
RCC\_APB1ENR\_TIM2EN  
MRCC\_private.h, 135  
RCC\_APB1ENR\_TIM3EN  
MRCC\_private.h, 135  
RCC\_APB1ENR\_TIM4EN  
MRCC\_private.h, 135  
RCC\_APB1ENR\_TIM5EN  
MRCC\_private.h, 135  
RCC\_APB1ENR\_USART2EN  
MRCC\_private.h, 134  
RCC\_APB1ENR\_WWDGEN  
MRCC\_private.h, 135  
RCC\_APB2  
MRCC\_interface.h, 103  
RCC\_APB2ENR\_ADC1EN  
MRCC\_private.h, 137  
RCC\_APB2ENR\_SDI0EN  
MRCC\_private.h, 136  
RCC\_APB2ENR\_SPI1EN  
MRCC\_private.h, 136  
RCC\_APB2ENR\_SPI4EN  
MRCC\_private.h, 136  
RCC\_APB2ENR\_SYSCFGEN  
MRCC\_private.h, 136  
RCC\_APB2ENR\_TIM10EN  
MRCC\_private.h, 136  
RCC\_APB2ENR\_TIM11EN  
MRCC\_private.h, 135  
RCC\_APB2ENR\_TIM1EN  
MRCC\_private.h, 137  
RCC\_APB2ENR\_TIM9EN  
MRCC\_private.h, 136  
RCC\_APB2ENR\_USART1EN  
MRCC\_private.h, 137  
RCC\_APB2ENR\_USART6EN  
MRCC\_private.h, 137  
RCC\_BASE\_ADDRESS  
MRCC\_private.h, 121  
RCC\_CFGR\_HPRE\_b0  
MRCC\_private.h, 130  
RCC\_CFGR\_HPRE\_b1  
MRCC\_private.h, 131  
RCC\_CFGR\_HPRE\_b2  
MRCC\_private.h, 131  
RCC\_CFGR\_HPRE\_b3  
MRCC\_private.h, 131  
RCC\_CFGR\_I2SSRC  
MRCC\_private.h, 128  
RCC\_CFGR\_MOC1\_b0  
MRCC\_private.h, 128  
RCC\_CFGR\_MOC1\_b1  
MRCC\_private.h, 128  
RCC\_CFGR\_MOC1PRE\_b0  
MRCC\_private.h, 128  
RCC\_CFGR\_MOC1PRE\_b1  
MRCC\_private.h, 128  
RCC\_CFGR\_MOC1PRE\_b2  
MRCC\_private.h, 128  
RCC\_CFGR\_MOC2\_b0  
MRCC\_private.h, 127  
RCC\_CFGR\_MOC2\_b1  
MRCC\_private.h, 127  
RCC\_CFGR\_MOC2PRE\_b0  
MRCC\_private.h, 127  
RCC\_CFGR\_MOC2PRE\_b1  
MRCC\_private.h, 127  
RCC\_CFGR\_MOC2PRE\_b2  
MRCC\_private.h, 127  
RCC\_CFGR\_PPRE1\_b0  
MRCC\_private.h, 130  
RCC\_CFGR\_PPRE1\_b1  
MRCC\_private.h, 130  
RCC\_CFGR\_PPRE1\_b2  
MRCC\_private.h, 130  
RCC\_CFGR\_PPRE2\_b0  
MRCC\_private.h, 129  
RCC\_CFGR\_PPRE2\_b1  
MRCC\_private.h, 130  
RCC\_CFGR\_PPRE2\_b2  
MRCC\_private.h, 130  
RCC\_CFGR\_RTCPRE\_b0  
MRCC\_private.h, 129  
RCC\_CFGR\_RTCPRE\_b1  
MRCC\_private.h, 129  
RCC\_CFGR\_RTCPRE\_b2  
MRCC\_private.h, 129  
RCC\_CFGR\_RTCPRE\_b3  
MRCC\_private.h, 129  
RCC\_CFGR\_RTCPRE\_b4  
MRCC\_private.h, 129  
RCC\_CFGR\_SW\_b0  
MRCC\_private.h, 131  
RCC\_CFGR\_SW\_b1  
MRCC\_private.h, 132  
RCC\_CFGR\_SWS\_b0  
MRCC\_private.h, 131  
RCC\_CFGR\_SWS\_b1  
MRCC\_private.h, 131  
RCC\_CR\_CSSON  
MRCC\_private.h, 122  
RCC\_CR\_HSEBYP  
MRCC\_private.h, 122  
RCC\_CR\_HSEON  
MRCC\_private.h, 123  
RCC\_CR\_HSERDY

[MRCC\\_private.h, 122](#)  
[RCC\\_CR\\_HSION](#)  
[MRCC\\_private.h, 123](#)  
[RCC\\_CR\\_HSIRDY](#)  
[MRCC\\_private.h, 123](#)  
[RCC\\_CR\\_PLLI2SON](#)  
[MRCC\\_private.h, 122](#)  
[RCC\\_CR\\_PLLI2SRDY](#)  
[MRCC\\_private.h, 121](#)  
[RCC\\_CR\\_PLLON](#)  
[MRCC\\_private.h, 122](#)  
[RCC\\_CR\\_PLLRDY](#)  
[MRCC\\_private.h, 122](#)  
[RCC\\_MemoryMapType, 44](#)  
[AHB1ENR, 47](#)  
[AHB1LPENR, 48](#)  
[AHB1RSTR, 46](#)  
[AHB2ENR, 47](#)  
[AHB2LPENR, 48](#)  
[AHB2RSTR, 46](#)  
[APB1ENR, 48](#)  
[APB1LPENR, 49](#)  
[APB1RSTR, 46](#)  
[APB2ENR, 48](#)  
[APB2LPENR, 49](#)  
[APB2RSTR, 46](#)  
[BDCR, 50](#)  
[CFGR, 45](#)  
[CIR, 45](#)  
[CR, 45](#)  
[CSR, 50](#)  
[DCKCFGR, 50](#)  
[PLLCFGR, 45](#)  
[PLLI2SCFGR, 50](#)  
[Reserved1, 46](#)  
[Reserved10, 49](#)  
[Reserved11, 49](#)  
[Reserved12, 49](#)  
[Reserved13, 50](#)  
[Reserved14, 50](#)  
[Reserved2, 46](#)  
[Reserved3, 47](#)  
[Reserved4, 47](#)  
[Reserved5, 47](#)  
[Reserved6, 47](#)  
[Reserved7, 48](#)  
[Reserved8, 48](#)  
[Reserved9, 49](#)  
[SSCGR, 50](#)  
[RCC\\_PLLCFGR\\_PLLM\\_b0](#)  
[MRCC\\_private.h, 126](#)  
[RCC\\_PLLCFGR\\_PLLM\\_b1](#)  
[MRCC\\_private.h, 126](#)  
[RCC\\_PLLCFGR\\_PLLM\\_b2](#)  
[MRCC\\_private.h, 126](#)  
[RCC\\_PLLCFGR\\_PLLM\\_b3](#)  
[MRCC\\_private.h, 126](#)  
[RCC\\_PLLCFGR\\_PLLM\\_b4](#)  
[MRCC\\_private.h, 126](#)  
[RCC\\_PLLCFGR\\_PLLM\\_b5](#)  
[MRCC\\_private.h, 127](#)  
[RCC\\_PLLCFGR\\_PLLN\\_b0](#)  
[MRCC\\_private.h, 124](#)  
[RCC\\_PLLCFGR\\_PLLN\\_b1](#)  
[MRCC\\_private.h, 124](#)  
[RCC\\_PLLCFGR\\_PLLN\\_b2](#)  
[MRCC\\_private.h, 125](#)  
[RCC\\_PLLCFGR\\_PLLN\\_b3](#)  
[MRCC\\_private.h, 125](#)  
[RCC\\_PLLCFGR\\_PLLN\\_b4](#)  
[MRCC\\_private.h, 125](#)  
[RCC\\_PLLCFGR\\_PLLN\\_b5](#)  
[MRCC\\_private.h, 125](#)  
[RCC\\_PLLCFGR\\_PLLN\\_b6](#)  
[MRCC\\_private.h, 125](#)  
[RCC\\_PLLCFGR\\_PLLN\\_b7](#)  
[MRCC\\_private.h, 125](#)  
[RCC\\_PLLCFGR\\_PLLN\\_b8](#)  
[MRCC\\_private.h, 126](#)  
[RCC\\_PLLCFGR\\_PLLP\\_b0](#)  
[MRCC\\_private.h, 124](#)  
[RCC\\_PLLCFGR\\_PLLP\\_b1](#)  
[MRCC\\_private.h, 124](#)  
[RCC\\_PLLCFGR\\_PLLQ\\_b0](#)  
[MRCC\\_private.h, 123](#)  
[RCC\\_PLLCFGR\\_PLLQ\\_b1](#)  
[MRCC\\_private.h, 123](#)  
[RCC\\_PLLCFGR\\_PLLQ\\_b2](#)  
[MRCC\\_private.h, 123](#)  
[RCC\\_PLLCFGR\\_PLLQ\\_b3](#)  
[MRCC\\_private.h, 124](#)  
[RCC\\_PLLCFGR\\_PLLSRC](#)  
[MRCC\\_private.h, 124](#)  
[README.md, 51](#)  
[Reserved1](#)  
[RCC\\_MemoryMapType, 46](#)  
[Reserved10](#)  
[RCC\\_MemoryMapType, 49](#)  
[Reserved11](#)  
[RCC\\_MemoryMapType, 49](#)  
[Reserved12](#)  
[RCC\\_MemoryMapType, 49](#)  
[Reserved13](#)  
[RCC\\_MemoryMapType, 50](#)  
[Reserved14](#)  
[RCC\\_MemoryMapType, 50](#)  
[Reserved2](#)  
[RCC\\_MemoryMapType, 46](#)  
[Reserved3](#)  
[RCC\\_MemoryMapType, 47](#)  
[Reserved4](#)  
[RCC\\_MemoryMapType, 47](#)  
[Reserved5](#)  
[RCC\\_MemoryMapType, 47](#)  
[Reserved6](#)  
[RCC\\_MemoryMapType, 47](#)

- Reserved7
  - RCC\_MemoryMapType, [48](#)
- Reserved8
  - RCC\_MemoryMapType, [48](#)
- Reserved9
  - RCC\_MemoryMapType, [49](#)
- RTCPRE
  - MRCC\_config.h, [87](#)
- s16\_t
  - Standard types, [17](#)
- s32\_t
  - Standard types, [17](#)
- s8\_t
  - Standard types, [16](#)
- SDIOEN
  - MRCC\_config.h, [93](#)
- SET\_BIT
  - Bit Manipulation Math Macros, [9](#)
- SET\_BITS
  - Bit Group Manipulation Math Macros, [11](#)
- SPI1EN
  - MRCC\_config.h, [92](#)
- SPI2EN
  - MRCC\_config.h, [91](#)
- SPI3EN
  - MRCC\_config.h, [90](#)
- SPI4EN
  - MRCC\_config.h, [92](#)
- src/COTS/LIB/LSTD\_BITMATH.h, [51](#), [52](#)
- src/COTS/LIB/LSTD\_COMPILER.h, [52](#), [53](#)
- src/COTS/LIB/LSTD\_MCU\_UTILITIES.h, [53](#), [54](#)
- src/COTS/LIB/LSTD\_TYPES.h, [54](#), [55](#)
- src/COTS/LIB/LSTD\_VALUES.h, [55](#), [56](#)
- src/COTS/MCAL/GPIO/GPIO\_config.h, [56](#), [57](#)
- src/COTS/MCAL/GPIO/GPIO\_interface.h, [57](#), [66](#)
- src/COTS/MCAL/GPIO/GPIO\_private.h, [68](#), [69](#)
- src/COTS/MCAL/GPIO/GPIO\_program.c, [69](#), [77](#)
- src/COTS/MCAL/RCC/MRCC\_config.h, [83](#), [94](#)
- src/COTS/MCAL/RCC/MRCC\_interface.h, [101](#), [116](#)
- src/COTS/MCAL/RCC/MRCC\_private.h, [116](#), [159](#)
- src/COTS/MCAL/RCC/MRCC\_program.c, [163](#), [169](#)
- SSCGR
  - RCC\_MemoryMapType, [50](#)
- Standard types, [15](#)
  - bool\_t, [15](#)
  - fl32\_t, [17](#)
  - fl64\_t, [17](#)
  - s16\_t, [17](#)
  - s32\_t, [17](#)
  - s8\_t, [16](#)
  - u16\_t, [16](#)
  - u32\_t, [16](#)
  - u8\_t, [16](#)
- Standard values, [18](#)
  - FALSE, [18](#)
  - FLAG\_CLEARED, [19](#)
  - FLAG\_SET, [19](#)
  - INITIAL\_ZERO, [19](#)
  - NULL, [18](#)
  - TRUE, [18](#)
- STATIC
  - Compiler standard macros, [14](#)
- SW
  - MRCC\_config.h, [88](#)
- SWS
  - MRCC\_config.h, [88](#)
- SYSCFGEN
  - MRCC\_config.h, [92](#)
- SYSCLK
  - MRCC\_private.h, [138](#)
- SYSCLKby128
  - MRCC\_private.h, [141](#)
- SYSCLKby16
  - MRCC\_private.h, [141](#)
- SYSCLKby2
  - MRCC\_private.h, [140](#)
- SYSCLKby256
  - MRCC\_private.h, [141](#)
- SYSCLKby4
  - MRCC\_private.h, [140](#)
- SYSCLKby512
  - MRCC\_private.h, [141](#)
- SYSCLKby64
  - MRCC\_private.h, [141](#)
- SYSCLKby8
  - MRCC\_private.h, [141](#)
- TIM10EN
  - MRCC\_config.h, [92](#)
- TIM11EN
  - MRCC\_config.h, [92](#)
- TIM1EN
  - MRCC\_config.h, [93](#)
- TIM2EN
  - MRCC\_config.h, [91](#)
- TIM3EN
  - MRCC\_config.h, [91](#)
- TIM4EN
  - MRCC\_config.h, [91](#)
- TIM5EN
  - MRCC\_config.h, [91](#)
- TIM9EN
  - MRCC\_config.h, [92](#)
- TOGGLE\_BIT
  - Bit Manipulation Math Macros, [10](#)
- TOGGLE\_BITS
  - Bit Group Manipulation Math Macros, [11](#)
- TRUE
  - Standard values, [18](#)
- u16\_t
  - Standard types, [16](#)
- u32\_t
  - Standard types, [16](#)
- u8\_t
  - Standard types, [16](#)
- USART1EN

- MRCC\_config.h, [93](#)
- USART2EN
  - MRCC\_config.h, [90](#)
- USART6EN
  - MRCC\_config.h, [93](#)
- Utilities macros, [15](#)
  - MY\_MS\_DELAY, [15](#)
- VAR
  - Compiler standard macros, [12](#)
- WWDGEN
  - MRCC\_config.h, [91](#)