

# LPCOpen Platform for LPC5411X microcontrollers

## 5411x

Generated by Doxygen 1.8.10

Fri Mar 4 2016 18:34:19



# Contents

<b>1</b>	<b>LPCOpen Platform for the NXP LPC5411X family of Microcontrollers</b>	<b>1</b>
<b>2</b>	<b>MISRA-C:2004 Compliance Exceptions</b>	<b>3</b>
<b>3</b>	<b>Module Index</b>	<b>5</b>
3.1	Modules . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>9</b>
4.1	Data Structures . . . . .	9
<b>5</b>	<b>File Index</b>	<b>13</b>
5.1	File List . . . . .	13
<b>6</b>	<b>Module Documentation</b>	<b>15</b>
6.1	BOARD: Common board components used with board drivers . . . . .	15
6.2	BOARD: LPC5411X boards . . . . .	16
6.2.1	Detailed Description . . . . .	16
6.3	Board specific drivers and support functions . . . . .	17
6.3.1	Detailed Description . . . . .	17
6.4	CHIP: LPC5411X A/D conversion driver . . . . .	18
6.4.1	Detailed Description . . . . .	18
6.4.2	Macro Definition Documentation . . . . .	21
6.4.2.1	ADC_CALIB . . . . .	21
6.4.2.2	ADC_CALREQD . . . . .	21
6.4.2.3	ADC_CR_ASYNC_MODE . . . . .	21
6.4.2.4	ADC_CR_BITACC . . . . .	21
6.4.2.5	ADC_CR_BYPASS . . . . .	21
6.4.2.6	ADC_CR_CALMODEBIT . . . . .	21
6.4.2.7	ADC_CR_CLKDIV . . . . .	22
6.4.2.8	ADC_CR_CLKDIV_BITPOS . . . . .	22
6.4.2.9	ADC_CR_CLKDIV_MASK . . . . .	22
6.4.2.10	ADC_CR_LPWRMODEBIT . . . . .	22
6.4.2.11	ADC_CR_RESOL . . . . .	22

6.4.2.12	ADC_CR_TSAMP . . . . .	22
6.4.2.13	ADC_DR_CHAN_BITPOS . . . . .	22
6.4.2.14	ADC_DR_CHAN_MASK . . . . .	22
6.4.2.15	ADC_DR_CHANNEL . . . . .	22
6.4.2.16	ADC_DR_DATAVALID . . . . .	23
6.4.2.17	ADC_DR_DONE . . . . .	23
6.4.2.18	ADC_DR_OVERRUN . . . . .	23
6.4.2.19	ADC_DR_RESULT . . . . .	23
6.4.2.20	ADC_DR_RESULT_BITPOS . . . . .	23
6.4.2.21	ADC_DR_THCMPCROSS . . . . .	23
6.4.2.22	ADC_DR_THCMPCROSS_BITPOS . . . . .	23
6.4.2.23	ADC_DR_THCMPCROSS_MASK . . . . .	23
6.4.2.24	ADC_DR_THCMPRANGE . . . . .	23
6.4.2.25	ADC_DR_THCMPRANGE_BITPOS . . . . .	23
6.4.2.26	ADC_DR_THCMPRANGE_MASK . . . . .	24
6.4.2.27	ADC_FLAGS_OVRUN_INT_MASK . . . . .	24
6.4.2.28	ADC_FLAGS_OVRUN_MASK . . . . .	24
6.4.2.29	ADC_FLAGS_SEQA_INT_MASK . . . . .	24
6.4.2.30	ADC_FLAGS_SEQA_OVRUN_MASK . . . . .	24
6.4.2.31	ADC_FLAGS_SEQB_INT_MASK . . . . .	24
6.4.2.32	ADC_FLAGS_SEQB_OVRUN_MASK . . . . .	24
6.4.2.33	ADC_FLAGS_SEQN_INT_MASK . . . . .	24
6.4.2.34	ADC_FLAGS_SEQN_OVRUN_MASK . . . . .	24
6.4.2.35	ADC_FLAGS_THCMP_INT_MASK . . . . .	25
6.4.2.36	ADC_FLAGS_THCMP_MASK . . . . .	25
6.4.2.37	ADC_INTEN_CMP_CROSSTH . . . . .	25
6.4.2.38	ADC_INTEN_CMP_DISBALE . . . . .	25
6.4.2.39	ADC_INTEN_CMP_ENABLE . . . . .	25
6.4.2.40	ADC_INTEN_CMP_MASK . . . . .	25
6.4.2.41	ADC_INTEN_CMP_OUTSIDETH . . . . .	25
6.4.2.42	ADC_INTEN_OVRUN_ENABLE . . . . .	25
6.4.2.43	ADC_INTEN_SEQA_ENABLE . . . . .	25
6.4.2.44	ADC_INTEN_SEQB_ENABLE . . . . .	26
6.4.2.45	ADC_INTEN_SEQN_ENABLE . . . . .	26
6.4.2.46	ADC_MAX_CHANNEL_NUM . . . . .	26
6.4.2.47	ADC_MAX_SAMPLE_RATE . . . . .	26
6.4.2.48	ADC_SAMPLE_RATE_CONFIG_MASK . . . . .	26
6.4.2.49	ADC_SEQ_CTRL_BURST . . . . .	26
6.4.2.50	ADC_SEQ_CTRL_CHANNEL_EN . . . . .	26
6.4.2.51	ADC_SEQ_CTRL_HWTRIG_POLPOS . . . . .	26

6.4.2.52	ADC_SEQ_CTRL_HWTRIG_SYNCBYPASS	26
6.4.2.53	ADC_SEQ_CTRL_LOWPRIO	26
6.4.2.54	ADC_SEQ_CTRL_MODE_EOS	27
6.4.2.55	ADC_SEQ_CTRL_SEQ_ENA	27
6.4.2.56	ADC_SEQ_CTRL_SINGLESTEP	27
6.4.2.57	ADC_SEQ_CTRL_START	27
6.4.2.58	ADC_SEQ_CTRL_TRIGGER	27
6.4.2.59	ADC_SEQ_GDAT_CHAN_BITPOS	27
6.4.2.60	ADC_SEQ_GDAT_CHAN_MASK	27
6.4.2.61	ADC_SEQ_GDAT_DATAVALID	27
6.4.2.62	ADC_SEQ_GDAT_OVERRUN	27
6.4.2.63	ADC_SEQ_GDAT_RESULT_BITPOS	28
6.4.2.64	ADC_SEQ_GDAT_RESULT_MASK	28
6.4.2.65	ADC_SEQ_GDAT_THCMPCROSS_BITPOS	28
6.4.2.66	ADC_SEQ_GDAT_THCMPCROSS_MASK	28
6.4.2.67	ADC_SEQ_GDAT_THCMPRANGE_BITPOS	28
6.4.2.68	ADC_SEQ_GDAT_THCMPRANGE_MASK	28
6.4.2.69	ADC_STARTUP_ENABLE	28
6.4.2.70	ADC_STARTUP_INIT	28
6.4.2.71	ADC_THR_VAL_MASK	28
6.4.2.72	ADC_THR_VAL_POS	29
6.4.2.73	ADC_THRSEL_CHAN_SEL_THR1	29
6.4.3	Enumeration Type Documentation	29
6.4.3.1	ADC_DR_THCMPCROSS_T	29
6.4.3.2	ADC_DR_THCMPRANGE_T	29
6.4.3.3	ADC_INTEN_THCMP_T	29
6.4.3.4	ADC_SEQ_IDX_T	30
6.4.3.5	ADC_TSAMP_T	30
6.4.4	Function Documentation	30
6.4.4.1	Chip_ADC_Calibration(LPC_ADC_T *pADC)	30
6.4.4.2	Chip_ADC_ClearFlags(LPC_ADC_T *pADC, uint32_t flags)	30
6.4.4.3	Chip_ADC_ClearSequencerBits(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex, uint32_t bits)	31
6.4.4.4	Chip_ADC_ClearTHRSELBits(LPC_ADC_T *pADC, uint32_t mask)	31
6.4.4.5	Chip_ADC_DeInit(LPC_ADC_T *pADC)	31
6.4.4.6	Chip_ADC_DisableInt(LPC_ADC_T *pADC, uint32_t intMask)	32
6.4.4.7	Chip_ADC_DisableSequencer(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex)	32
6.4.4.8	Chip_ADC_EnableInt(LPC_ADC_T *pADC, uint32_t intMask)	32
6.4.4.9	Chip_ADC_EnableSequencer(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex)	33
6.4.4.10	Chip_ADC_GetDataReg(LPC_ADC_T *pADC, uint8_t index)	33

6.4.4.11	Chip_ADC_GetDivider(LPC_ADC_T *pADC)	34
6.4.4.12	Chip_ADC_GetFlags(LPC_ADC_T *pADC)	35
6.4.4.13	Chip_ADC_GetGlobalDataReg(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex)	35
6.4.4.14	Chip_ADC_GetSequencerCtrl(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex)	36
6.4.4.15	Chip_ADC_Init(LPC_ADC_T *pADC, uint32_t flags)	36
6.4.4.16	Chip_ADC_SelectTempSensorInput(LPC_ADC_T *pADC)	36
6.4.4.17	Chip_ADC_SelectTH0Channels(LPC_ADC_T *pADC, uint32_t channels)	37
6.4.4.18	Chip_ADC_SelectTH1Channels(LPC_ADC_T *pADC, uint32_t channels)	37
6.4.4.19	Chip_ADC_SetClockRate(LPC_ADC_T *pADC, uint32_t rate)	37
6.4.4.20	Chip_ADC_SetDivider(LPC_ADC_T *pADC, uint8_t div)	38
6.4.4.21	Chip_ADC_SetSequencerBits(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex, uint32_t bits)	38
6.4.4.22	Chip_ADC_SetThresholdInt(LPC_ADC_T *pADC, uint8_t ch, ADC_INTEN_T HCMP_T thInt)	39
6.4.4.23	Chip_ADC_SetThrHighValue(LPC_ADC_T *pADC, uint8_t thrnum, uint16_t value)	40
6.4.4.24	Chip_ADC_SetThrLowValue(LPC_ADC_T *pADC, uint8_t thrnum, uint16_t value)	40
6.4.4.25	Chip_ADC_SetTHRSELBits(LPC_ADC_T *pADC, uint32_t mask)	40
6.4.4.26	Chip_ADC_SetupSequencer(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex, uint32_t options)	41
6.4.4.27	Chip_ADC_StartBurstSequencer(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex)	42
6.4.4.28	Chip_ADC_StartSequencer(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex)	42
6.4.4.29	Chip_ADC_StopBurstSequencer(LPC_ADC_T *pADC, ADC_SEQ_IDX_T seqIndex)	43
6.5	CHIP: CHIP_LPC5411X family IRQ vector names and mapped NVIC IRQ numbers	44
6.6	CHIP: Common Chip ISP/IAP commands and return codes	46
6.6.1	Detailed Description	46
6.6.2	Macro Definition Documentation	47
6.6.2.1	IAP_ADDR_ERROR	47
6.6.2.2	IAP_ADDR_NOT_MAPPED	47
6.6.2.3	IAP_BLANK_CHECK_SECTOR_CMD	47
6.6.2.4	IAP_BUSY	47
6.6.2.5	IAP_CMD_LOCKED	47
6.6.2.6	IAP_CMD_SUCCESS	47
6.6.2.7	IAP_COMPARE_CMD	47
6.6.2.8	IAP_COMPARE_ERROR	48
6.6.2.9	IAP_COUNT_ERROR	48
6.6.2.10	IAP_CRP_ENABLED	48
6.6.2.11	IAP_DST_ADDR_ERROR	48
6.6.2.12	IAP_DST_ADDR_NOT_MAPPED	48
6.6.2.13	IAP_EEPROM_READ	48

6.6.2.14	IAP_EEPROM_WRITE . . . . .	48
6.6.2.15	IAP_ERASE_PAGE_CMD . . . . .	48
6.6.2.16	IAP_ERSECTOR_CMD . . . . .	48
6.6.2.17	IAP_INVALID_BAUD_RATE . . . . .	49
6.6.2.18	IAP_INVALID_CODE . . . . .	49
6.6.2.19	IAP_INVALID_COMMAND . . . . .	49
6.6.2.20	IAP_INVALID_SECTOR . . . . .	49
6.6.2.21	IAP_INVALID_STOP_BIT . . . . .	49
6.6.2.22	IAP_PARAM_ERROR . . . . .	49
6.6.2.23	IAP_PREWRITE_CMD . . . . .	49
6.6.2.24	IAP_READ_BOOT_CODE_CMD . . . . .	49
6.6.2.25	IAP_READ_UID_CMD . . . . .	49
6.6.2.26	IAP_REINVOKE_ISP_CMD . . . . .	50
6.6.2.27	IAP_REPID_CMD . . . . .	50
6.6.2.28	IAP_SECTOR_NOT_BLANK . . . . .	50
6.6.2.29	IAP_SECTOR_NOT_PREPARED . . . . .	50
6.6.2.30	IAP_SRC_ADDR_ERROR . . . . .	50
6.6.2.31	IAP_SRC_ADDR_NOT_MAPPED . . . . .	50
6.6.2.32	IAP_WRISECTOR_CMD . . . . .	50
6.6.3	Typedef Documentation . . . . .	50
6.6.3.1	IAP_ENTRY_T . . . . .	50
6.6.4	Function Documentation . . . . .	50
6.6.4.1	Chip_IAP_BlankCheckSector(uint32_t strSector, uint32_t endSector) . . . . .	50
6.6.4.2	Chip_IAP_Compare(uint32_t dstAdd, uint32_t srcAdd, uint32_t bytescmp) . . . . .	51
6.6.4.3	Chip_IAP_CopyRamToFlash(uint32_t dstAdd, uint32_t *srcAdd, uint32_t byteswrt) . . . . .	51
6.6.4.4	Chip_IAP_ErasePage(uint32_t strPage, uint32_t endPage) . . . . .	51
6.6.4.5	Chip_IAP_EraseSector(uint32_t strSector, uint32_t endSector) . . . . .	52
6.6.4.6	Chip_IAP_PreSectorForReadWrite(uint32_t strSector, uint32_t endSector) . . . . .	52
6.6.4.7	Chip_IAP_ReadBootCode(void) . . . . .	53
6.6.4.8	Chip_IAP_ReadPID(void) . . . . .	53
6.6.4.9	Chip_IAP_ReadUID(void) . . . . .	53
6.6.4.10	Chip_IAP_ReinvokeISP(void) . . . . .	53
6.7	CHIP: FPU initialization . . . . .	54
6.7.1	Detailed Description . . . . .	54
6.7.2	Function Documentation . . . . .	54
6.7.2.1	fpulnit(void) . . . . .	54
6.8	CHIP: LPC Common Types . . . . .	55
6.8.1	Detailed Description . . . . .	55
6.9	CHIP: LPC5410x family CMSIS include files . . . . .	56
6.10	CHIP: LPC5411X 32-bit Timer driver . . . . .	57

6.10.1	Detailed Description	57
6.10.2	Macro Definition Documentation	59
6.10.2.1	TIMER_CAP_FALLING	59
6.10.2.2	TIMER_CAP_INT	59
6.10.2.3	TIMER_CAP_RISING	59
6.10.2.4	TIMER_CCR_MASK	59
6.10.2.5	TIMER_CTCR_MASK	59
6.10.2.6	TIMER_CTRL_MASK	59
6.10.2.7	TIMER_EMR_MASK	59
6.10.2.8	TIMER_ENABLE	59
6.10.2.9	TIMER_INT_ON_CAP	59
6.10.2.10	TIMER_INT_ON_MATCH	60
6.10.2.11	TIMER_IR_CLR	60
6.10.2.12	TIMER_MATCH_INT	60
6.10.2.13	TIMER_MCR_MASK	60
6.10.2.14	TIMER_RESET	60
6.10.2.15	TIMER_RESET_ON_MATCH	60
6.10.2.16	TIMER_STOP_ON_MATCH	60
6.10.3	Enumeration Type Documentation	60
6.10.3.1	TIMER_CAP_SRC_STATE_T	60
6.10.3.2	TIMER_PIN_MATCH_STATE_T	61
6.10.4	Function Documentation	61
6.10.4.1	Chip_TIMER_CaptureDisableInt(LPC_TIMER_T *pTMR, int8_t capnum)	61
6.10.4.2	Chip_TIMER_CaptureEnableInt(LPC_TIMER_T *pTMR, int8_t capnum)	61
6.10.4.3	Chip_TIMER_CaptureFallingEdgeDisable(LPC_TIMER_T *pTMR, int8_t capnum)	61
6.10.4.4	Chip_TIMER_CaptureFallingEdgeEnable(LPC_TIMER_T *pTMR, int8_t capnum)	62
6.10.4.5	Chip_TIMER_CapturePending(LPC_TIMER_T *pTMR, int8_t capnum)	62
6.10.4.6	Chip_TIMER_CaptureRisingEdgeDisable(LPC_TIMER_T *pTMR, int8_t capnum)	62
6.10.4.7	Chip_TIMER_CaptureRisingEdgeEnable(LPC_TIMER_T *pTMR, int8_t capnum)	63
6.10.4.8	Chip_TIMER_ClearCapture(LPC_TIMER_T *pTMR, int8_t capnum)	64
6.10.4.9	Chip_TIMER_ClearMatch(LPC_TIMER_T *pTMR, int8_t matchnum)	64
6.10.4.10	Chip_TIMER_DeInit(LPC_TIMER_T *pTMR)	64
6.10.4.11	Chip_TIMER_Disable(LPC_TIMER_T *pTMR)	65
6.10.4.12	Chip_TIMER_Enable(LPC_TIMER_T *pTMR)	65
6.10.4.13	Chip_TIMER_ExtMatchControlSet(LPC_TIMER_T *pTMR, int8_t initial_state, TIMER_PIN_MATCH_STATE_T matchState, int8_t matchnum)	65
6.10.4.14	Chip_TIMER_Init(LPC_TIMER_T *pTMR)	66
6.10.4.15	Chip_TIMER_MatchDisableInt(LPC_TIMER_T *pTMR, int8_t matchnum)	66
6.10.4.16	Chip_TIMER_MatchEnableInt(LPC_TIMER_T *pTMR, int8_t matchnum)	66
6.10.4.17	Chip_TIMER_MatchPending(LPC_TIMER_T *pTMR, int8_t matchnum)	67



6.10.4.18	Chip_TIMER_PrescaleSet(LPC_TIMER_T *pTMR, uint32_t prescale)	67
6.10.4.19	Chip_TIMER_ReadCapture(LPC_TIMER_T *pTMR, int8_t capnum)	67
6.10.4.20	Chip_TIMER_ReadCount(LPC_TIMER_T *pTMR)	68
6.10.4.21	Chip_TIMER_ReadPrescale(LPC_TIMER_T *pTMR)	68
6.10.4.22	Chip_TIMER_Reset(LPC_TIMER_T *pTMR)	68
6.10.4.23	Chip_TIMER_ResetOnMatchDisable(LPC_TIMER_T *pTMR, int8_t matchnum)	69
6.10.4.24	Chip_TIMER_ResetOnMatchEnable(LPC_TIMER_T *pTMR, int8_t matchnum)	69
6.10.4.25	Chip_TIMER_SetMatch(LPC_TIMER_T *pTMR, int8_t matchnum, uint32_t matchval)	69
6.10.4.26	Chip_TIMER_StopOnMatchDisable(LPC_TIMER_T *pTMR, int8_t matchnum)	70
6.10.4.27	Chip_TIMER_StopOnMatchEnable(LPC_TIMER_T *pTMR, int8_t matchnum)	71
6.10.4.28	Chip_TIMER_TIMER_SetCountClockSrc(LPC_TIMER_T *pTMR, TIMER_CAP_SRC_STATE_T capSrc, int8_t capnum)	71
6.11	CHIP: LPC5411X CPU multi-core support driver	72
6.11.1	Detailed Description	72
6.11.2	Enumeration Type Documentation	72
6.11.2.1	CORESELECT_T	72
6.11.3	Function Documentation	72
6.11.3.1	Chip_CPU_CM0Boot(uint32_t *coentry, uint32_t *costackptr)	72
6.11.3.2	Chip_CPU_CM4Boot(uint32_t *coentry, uint32_t *costackptr)	73
6.11.3.3	Chip_CPU_IsM4Core(void)	73
6.11.3.4	Chip_CPU_IsMasterCore(void)	73
6.11.3.5	Chip_CPU_SelectMasterCore(CORESELECT_T master, CORESELECT_T ownerPower)	73
6.12	CHIP: LPC5411X Clock Driver	74
6.12.1	Detailed Description	74
6.12.2	Macro Definition Documentation	77
6.12.2.1	Chip_Clock_GetIntOscRate	77
6.12.2.2	SYSCON_FRO12MHZ_FREQ	77
6.12.2.3	SYSCON_FRO48MHZ_FREQ	77
6.12.2.4	SYSCON_FRO96MHZ_FREQ	77
6.12.2.5	SYSCON_RTC_FREQ	78
6.12.2.6	SYSCON_WDTOSC_FREQ	78
6.12.3	Enumeration Type Documentation	78
6.12.3.1	CHIP_ASYNC_SYSCON_SRC_T	78
6.12.3.2	CHIP_SYSCON_ADCCLKSELSRC_T	78
6.12.3.3	CHIP_SYSCON_CLKOUTSRC_T	78
6.12.3.4	CHIP_SYSCON_CLOCK_T	79
6.12.3.5	CHIP_SYSCON_FLEXCOMMCLKSELSRC_T	80
6.12.3.6	CHIP_SYSCON_FRGCLKSRC_T	80
6.12.3.7	CHIP_SYSCON_MAIN_A_CLKSRC_T	80

6.12.3.8	CHIP_SYSCON_MAIN_B_CLKSRC_T	80
6.12.3.9	CHIP_SYSCON_MAINCLKSRC_T	81
6.12.3.10	CHIP_SYSCON_MCLKSRC_T	81
6.12.3.11	CHIP_SYSCON_USBCLSRC_T	81
6.12.3.12	WDT_OSC_FREQ_T	81
6.12.4	Function Documentation	82
6.12.4.1	Chip_Clock_DisablePeriphClock(CHIP_SYSCON_CLOCK_T clk)	82
6.12.4.2	Chip_Clock_DisableRTCOsc(void)	82
6.12.4.3	Chip_Clock_EnablePeriphClock(CHIP_SYSCON_CLOCK_T clk)	83
6.12.4.4	Chip_Clock_EnableRTCOsc(void)	84
6.12.4.5	Chip_Clock_GetADCClockDiv(void)	84
6.12.4.6	Chip_Clock_GetADCClockRate(void)	84
6.12.4.7	Chip_Clock_GetADCClockSource(void)	84
6.12.4.8	Chip_Clock_GetAsyncSyscon_ClockRate(void)	85
6.12.4.9	Chip_Clock_GetAsyncSysconClockSource(void)	85
6.12.4.10	Chip_Clock_GetCLKOUTDiv(void)	85
6.12.4.11	Chip_Clock_GetCLKOUTSource(void)	85
6.12.4.12	Chip_Clock_GetExtClockInRate(void)	85
6.12.4.13	Chip_Clock_GetFLEXCOMMClockRate(uint32_t id)	85
6.12.4.14	Chip_Clock_GetFLEXCOMMClockSource(uint32_t idx)	86
6.12.4.15	Chip_Clock_GetFRGClockRate(void)	86
6.12.4.16	Chip_Clock_GetFRGClockSource(void)	86
6.12.4.17	Chip_Clock_GetFRGInClockRate(void)	86
6.12.4.18	Chip_Clock_GetFROHFRate(void)	87
6.12.4.19	Chip_Clock_GetMain_A_ClockRate(void)	87
6.12.4.20	Chip_Clock_GetMain_A_ClockSource(void)	87
6.12.4.21	Chip_Clock_GetMain_B_ClockRate(void)	87
6.12.4.22	Chip_Clock_GetMain_B_ClockSource(void)	87
6.12.4.23	Chip_Clock_GetMainClockRate(void)	87
6.12.4.24	Chip_Clock_GetMainClockSource(void)	88
6.12.4.25	Chip_Clock_GetMCLKDir(void)	88
6.12.4.26	Chip_Clock_GetMCLKDiv(void)	88
6.12.4.27	Chip_Clock_GetMCLKSource(void)	88
6.12.4.28	Chip_Clock_GetRTCOsc(void)	88
6.12.4.29	Chip_Clock_GetRTCOscRate(void)	88
6.12.4.30	Chip_Clock_GetSysClockDiv(void)	88
6.12.4.31	Chip_Clock_GetSystemClockRate(void)	89
6.12.4.32	Chip_Clock_GetSysTickClockDiv(void)	89
6.12.4.33	Chip_Clock_GetSysTickClockRate(void)	89
6.12.4.34	Chip_Clock_GetUSBClockDiv(void)	89

6.12.4.35	Chip_Clock_GetUSBClockSource(void)	90
6.12.4.36	Chip_Clock_GetWDTOSCRate(void)	90
6.12.4.37	Chip_Clock_SetADCClockDiv(uint32_t div)	90
6.12.4.38	Chip_Clock_SetADCClockSource(CHIP_SYSCON_ADCCLKSELSRC_T src)	90
6.12.4.39	Chip_Clock_SetAsyncSysconClockSource(CHIP_ASYNC_SYSCON_SRC_T src)	90
6.12.4.40	Chip_Clock_SetCLKOUTSource(CHIP_SYSCON_CLKOUTSRC_T src, uint32_t div)	91
6.12.4.41	Chip_Clock_SetFLEXCOMMClockSource(uint32_t idx, CHIP_SYSCON_FLEXCOMMCLKSELSRC_T src)	91
6.12.4.42	Chip_Clock_SetFRGClockRate(uint32_t rate)	91
6.12.4.43	Chip_Clock_SetFRGClockSource(CHIP_SYSCON_FRGCLKSRC_T src)	92
6.12.4.44	Chip_Clock_SetMain_A_ClockSource(CHIP_SYSCON_MAIN_A_CLKSRC_T src)	92
6.12.4.45	Chip_Clock_SetMain_B_ClockSource(CHIP_SYSCON_MAIN_B_CLKSRC_T src)	92
6.12.4.46	Chip_Clock_SetMainClockSource(CHIP_SYSCON_MAINCLKSRC_T src)	92
6.12.4.47	Chip_Clock_SetMCLKClockSource(CHIP_SYSCON_MCLKSRC_T src, uint32_t div)	93
6.12.4.48	Chip_Clock_SetMCLKDir(int dir)	93
6.12.4.49	Chip_Clock_SetMCLKDirInput(void)	93
6.12.4.50	Chip_Clock_SetMCLKDirOutput(void)	93
6.12.4.51	Chip_Clock_SetSysClockDiv(uint32_t div)	94
6.12.4.52	Chip_Clock_SetSysTickClockDiv(uint32_t div)	95
6.12.4.53	Chip_Clock_SetUSBClockSource(CHIP_SYSCON_USBCLKSRC_T src, uint32_t div)	95
6.12.4.54	Chip_Clock_SetWDTOSCRate(WDT_OSC_FREQ_T freq, uint32_t div)	95
6.13	CHIP: LPC5411X Cyclic Redundancy Check Engine driver	97
6.13.1	Detailed Description	97
6.13.2	Macro Definition Documentation	98
6.13.2.1	CRC_MODE_POLY_BITMASK	98
6.13.2.2	CRC_MODE_POLY_CCITT	98
6.13.2.3	CRC_MODE_POLY_CRC16	98
6.13.2.4	CRC_MODE_POLY_CRC32	98
6.13.2.5	CRC_MODE_SUM_BIT_RVS	98
6.13.2.6	CRC_MODE_SUM_BITMASK	98
6.13.2.7	CRC_MODE_SUM_CMPL	99
6.13.2.8	CRC_MODE_WRDATA_BIT_RVS	99
6.13.2.9	CRC_MODE_WRDATA_BITMASK	99
6.13.2.10	CRC_MODE_WRDATA_CMPL	99
6.13.2.11	CRC_SEED_CCITT	99
6.13.2.12	CRC_SEED_CRC16	99
6.13.2.13	CRC_SEED_CRC32	99
6.13.2.14	MODE_CFG_CCITT	99

6.13.2.15	MODE_CFG_CRC16	99
6.13.2.16	MODE_CFG_CRC32	99
6.13.3	Enumeration Type Documentation	99
6.13.3.1	CRC_POLY_T	99
6.13.4	Function Documentation	100
6.13.4.1	Chip_CRC_CRC16(LPC_CRC_T *pCRC, const uint16_t *data, uint32_t hwords)	100
6.13.4.2	Chip_CRC_CRC32(LPC_CRC_T *pCRC, const uint32_t *data, uint32_t words)	100
6.13.4.3	Chip_CRC_CRC8(LPC_CRC_T *pCRC, const uint8_t *data, uint32_t bytes)	100
6.13.4.4	Chip_CRC_Deinit(LPC_CRC_T *pCRC)	100
6.13.4.5	Chip_CRC_GetMode(LPC_CRC_T *pCRC)	101
6.13.4.6	Chip_CRC_GetSeed(LPC_CRC_T *pCRC)	101
6.13.4.7	Chip_CRC_Init(LPC_CRC_T *pCRC)	101
6.13.4.8	Chip_CRC_SetMode(LPC_CRC_T *pCRC, uint32_t mode)	101
6.13.4.9	Chip_CRC_SetPoly(LPC_CRC_T *pCRC, CRC_POLY_T poly, uint32_t flags)	102
6.13.4.10	Chip_CRC_SetSeed(LPC_CRC_T *pCRC, uint32_t seed)	102
6.13.4.11	Chip_CRC_Sum(LPC_CRC_T *pCRC)	102
6.13.4.12	Chip_CRC_UseCCITT(LPC_CRC_T *pCRC)	103
6.13.4.13	Chip_CRC_UseCRC16(LPC_CRC_T *pCRC)	103
6.13.4.14	Chip_CRC_UseCRC32(LPC_CRC_T *pCRC)	103
6.13.4.15	Chip_CRC_UseDefaultConfig(LPC_CRC_T *pCRC, CRC_POLY_T poly)	103
6.13.4.16	Chip_CRC_Write16(LPC_CRC_T *pCRC, uint16_t data)	104
6.13.4.17	Chip_CRC_Write32(LPC_CRC_T *pCRC, uint32_t data)	104
6.13.4.18	Chip_CRC_Write8(LPC_CRC_T *pCRC, uint8_t data)	104
6.14	CHIP: LPC5411X DMA Controller driver channel specific functions (legacy)	105
6.14.1	Detailed Description	105
6.14.2	Function Documentation	105
6.14.2.1	Chip_DMA_ClearTranBits(LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t mask)	105
6.14.2.2	Chip_DMA_GetChannelStatus(LPC_DMA_T *pDMA, DMA_CHID_T ch)	106
6.14.2.3	Chip_DMA_SetChannelInvalid(LPC_DMA_T *pDMA, DMA_CHID_T ch)	106
6.14.2.4	Chip_DMA_SetChannelValid(LPC_DMA_T *pDMA, DMA_CHID_T ch)	106
6.14.2.5	Chip_DMA_SetTranBits(LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t mask)	107
6.14.2.6	Chip_DMA_SetupChannelConfig(LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t cfg)	108
6.14.2.7	Chip_DMA_SetupChannelTransfer(LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t cfg)	108
6.14.2.8	Chip_DMA_SetupChannelTransferSize(LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t trans)	109
6.14.2.9	Chip_DMA_SetupTranChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch, DMA_CHDESC_T *desc)	109
6.14.2.10	Chip_DMA_SWTriggerChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	110
6.14.3	Variable Documentation	110

6.14.3.1	Chip_DMA_Table	110
6.15	CHIP: LPC5411X DMA Controller driver common channel functions (legacy)	111
6.15.1	Detailed Description	111
6.15.2	Function Documentation	111
6.15.2.1	Chip_DMA_AbortChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	111
6.15.2.2	Chip_DMA_ClearActiveIntAChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	112
6.15.2.3	Chip_DMA_ClearActiveIntBChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	112
6.15.2.4	Chip_DMA_ClearErrorIntChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	112
6.15.2.5	Chip_DMA_DisableChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	113
6.15.2.6	Chip_DMA_DisableIntChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	113
6.15.2.7	Chip_DMA_EnableChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	113
6.15.2.8	Chip_DMA_EnableIntChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	113
6.15.2.9	Chip_DMA_GetActiveChannels(LPC_DMA_T *pDMA)	114
6.15.2.10	Chip_DMA_GetActiveIntAChannels(LPC_DMA_T *pDMA)	114
6.15.2.11	Chip_DMA_GetActiveIntBChannels(LPC_DMA_T *pDMA)	114
6.15.2.12	Chip_DMA_GetBusyChannels(LPC_DMA_T *pDMA)	115
6.15.2.13	Chip_DMA_GetEnabledChannels(LPC_DMA_T *pDMA)	115
6.15.2.14	Chip_DMA_GetEnableIntChannels(LPC_DMA_T *pDMA)	115
6.15.2.15	Chip_DMA_GetErrorIntChannels(LPC_DMA_T *pDMA)	116
6.15.2.16	Chip_DMA_SetTrigChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	116
6.15.2.17	Chip_DMA_SetValidChannel(LPC_DMA_T *pDMA, DMA_CHID_T ch)	116
6.16	CHIP: LPC5411X DMA Controller driver common functions (legacy)	118
6.16.1	Detailed Description	118
6.16.2	Function Documentation	118
6.16.2.1	Chip_DMA_DeInit(LPC_DMA_T *pDMA)	118
6.16.2.2	Chip_DMA_Disable(LPC_DMA_T *pDMA)	118
6.16.2.3	Chip_DMA_Enable(LPC_DMA_T *pDMA)	118
6.16.2.4	Chip_DMA_GetIntStatus(LPC_DMA_T *pDMA)	119
6.16.2.5	Chip_DMA_GetSRAMBase(LPC_DMA_T *pDMA)	119
6.16.2.6	Chip_DMA_Init(LPC_DMA_T *pDMA)	119
6.16.2.7	Chip_DMA_SetSRAMBase(LPC_DMA_T *pDMA, uint32_t base)	119
6.17	CHIP: LPC5411X DMA Engine driver (legacy)	121
6.17.1	Detailed Description	121
6.17.2	Macro Definition Documentation	122
6.17.2.1	DMA_ADDR	122
6.17.2.2	DMA_CFG_BURSTPOWER	122
6.17.2.3	DMA_CFG_BURSTPOWER_1	122
6.17.2.4	DMA_CFG_BURSTPOWER_1024	123
6.17.2.5	DMA_CFG_BURSTPOWER_128	123
6.17.2.6	DMA_CFG_BURSTPOWER_16	123

6.17.2.7 DMA_CFG_BURSTPOWER_2 . . . . .	123
6.17.2.8 DMA_CFG_BURSTPOWER_256 . . . . .	123
6.17.2.9 DMA_CFG_BURSTPOWER_32 . . . . .	123
6.17.2.10 DMA_CFG_BURSTPOWER_4 . . . . .	123
6.17.2.11 DMA_CFG_BURSTPOWER_512 . . . . .	123
6.17.2.12 DMA_CFG_BURSTPOWER_64 . . . . .	123
6.17.2.13 DMA_CFG_BURSTPOWER_8 . . . . .	124
6.17.2.14 DMA_CFG_CHPRIORITY . . . . .	124
6.17.2.15 DMA_CFG_DSTBURSTWRAP . . . . .	124
6.17.2.16 DMA_CFG_HWTRIGEN . . . . .	124
6.17.2.17 DMA_CFG_PERIPHREQEN . . . . .	124
6.17.2.18 DMA_CFG_SRCBURSTWRAP . . . . .	124
6.17.2.19 DMA_CFG_TRIGBURST_BURST . . . . .	124
6.17.2.20 DMA_CFG_TRIGBURST_SNGL . . . . .	124
6.17.2.21 DMA_CFG_TRIGPOL_HIGH . . . . .	124
6.17.2.22 DMA_CFG_TRIGPOL_LOW . . . . .	125
6.17.2.23 DMA_CFG_TRIGTYPE_EDGE . . . . .	125
6.17.2.24 DMA_CFG_TRIGTYPE_LEVEL . . . . .	125
6.17.2.25 DMA_CTLSTAT_TRIG . . . . .	125
6.17.2.26 DMA_CTLSTAT_VALIDPENDING . . . . .	125
6.17.2.27 DMA_INTSTAT_ACTIVEERRINT . . . . .	125
6.17.2.28 DMA_INTSTAT_ACTIVEINT . . . . .	125
6.17.2.29 DMA_XFERCFG_CFGVALID . . . . .	125
6.17.2.30 DMA_XFERCFG_CLRTRIG . . . . .	125
6.17.2.31 DMA_XFERCFG_DSTINC_0 . . . . .	126
6.17.2.32 DMA_XFERCFG_DSTINC_1 . . . . .	126
6.17.2.33 DMA_XFERCFG_DSTINC_2 . . . . .	126
6.17.2.34 DMA_XFERCFG_DSTINC_4 . . . . .	126
6.17.2.35 DMA_XFERCFG_RELOAD . . . . .	126
6.17.2.36 DMA_XFERCFG_SETINTA . . . . .	126
6.17.2.37 DMA_XFERCFG_SETINTB . . . . .	126
6.17.2.38 DMA_XFERCFG_SRCINC_0 . . . . .	126
6.17.2.39 DMA_XFERCFG_SRCINC_1 . . . . .	126
6.17.2.40 DMA_XFERCFG_SRCINC_2 . . . . .	127
6.17.2.41 DMA_XFERCFG_SRCINC_4 . . . . .	127
6.17.2.42 DMA_XFERCFG_SWTRIG . . . . .	127
6.17.2.43 DMA_XFERCFG_WIDTH_16 . . . . .	127
6.17.2.44 DMA_XFERCFG_WIDTH_32 . . . . .	127
6.17.2.45 DMA_XFERCFG_WIDTH_8 . . . . .	127
6.17.2.46 DMA_XFERCFG_XFERCOUNT . . . . .	127

6.17.2.47 MAX_DMA_CHANNEL . . . . .	127
6.17.3 Enumeration Type Documentation . . . . .	127
6.17.3.1 DMA_CHID_T . . . . .	127
6.18 CHIP: LPC5411X DMA Service driver . . . . .	129
6.18.1 Detailed Description . . . . .	129
6.18.2 Typedef Documentation . . . . .	129
6.18.2.1 DMA_CALLBACK_T . . . . .	129
6.18.3 Function Documentation . . . . .	129
6.18.3.1 Chip_DMASERVICE_DoubleBuffer(const DMA_PERIPHERAL_CONTEXT_T *pContext, uint32_t pMem, uint32_t length, DMA_DUAL_DESCRIPTOR_T *pD)	129
6.18.3.2 Chip_DMASERVICE_Init(DMA_CHDESC_T *base) . . . . .	130
6.18.3.3 Chip_DMASERVICE_Isr(void) . . . . .	130
6.18.3.4 Chip_DMASERVICE_RegisterCb(const DMA_PERIPHERAL_CONTEXT_T *p↵ Context, DMA_CALLBACK_T pCallback) . . . . .	130
6.18.3.5 Chip_DMASERVICE_SingleBuffer(const DMA_PERIPHERAL_CONTEXT_↵ T *pContext, uint32_t pMem, uint32_t length) . . . . .	131
6.18.4 Variable Documentation . . . . .	132
6.18.4.1 channel . . . . .	132
6.18.4.2 descr . . . . .	132
6.18.4.3 dst_increment . . . . .	132
6.18.4.4 register_location . . . . .	132
6.18.4.5 src_increment . . . . .	132
6.18.4.6 width . . . . .	132
6.18.4.7 write . . . . .	132
6.19 CHIP: LPC5411X DMIC driver . . . . .	133
6.19.1 Detailed Description . . . . .	133
6.19.2 Macro Definition Documentation . . . . .	135
6.19.2.1 DMIC_DCGAIN_REDUCE_P . . . . .	135
6.19.2.2 DMIC_DCPOLE_P . . . . .	135
6.19.2.3 DMIC_FIFO_DMAEN . . . . .	135
6.19.2.4 DMIC_FIFO_DMAEN_P . . . . .	135
6.19.2.5 DMIC_FIFO_ENABLE . . . . .	135
6.19.2.6 DMIC_FIFO_ENABLE_P . . . . .	135
6.19.2.7 DMIC_FIFO_INT . . . . .	135
6.19.2.8 DMIC_FIFO_INT_P . . . . .	136
6.19.2.9 DMIC_FIFO_INTREN . . . . .	136
6.19.2.10 DMIC_FIFO_INTREN_P . . . . .	136
6.19.2.11 DMIC_FIFO_OVERRUN . . . . .	136
6.19.2.12 DMIC_FIFO_OVERRUN_P . . . . .	136
6.19.2.13 DMIC_FIFO_RESETN . . . . .	136
6.19.2.14 DMIC_FIFO_RESETN_P . . . . .	136

6.19.2.15	DMIC_FIFO_TLVL_P	136
6.19.2.16	DMIC_FIFO_UNDERRUN	136
6.19.2.17	DMIC_FIFO_UNDERRUN_P	136
6.19.2.18	DMIC_PHY_FALL	136
6.19.2.19	DMIC_PHY_FALL_P	136
6.19.2.20	DMIC_PHY_HALF	137
6.19.2.21	DMIC_PHY_HALF_P	137
6.19.2.22	DMIC_SATURATE_AT16BIT_P	137
6.19.3	Enumeration Type Documentation	137
6.19.3.1	COMPENSATION_T	137
6.19.3.2	DC_REMOVAL_T	137
6.19.3.3	DMIC_IO_T	137
6.19.3.4	OP_MODE_T	138
6.19.3.5	PDM_DIV_T	138
6.19.3.6	STEREO_SIDE_T	138
6.19.4	Function Documentation	138
6.19.4.1	Chip_DMIC_CfgChannel(LPC_DMIC_T *pDMIC, uint32_t channel, DMIC_CHAN↔ NEL_CONFIG_T *channel_cfg)	138
6.19.4.2	Chip_DMIC_CfgChannelDc(LPC_DMIC_T *pDMIC, uint32_t channel, DC_RE↔ MOVAL_T dc_cut_level, uint32_t post_dc_gain_reduce, bool saturate16bit)	139
6.19.4.3	Chip_DMIC_CfgIO(LPC_DMIC_T *pDMIC, DMIC_IO_T cfg)	139
6.19.4.4	Chip_DMIC_EnableChannel(LPC_DMIC_T *pDMIC, uint32_t channelmask)	139
6.19.4.5	Chip_DMIC_FifoChannel(LPC_DMIC_T *pDMIC, uint32_t channel, uint32_t trig_level, uint32_t enable, uint32_t resetn)	140
6.19.4.6	Chip_DMIC_FifoClearStatus(LPC_DMIC_T *pDMIC, uint32_t channel, uint32_t mask)	141
6.19.4.7	Chip_DMIC_FifoGetData(LPC_DMIC_T *pDMIC, uint32_t channel)	141
6.19.4.8	Chip_DMIC_FifoGetStatus(LPC_DMIC_T *pDMIC, uint32_t channel)	141
6.19.4.9	Chip_DMIC_Init(const CHIP_SYSCON_CLOCK_T clock, const CHIP_SYSCO↔ N_PERIPH_RESET_T reset)	142
6.19.4.10	Chip_DMIC_SetOpMode(LPC_DMIC_T *pDMIC, OP_MODE_T mode)	143
6.19.4.11	Chip_DMIC_Use2fs(LPC_DMIC_T *pDMIC, bool use2fs)	143
6.19.5	Variable Documentation	143
6.19.5.1	CHANEN	143
6.19.5.2	CHANNEL	143
6.19.5.3	DC_CTRL	143
6.19.5.4	DIVHFCLK	144
6.19.5.5	divhfclk	144
6.19.5.6	dmic_ch0_dma_context	144
6.19.5.7	dmic_ch0_dma_interleaved_context	144
6.19.5.8	dmic_ch1_dma_context	144



6.19.5.9	dmic_ch1_dma_interleaved_context	144
6.19.5.10	FIFO_CTRL	144
6.19.5.11	FIFO_DATA	144
6.19.5.12	fifo_ints	144
6.19.5.13	fifo_overflow	144
6.19.5.14	FIFO_STATUS	144
6.19.5.15	fifo_underrun	145
6.19.5.16	GAINSHFT	145
6.19.5.17	gainshft	145
6.19.5.18	HWVADGAIN	145
6.19.5.19	HWVADHPFS	145
6.19.5.20	HWVADLOWZ	145
6.19.5.21	HWVADRSTT	145
6.19.5.22	HWVADST10	145
6.19.5.23	HWVADTHGN	145
6.19.5.24	HWVADTHGS	145
6.19.5.25	ID	145
6.19.5.26	IOCFG	145
6.19.5.27	OSR	146
6.19.5.28	osr	146
6.19.5.29	PHY_CTRL	146
6.19.5.30	preac2coef	146
6.19.5.31	PREAC2FSCOEF	146
6.19.5.32	preac4coef	146
6.19.5.33	PREAC4FSCOEF	146
6.19.5.34	reserved	146
6.19.5.35	reserved	146
6.19.5.36	reserved0	146
6.19.5.37	reserved1	146
6.19.5.38	reserved1	146
6.19.5.39	side	147
6.19.5.40	TDM19EN	147
6.19.5.41	TDM96EN	147
6.19.5.42	USE2FS	147
6.20	CHIP: LPC5411X Enhanced boot block support	148
6.20.1	Detailed Description	148
6.20.2	Macro Definition Documentation	148
6.20.2.1	IMAGE_BOOT_BLOCK_OFF	148
6.20.2.2	IMAGE_DUAL_ENH_SIG	148
6.20.2.3	IMAGE_ENH_BLOCK_MARKER	148

6.20.2.4	IMAGE_ENH_MARKER_OFF . . . . .	148
6.20.2.5	IMAGE_SINGLE_ENH_SIG . . . . .	148
6.20.2.6	SETPORTPIN . . . . .	149
6.20.3	Enumeration Type Documentation . . . . .	149
6.20.3.1	IFSEL_T . . . . .	149
6.20.3.2	IMAGE_T . . . . .	149
6.21	CHIP: LPC5411X GPIO driver . . . . .	150
6.21.1	Detailed Description . . . . .	150
6.21.2	Function Documentation . . . . .	151
6.21.2.1	Chip_GPIO_ClearValue(LPC_GPIO_T *pGPIO, uint8_t portNum, uint32_t bitValue)	151
6.21.2.2	Chip_GPIO_DeInit(LPC_GPIO_T *pGPIO) . . . . .	152
6.21.2.3	Chip_GPIO_GetMaskedPortValue(LPC_GPIO_T *pGPIO, uint8_t port) . . . . .	153
6.21.2.4	Chip_GPIO_GetPinDIR(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin) . . . . .	153
6.21.2.5	Chip_GPIO_GetPinState(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin) . . . . .	153
6.21.2.6	Chip_GPIO_GetPortDIR(LPC_GPIO_T *pGPIO, uint8_t port) . . . . .	154
6.21.2.7	Chip_GPIO_GetPortMask(LPC_GPIO_T *pGPIO, uint8_t port) . . . . .	155
6.21.2.8	Chip_GPIO_GetPortValue(LPC_GPIO_T *pGPIO, uint8_t port) . . . . .	155
6.21.2.9	Chip_GPIO_Init(LPC_GPIO_T *pGPIO) . . . . .	155
6.21.2.10	Chip_GPIO_ReadDirBit(LPC_GPIO_T *pGPIO, uint32_t port, uint8_t bit) . . . . .	156
6.21.2.11	Chip_GPIO_ReadPortBit(LPC_GPIO_T *pGPIO, uint32_t port, uint8_t pin) . . . . .	156
6.21.2.12	Chip_GPIO_ReadValue(LPC_GPIO_T *pGPIO, uint8_t portNum) . . . . .	156
6.21.2.13	Chip_GPIO_SetDir(LPC_GPIO_T *pGPIO, uint8_t portNum, uint32_t bitValue, uint8_t out) . . . . .	157
6.21.2.14	Chip_GPIO_SetMaskedPortValue(LPC_GPIO_T *pGPIO, uint8_t port, uint32_t value) . . . . .	157
6.21.2.15	Chip_GPIO_SetPinDIR(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin, bool out- put) . . . . .	157
6.21.2.16	Chip_GPIO_SetPinDIRInput(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin) . . . . .	158
6.21.2.17	Chip_GPIO_SetPinDIROutput(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin) . . . . .	158
6.21.2.18	Chip_GPIO_SetPinOutHigh(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin) . . . . .	158
6.21.2.19	Chip_GPIO_SetPinOutLow(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin) . . . . .	158
6.21.2.20	Chip_GPIO_SetPinState(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin, bool setting) . . . . .	159
6.21.2.21	Chip_GPIO_SetPinToggle(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin) . . . . .	159
6.21.2.22	Chip_GPIO_SetPortDIR(LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pinMask, bool outSet) . . . . .	159
6.21.2.23	Chip_GPIO_SetPortDIRInput(LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pinMask) . . . . .	160
6.21.2.24	Chip_GPIO_SetPortDIROutput(LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pinMask) . . . . .	160
6.21.2.25	Chip_GPIO_SetPortMask(LPC_GPIO_T *pGPIO, uint8_t port, uint32_t mask) . . . . .	161
6.21.2.26	Chip_GPIO_SetPortOutHigh(LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pins) . . . . .	162

6.21.2.27	Chip_GPIO_SetPortOutLow(LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pins)	162
6.21.2.28	Chip_GPIO_SetPortToggle(LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pins)	162
6.21.2.29	Chip_GPIO_SetPortValue(LPC_GPIO_T *pGPIO, uint8_t port, uint32_t value)	163
6.21.2.30	Chip_GPIO_SetValue(LPC_GPIO_T *pGPIO, uint8_t portNum, uint32_t bitValue)	163
6.21.2.31	Chip_GPIO_WriteDirBit(LPC_GPIO_T *pGPIO, uint32_t port, uint8_t pin, bool setting)	163
6.21.2.32	Chip_GPIO_WritePortBit(LPC_GPIO_T *pGPIO, uint32_t port, uint8_t pin, bool setting)	164
6.22	CHIP: LPC5411X GPIO group driver	165
6.22.1	Detailed Description	165
6.22.2	Macro Definition Documentation	166
6.22.2.1	GPIOGR_COMB	166
6.22.2.2	GPIOGR_INT	166
6.22.2.3	GPIOGR_TRIG	166
6.22.3	Function Documentation	166
6.22.3.1	Chip_GPIOGP_ClearIntStatus(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)	166
6.22.3.2	Chip_GPIOGP_DeInit(LPC_GPIOGROUPINT_T *pGPIOGPINT)	166
6.22.3.3	Chip_GPIOGP_DisableGroupPins(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)	166
6.22.3.4	Chip_GPIOGP_EnableGroupPins(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)	167
6.22.3.5	Chip_GPIOGP_GetIntStatus(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)	167
6.22.3.6	Chip_GPIOGP_Init(LPC_GPIOGROUPINT_T *pGPIOGPINT)	167
6.22.3.7	Chip_GPIOGP_SelectAndMode(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)	168
6.22.3.8	Chip_GPIOGP_SelectEdgeMode(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)	168
6.22.3.9	Chip_GPIOGP_SelectHighLevel(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)	168
6.22.3.10	Chip_GPIOGP_SelectLevelMode(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)	169
6.22.3.11	Chip_GPIOGP_SelectLowLevel(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)	170
6.22.3.12	Chip_GPIOGP_SelectOrMode(LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)	170
6.23	CHIP: LPC5411X I2C master-only driver	171
6.23.1	Detailed Description	171
6.23.2	Macro Definition Documentation	172
6.23.2.1	I2CM_STATUS_ARBLOST	172
6.23.2.2	I2CM_STATUS_BUS_ERROR	172
6.23.2.3	I2CM_STATUS_BUSY	172
6.23.2.4	I2CM_STATUS_ERROR	172

6.23.2.5	I2CM_STATUS_NAK_ADR	172
6.23.2.6	I2CM_STATUS_NAK_DAT	172
6.23.2.7	I2CM_STATUS_OK	172
6.23.3	Function Documentation	173
6.23.3.1	Chip_I2CM_ClearStatus(LPC_I2C_T *pI2C, uint32_t clrStatus)	173
6.23.3.2	Chip_I2CM_Disable(LPC_I2C_T *pI2C)	173
6.23.3.3	Chip_I2CM_Enable(LPC_I2C_T *pI2C)	173
6.23.3.4	Chip_I2CM_GetMasterState(LPC_I2C_T *pI2C)	173
6.23.3.5	Chip_I2CM_GetStatus(LPC_I2C_T *pI2C)	174
6.23.3.6	Chip_I2CM_IsMasterPending(LPC_I2C_T *pI2C)	174
6.23.3.7	Chip_I2CM_MasterContinue(LPC_I2C_T *pI2C)	174
6.23.3.8	Chip_I2CM_ReadByte(LPC_I2C_T *pI2C)	175
6.23.3.9	Chip_I2CM_SendStart(LPC_I2C_T *pI2C)	175
6.23.3.10	Chip_I2CM_SendStop(LPC_I2C_T *pI2C)	175
6.23.3.11	Chip_I2CM_SetBusSpeed(LPC_I2C_T *pI2C, uint32_t busSpeed)	176
6.23.3.12	Chip_I2CM_SetDutyCycle(LPC_I2C_T *pI2C, uint16_t sclH, uint16_t sclL)	176
6.23.3.13	Chip_I2CM_WriteByte(LPC_I2C_T *pI2C, uint8_t data)	177
6.23.3.14	Chip_I2CM_Xfer(LPC_I2C_T *pI2C, I2CM_XFER_T *xfer)	177
6.23.3.15	Chip_I2CM_XferBlocking(LPC_I2C_T *pI2C, I2CM_XFER_T *xfer)	178
6.23.3.16	Chip_I2CM_XferHandler(LPC_I2C_T *pI2C, I2CM_XFER_T *xfer)	178
6.24	CHIP: LPC5411X I2C slave-only driver	179
6.24.1	Detailed Description	179
6.24.2	Typedef Documentation	180
6.24.2.1	I2CSlaveXferDone	180
6.24.2.2	I2CSlaveXferRecv	180
6.24.2.3	I2CSlaveXferSend	180
6.24.2.4	I2CSlaveXferStart	181
6.24.3	Function Documentation	181
6.24.3.1	Chip_I2CS_ClearStatus(LPC_I2C_T *pI2C, uint32_t clrStatus)	181
6.24.3.2	Chip_I2CS_Disable(LPC_I2C_T *pI2C)	181
6.24.3.3	Chip_I2CS_DisableSlaveAddr(LPC_I2C_T *pI2C, uint8_t slvNum)	181
6.24.3.4	Chip_I2CS_Enable(LPC_I2C_T *pI2C)	182
6.24.3.5	Chip_I2CS_EnableSlaveAddr(LPC_I2C_T *pI2C, uint8_t slvNum)	182
6.24.3.6	Chip_I2CS_GetSlaveAddr(LPC_I2C_T *pI2C, uint8_t slvNum)	182
6.24.3.7	Chip_I2CS_GetSlaveMatchIndex(LPC_I2C_T *pI2C)	182
6.24.3.8	Chip_I2CS_GetSlaveState(LPC_I2C_T *pI2C)	183
6.24.3.9	Chip_I2CS_GetStatus(LPC_I2C_T *pI2C)	183
6.24.3.10	Chip_I2CS_IsSlaveDeSelected(LPC_I2C_T *pI2C)	183
6.24.3.11	Chip_I2CS_IsSlavePending(LPC_I2C_T *pI2C)	184
6.24.3.12	Chip_I2CS_IsSlaveSelected(LPC_I2C_T *pI2C)	184

6.24.3.13	Chip_I2CS_ReadByte(LPC_I2C_T *pl2C)	184
6.24.3.14	Chip_I2CS_SetSlaveAddr(LPC_I2C_T *pl2C, uint8_t slvNum, uint8_t slvAddr)	184
6.24.3.15	Chip_I2CS_SetSlaveQual0(LPC_I2C_T *pl2C, bool extend, uint8_t slvAddr)	185
6.24.3.16	Chip_I2CS_SlaveContinue(LPC_I2C_T *pl2C)	185
6.24.3.17	Chip_I2CS_SlaveDisableDMA(LPC_I2C_T *pl2C)	185
6.24.3.18	Chip_I2CS_SlaveEnableDMA(LPC_I2C_T *pl2C)	186
6.24.3.19	Chip_I2CS_SlaveNACK(LPC_I2C_T *pl2C)	186
6.24.3.20	Chip_I2CS_WriteByte(LPC_I2C_T *pl2C, uint8_t data)	186
6.24.3.21	Chip_I2CS_XferHandler(LPC_I2C_T *pl2C, const I2CS_XFER_T *xfers)	187
6.25	CHIP: LPC5411X IOCON register block and driver	188
6.25.1	Detailed Description	188
6.25.2	Macro Definition Documentation	189
6.25.2.1	IOCON_ANALOG_EN	189
6.25.2.2	IOCON_CLKDIV	189
6.25.2.3	IOCON_DIGITAL_EN	189
6.25.2.4	IOCON_FASTI2C_EN	189
6.25.2.5	IOCON_FUNC0	189
6.25.2.6	IOCON_FUNC1	189
6.25.2.7	IOCON_FUNC2	189
6.25.2.8	IOCON_FUNC3	189
6.25.2.9	IOCON_FUNC4	189
6.25.2.10	IOCON_FUNC5	190
6.25.2.11	IOCON_FUNC6	190
6.25.2.12	IOCON_FUNC7	190
6.25.2.13	IOCON_GPIO_MODE	190
6.25.2.14	IOCON_HYS_EN	190
6.25.2.15	IOCON_I2C_SLEW	190
6.25.2.16	IOCON_INPFILT_OFF	190
6.25.2.17	IOCON_INPFILT_ON	190
6.25.2.18	IOCON_INV_EN	190
6.25.2.19	IOCON_MODE_INACT	191
6.25.2.20	IOCON_MODE_PULLDOWN	191
6.25.2.21	IOCON_MODE_PULLUP	191
6.25.2.22	IOCON_MODE_REPEATER	191
6.25.2.23	IOCON_OPENDRAIN_EN	191
6.25.2.24	IOCON_S_MODE	191
6.25.2.25	IOCON_S_MODE_0CLK	191
6.25.2.26	IOCON_S_MODE_1CLK	191
6.25.2.27	IOCON_S_MODE_2CLK	191
6.25.2.28	IOCON_S_MODE_3CLK	192

6.25.2.29	IOCON_STDI2C_EN . . . . .	192
6.25.3	Function Documentation . . . . .	192
6.25.3.1	Chip_IOCON_PinMux(LPC_IOCON_T *pIOCON, uint8_t port, uint8_t pin, uint16_t mode, uint8_t func) . . . . .	192
6.25.3.2	Chip_IOCON_PinMuxSet(LPC_IOCON_T *pIOCON, uint8_t port, uint8_t pin, uint32_t modefunc) . . . . .	192
6.25.3.3	Chip_IOCON_SetPinMuxing(LPC_IOCON_T *pIOCON, const PINMUX_GRP↵ _T *pinArray, uint32_t arrayLength) . . . . .	192
6.26	CHIP: LPC5411X Input Mux Registers and Driver . . . . .	194
6.26.1	Detailed Description . . . . .	194
6.26.2	Enumeration Type Documentation . . . . .	194
6.26.2.1	DMA_TRIGSRC_T . . . . .	194
6.26.2.2	FREQMSR_SRC_T . . . . .	195
6.26.3	Function Documentation . . . . .	195
6.26.3.1	Chip_INMUX_PinIntSel(uint8_t pintSel, uint8_t portNum, uint8_t pinNum) . . . . .	195
6.26.3.2	Chip_INMUX_SetDMAOutMux(uint8_t index, uint8_t dmaCh) . . . . .	195
6.26.3.3	Chip_INMUX_SetDMATrigger(uint8_t ch, DMA_TRIGSRC_T trig) . . . . .	196
6.26.3.4	Chip_INMUX_SetFreqMeasRefClock(FREQMSR_SRC_T ref) . . . . .	196
6.26.3.5	Chip_INMUX_SetFreqMeasTargClock(FREQMSR_SRC_T targ) . . . . .	196
6.27	CHIP: LPC5411X M0 core CMSIS include file . . . . .	197
6.27.1	Detailed Description . . . . .	197
6.28	CHIP: LPC5411X M0 core Cortex CMSIS definitions . . . . .	198
6.28.1	Detailed Description . . . . .	198
6.28.2	Macro Definition Documentation . . . . .	198
6.28.2.1	__CM0PLUS_REV . . . . .	198
6.28.2.2	__MPU_PRESENT . . . . .	198
6.28.2.3	__NVIC_PRIO_BITS . . . . .	198
6.28.2.4	__Vendor_SysTickConfig . . . . .	199
6.28.2.5	__VTOR_PRESENT . . . . .	199
6.28.2.6	ADC0_SEQA_IRQHandler . . . . .	199
6.28.2.7	ADC0_SEQA_IRQn . . . . .	199
6.28.2.8	ADC0_SEQB_IRQHandler . . . . .	199
6.28.2.9	ADC0_SEQB_IRQn . . . . .	199
6.28.2.10	ADC0_THCMP_IRQHandler . . . . .	199
6.28.2.11	ADC0_THCMP_IRQn . . . . .	199
6.28.2.12	SCT_IRQHandler . . . . .	199
6.28.2.13	SCT_IRQn . . . . .	199
6.28.2.14	TIMER0_IRQHandler . . . . .	199
6.28.2.15	TIMER0_IRQn . . . . .	200
6.28.2.16	TIMER1_IRQHandler . . . . .	200
6.28.2.17	TIMER1_IRQn . . . . .	200

6.28.2.18	TIMER2_IRQHandler	200
6.28.2.19	TIMER2_IRQn	200
6.28.2.20	TIMER3_IRQHandler	200
6.28.2.21	TIMER3_IRQn	200
6.28.2.22	TIMER4_IRQHandler	200
6.28.2.23	TIMER4_IRQn	200
6.29	CHIP: LPC5411X M4 core CMSIS include file	201
6.29.1	Detailed Description	201
6.30	CHIP: LPC5411X M4 core Cortex CMSIS definitions	202
6.30.1	Detailed Description	202
6.30.2	Macro Definition Documentation	202
6.30.2.1	__CM4_REV	202
6.30.2.2	__FPU_PRESENT	202
6.30.2.3	__MPU_PRESENT	202
6.30.2.4	__NVIC_PRIO_BITS	202
6.30.2.5	__Vendor_SysTickConfig	203
6.30.2.6	ADC0_SEQA_IRQHandler	203
6.30.2.7	ADC0_SEQA_IRQn	203
6.30.2.8	ADC0_SEQB_IRQHandler	203
6.30.2.9	ADC0_SEQB_IRQn	203
6.30.2.10	ADC0_THCMP_IRQHandler	203
6.30.2.11	ADC0_THCMP_IRQn	203
6.30.2.12	SCT_IRQHandler	203
6.30.2.13	SCT_IRQn	203
6.30.2.14	TIMER0_IRQHandler	203
6.30.2.15	TIMER0_IRQn	203
6.30.2.16	TIMER1_IRQHandler	204
6.30.2.17	TIMER1_IRQn	204
6.30.2.18	TIMER2_IRQHandler	204
6.30.2.19	TIMER2_IRQn	204
6.30.2.20	TIMER3_IRQHandler	204
6.30.2.21	TIMER3_IRQn	204
6.30.2.22	TIMER4_IRQHandler	204
6.30.2.23	TIMER4_IRQn	204
6.31	CHIP: LPC5411X Mailbox M4/M0+ driver	205
6.31.1	Detailed Description	205
6.31.2	Macro Definition Documentation	205
6.31.2.1	MAILBOX_AVAIL	205
6.31.3	Enumeration Type Documentation	205
6.31.3.1	MBOX_IDX_T	205

6.31.4	Function Documentation	206
6.31.4.1	Chip_MBOX_ClearValueBits(LPC_MBOX_T *pMBOX, uint32_t cpu_id, uint32_t mboxClrBits)	206
6.31.4.2	Chip_MBOX_DeInit(LPC_MBOX_T *pMBOX)	206
6.31.4.3	Chip_MBOX_GetMutex(LPC_MBOX_T *pMBOX)	206
6.31.4.4	Chip_MBOX_GetValue(LPC_MBOX_T *pMBOX, uint32_t cpu_id)	207
6.31.4.5	Chip_MBOX_Init(LPC_MBOX_T *pMBOX)	208
6.31.4.6	Chip_MBOX_SetMutex(LPC_MBOX_T *pMBOX)	208
6.31.4.7	Chip_MBOX_SetValue(LPC_MBOX_T *pMBOX, uint32_t cpu_id, uint32_t mboxData)	208
6.31.4.8	Chip_MBOX_SetValueBits(LPC_MBOX_T *pMBOX, uint32_t cpu_id, uint32_t mboxSetBits)	209
6.32	CHIP: LPC5411X Micro Tick driver	210
6.32.1	Detailed Description	210
6.32.2	Macro Definition Documentation	210
6.32.2.1	UTICK_CTRL_DELAY_MASK	210
6.32.2.2	UTICK_CTRL_REPEAT	210
6.32.2.3	UTICK_STATUS_ACTIVE	211
6.32.2.4	UTICK_STATUS_INTR	211
6.32.2.5	UTICK_STATUS_MASK	211
6.32.3	Function Documentation	211
6.32.3.1	Chip_UTICK_ClearInterrupt(LPC_UTICK_T *pUTICK)	211
6.32.3.2	Chip_UTICK_DeInit(LPC_UTICK_T *pUTICK)	211
6.32.3.3	Chip_UTICK_GetStatus(LPC_UTICK_T *pUTICK)	211
6.32.3.4	Chip_UTICK_GetTick(LPC_UTICK_T *pUTICK)	212
6.32.3.5	Chip_UTICK_Halt(LPC_UTICK_T *pUTICK)	212
6.32.3.6	Chip_UTICK_Init(LPC_UTICK_T *pUTICK)	212
6.32.3.7	Chip_UTICK_SetDelayMs(LPC_UTICK_T *pUTICK, uint32_t delayMs, bool repeat)	212
6.32.3.8	Chip_UTICK_SetTick(LPC_UTICK_T *pUTICK, uint32_t tick_value, bool repeat)	213
6.33	CHIP: LPC5411X Multi-Rate Timer driver	214
6.33.1	Detailed Description	214
6.33.2	Macro Definition Documentation	215
6.33.2.1	LPC_MRT_CH	215
6.33.2.2	LPC_MRT_CH0	215
6.33.2.3	LPC_MRT_CH1	215
6.33.2.4	LPC_MRT_CH2	215
6.33.2.5	LPC_MRT_CH3	216
6.33.2.6	MRT0_INTFLAG	216
6.33.2.7	MRT1_INTFLAG	216
6.33.2.8	MRT2_INTFLAG	216
6.33.2.9	MRT3_INTFLAG	216



6.33.2.10	MRT_CHANNELS_NUM	216
6.33.2.11	MRT_CTRL_INTEN_MASK	216
6.33.2.12	MRT_CTRL_MODE_MASK	216
6.33.2.13	MRT_INTVAL_IVALUE	216
6.33.2.14	MRT_INTVAL_LOAD	216
6.33.2.15	MRT_NO_IDLE_CHANNEL	216
6.33.2.16	MRT_STAT_INTFLAG	216
6.33.2.17	MRT_STAT_RUNNING	217
6.33.2.18	MRTn_INTFLAG	217
6.33.3	Enumeration Type Documentation	217
6.33.3.1	MRT_MODE_T	217
6.33.4	Function Documentation	217
6.33.4.1	Chip_MRT_ClearIntPending(uint32_t mask)	217
6.33.4.2	Chip_MRT_DeInit(void)	217
6.33.4.3	Chip_MRT_GetEnabled(LPC_MRT_CH_T *pMRT)	217
6.33.4.4	Chip_MRT_GetIdleChannel(void)	218
6.33.4.5	Chip_MRT_GetIdleChannelShifted(void)	218
6.33.4.6	Chip_MRT_GetInterval(LPC_MRT_CH_T *pMRT)	218
6.33.4.7	Chip_MRT_GetIntPending(void)	218
6.33.4.8	Chip_MRT_GetIntPendingByChannel(uint8_t ch)	218
6.33.4.9	Chip_MRT_GetMode(LPC_MRT_CH_T *pMRT)	219
6.33.4.10	Chip_MRT_GetRegPtr(uint8_t ch)	219
6.33.4.11	Chip_MRT_GetTimer(LPC_MRT_CH_T *pMRT)	219
6.33.4.12	Chip_MRT_Init(void)	219
6.33.4.13	Chip_MRT_IntClear(LPC_MRT_CH_T *pMRT)	220
6.33.4.14	Chip_MRT_IntPending(LPC_MRT_CH_T *pMRT)	220
6.33.4.15	Chip_MRT_IsOneShotMode(LPC_MRT_CH_T *pMRT)	220
6.33.4.16	Chip_MRT_IsRepeatMode(LPC_MRT_CH_T *pMRT)	220
6.33.4.17	Chip_MRT_Running(LPC_MRT_CH_T *pMRT)	220
6.33.4.18	Chip_MRT_SetDisabled(LPC_MRT_CH_T *pMRT)	221
6.33.4.19	Chip_MRT_SetEnabled(LPC_MRT_CH_T *pMRT)	221
6.33.4.20	Chip_MRT_SetInterval(LPC_MRT_CH_T *pMRT, uint32_t interval)	221
6.33.4.21	Chip_MRT_SetMode(LPC_MRT_CH_T *pMRT, MRT_MODE_T mode)	222
6.34	CHIP: LPC5411X PLL Driver	224
6.34.1	Detailed Description	224
6.34.2	Macro Definition Documentation	226
6.34.2.1	PLL_CONFIGFLAG_FORCENOFRACT	226
6.34.2.2	PLL_CONFIGFLAG_USEINRATE	226
6.34.2.3	PLL_SETUPFLAG_ADGVOLT	226
6.34.2.4	PLL_SETUPFLAG_POWERUP	226

6.34.2.5	PLL_SETUPFLAG_WAITLOCK	227
6.34.3	Enumeration Type Documentation	227
6.34.3.1	CHIP_SYSCON_PLLCLKSRC_T	227
6.34.3.2	PLL_ERROR_T	227
6.34.3.3	SS_MODWVCTRL_T	227
6.34.3.4	SS_PROGMODDP_T	228
6.34.3.5	SS_PROGMODFM_T	228
6.34.4	Function Documentation	228
6.34.4.1	Chip_Clock_GetStoredPLLClockRate(void)	228
6.34.4.2	Chip_Clock_GetSystemPLLInClockRate(void)	228
6.34.4.3	Chip_Clock_GetSystemPLLOutClockRate(bool recompute)	229
6.34.4.4	Chip_Clock_GetSystemPLLOutFromSetup(PLL_SETUP_T *pSetup)	229
6.34.4.5	Chip_Clock_IsSystemPLLLocked(void)	229
6.34.4.6	Chip_Clock_SetBypassPLL(bool bypass)	229
6.34.4.7	Chip_Clock_SetPLLFreq(const PLL_SETUP_T *pSetup)	229
6.34.4.8	Chip_Clock_SetStoredPLLClockRate(uint32_t rate)	230
6.34.4.9	Chip_Clock_SetSystemPLLSource(CHIP_SYSCON_PLLCLKSRC_T src)	230
6.34.4.10	Chip_Clock_SetupPLLData(PLL_CONFIG_T *pControl, PLL_SETUP_T *pSetup)	230
6.34.4.11	Chip_Clock_SetupSystemPLL(uint32_t multiply_by, uint32_t input_freq)	231
6.34.4.12	Chip_Clock_SetupSystemPLLPrec(PLL_SETUP_T *pSetup)	231
6.35	CHIP: LPC5411X Peripheral addresses and register set declarations	232
6.35.1	Detailed Description	232
6.35.2	Macro Definition Documentation	233
6.35.2.1	LPC_ADC	233
6.35.2.2	LPC_ADC_BASE	233
6.35.2.3	LPC_ASYNC_SYSCON	233
6.35.2.4	LPC_ASYNC_SYSCON_BASE	233
6.35.2.5	LPC_CRC	233
6.35.2.6	LPC_CRC_BASE	233
6.35.2.7	LPC_DMA	233
6.35.2.8	LPC_DMA_BASE	234
6.35.2.9	LPC_DMIC	234
6.35.2.10	LPC_DMIC_BASE	234
6.35.2.11	LPC_FLASHMEM_BASE	234
6.35.2.12	LPC_FLEXCOMM0_BASE	234
6.35.2.13	LPC_FLEXCOMM1_BASE	234
6.35.2.14	LPC_FLEXCOMM2_BASE	234
6.35.2.15	LPC_FLEXCOMM3_BASE	234
6.35.2.16	LPC_FLEXCOMM4_BASE	234
6.35.2.17	LPC_FLEXCOMM5_BASE	234

6.35.2.18 LPC_FLEXCOMM6_BASE . . . . .	234
6.35.2.19 LPC_FLEXCOMM7_BASE . . . . .	234
6.35.2.20 LPC_FMC_BASE . . . . .	235
6.35.2.21 LPC_GINT . . . . .	235
6.35.2.22 LPC_GPIO . . . . .	235
6.35.2.23 LPC_GPIO_GROUPINT0_BASE . . . . .	235
6.35.2.24 LPC_GPIO_GROUPINT1_BASE . . . . .	235
6.35.2.25 LPC_GPIO_PORT_BASE . . . . .	235
6.35.2.26 LPC_INMUX . . . . .	235
6.35.2.27 LPC_INMUX_BASE . . . . .	235
6.35.2.28 LPC_IOCON . . . . .	235
6.35.2.29 LPC_IOCON_BASE . . . . .	235
6.35.2.30 LPC_ISPAP_BASE . . . . .	235
6.35.2.31 LPC_MBOX . . . . .	235
6.35.2.32 LPC_MBOX_BASE . . . . .	236
6.35.2.33 LPC_MRT . . . . .	236
6.35.2.34 LPC_MRT_BASE . . . . .	236
6.35.2.35 LPC_PIN_INT_BASE . . . . .	236
6.35.2.36 LPC_PININT . . . . .	236
6.35.2.37 LPC_PMU . . . . .	236
6.35.2.38 LPC_PMU_BASE . . . . .	236
6.35.2.39 LPC_ROM_BASE . . . . .	236
6.35.2.40 LPC_RTC . . . . .	236
6.35.2.41 LPC_RTC_BASE . . . . .	236
6.35.2.42 LPC_SCT . . . . .	236
6.35.2.43 LPC_SCT_BASE . . . . .	236
6.35.2.44 LPC_SPIFI_BASE . . . . .	237
6.35.2.45 LPC_SRAM0_BASE . . . . .	237
6.35.2.46 LPC_SRAM1_BASE . . . . .	237
6.35.2.47 LPC_SRAM2_BASE . . . . .	237
6.35.2.48 LPC_SRAMX_BASE . . . . .	237
6.35.2.49 LPC_SYSCON . . . . .	237
6.35.2.50 LPC_SYSCON_BASE . . . . .	237
6.35.2.51 LPC_TIMER0 . . . . .	237
6.35.2.52 LPC_TIMER0_BASE . . . . .	237
6.35.2.53 LPC_TIMER1 . . . . .	237
6.35.2.54 LPC_TIMER1_BASE . . . . .	237
6.35.2.55 LPC_TIMER2 . . . . .	237
6.35.2.56 LPC_TIMER2_BASE . . . . .	238
6.35.2.57 LPC_TIMER3 . . . . .	238

6.35.2.58	LPC_TIMER3_BASE	238
6.35.2.59	LPC_TIMER4	238
6.35.2.60	LPC_TIMER4_BASE	238
6.35.2.61	LPC_USB	238
6.35.2.62	LPC_USB_BASE	238
6.35.2.63	LPC_UTICK	238
6.35.2.64	LPC_UTICK_BASE	238
6.35.2.65	LPC_WWDT	238
6.35.2.66	LPC_WWDT_BASE	238
6.36	CHIP: LPC5411X Pin Interrupt and Pattern Match driver	239
6.36.1	Detailed Description	239
6.36.2	Macro Definition Documentation	240
6.36.2.1	PININT_ISEL_PMODE_MASK	240
6.36.2.2	PININT_PMCTRL_MASK	241
6.36.2.3	PININT_PMCTRL_PMATCH_SEL	241
6.36.2.4	PININT_PMCTRL_RXEV_ENA	241
6.36.2.5	PININT_SRC_BITCFG_MASK	241
6.36.2.6	PININT_SRC_BITCFG_START	241
6.36.2.7	PININT_SRC_BITSOURCE_MASK	241
6.36.2.8	PININT_SRC_BITSOURCE_START	241
6.36.2.9	PININTCH	241
6.36.2.10	PININTCH0	241
6.36.2.11	PININTCH1	241
6.36.2.12	PININTCH2	241
6.36.2.13	PININTCH3	241
6.36.2.14	PININTCH4	242
6.36.2.15	PININTCH5	242
6.36.2.16	PININTCH6	242
6.36.2.17	PININTCH7	242
6.36.3	Enumeration Type Documentation	242
6.36.3.1	Chip_PININT_BITSLICE_CFG_T	242
6.36.3.2	Chip_PININT_BITSLICE_T	242
6.36.3.3	Chip_PININT_SELECT_T	243
6.36.4	Function Documentation	243
6.36.4.1	Chip_PININT_ClearFallStates(LPC_PIN_INT_T *pPININT, uint32_t pins)	243
6.36.4.2	Chip_PININT_ClearIntStatus(LPC_PIN_INT_T *pPININT, uint32_t pins)	243
6.36.4.3	Chip_PININT_ClearRiseStates(LPC_PIN_INT_T *pPININT, uint32_t pins)	243
6.36.4.4	Chip_PININT_DeInit(LPC_PIN_INT_T *pPININT)	244
6.36.4.5	Chip_PININT_DisableIntHigh(LPC_PIN_INT_T *pPININT, uint32_t pins)	244
6.36.4.6	Chip_PININT_DisableIntLow(LPC_PIN_INT_T *pPININT, uint32_t pins)	244

6.36.4.7	Chip_PININT_DisablePatternMatch(LPC_PIN_INT_T *pPININT)	244
6.36.4.8	Chip_PININT_DisablePatternMatchRxEv(LPC_PIN_INT_T *pPININT)	245
6.36.4.9	Chip_PININT_EnableIntHigh(LPC_PIN_INT_T *pPININT, uint32_t pins)	245
6.36.4.10	Chip_PININT_EnableIntLow(LPC_PIN_INT_T *pPININT, uint32_t pins)	245
6.36.4.11	Chip_PININT_EnablePatternMatch(LPC_PIN_INT_T *pPININT)	245
6.36.4.12	Chip_PININT_EnablePatternMatchRxEv(LPC_PIN_INT_T *pPININT)	246
6.36.4.13	Chip_PININT_GetFallStates(LPC_PIN_INT_T *pPININT)	246
6.36.4.14	Chip_PININT_GetHighEnabled(LPC_PIN_INT_T *pPININT)	246
6.36.4.15	Chip_PININT_GetIntStatus(LPC_PIN_INT_T *pPININT)	246
6.36.4.16	Chip_PININT_GetLowEnabled(LPC_PIN_INT_T *pPININT)	247
6.36.4.17	Chip_PININT_GetPatternMatchState(LPC_PIN_INT_T *pPININT)	247
6.36.4.18	Chip_PININT_GetPinMode(LPC_PIN_INT_T *pPININT)	247
6.36.4.19	Chip_PININT_GetRiseStates(LPC_PIN_INT_T *pPININT)	247
6.36.4.20	Chip_PININT_Init(LPC_PIN_INT_T *pPININT)	248
6.36.4.21	Chip_PININT_SetPatternMatchConfig(LPC_PIN_INT_T *pPININT, Chip_PININT_BITSLICE_T sliceNum, Chip_PININT_BITSLICE_CFG_T slice_cfg, bool end_point)	248
6.36.4.22	Chip_PININT_SetPatternMatchSrc(LPC_PIN_INT_T *pPININT, Chip_PININT_SELECT_T channelNum, Chip_PININT_BITSLICE_T sliceNum)	248
6.36.4.23	Chip_PININT_SetPinModeEdge(LPC_PIN_INT_T *pPININT, uint32_t pins)	249
6.36.4.24	Chip_PININT_SetPinModeLevel(LPC_PIN_INT_T *pPININT, uint32_t pins)	249
6.37	CHIP: LPC5411X Power LIBRARY functions	250
6.37.1	Detailed Description	250
6.37.2	Macro Definition Documentation	251
6.37.2.1	LPC5411X_ROMVER_0	251
6.37.2.2	LPC5411X_ROMVER_1	251
6.37.2.3	LPC5411X_ROMVER_2	251
6.37.3	Enumeration Type Documentation	251
6.37.3.1	POWER_MODE_T	251
6.37.4	Function Documentation	252
6.37.4.1	Chip_POWER_EnterPowerMode(POWER_MODE_T mode, uint32_t peripheral_ctrl)	252
6.37.4.2	Chip_POWER_GetROMVersion(void)	252
6.37.4.3	Chip_POWER_SetFROHFRate(uint32_t freq)	252
6.37.4.4	Chip_POWER_SetLowPowerVoltage(uint32_t freq)	252
6.37.4.5	Chip_POWER_SetPLL(uint32_t multiply_by, uint32_t input_freq)	253
6.37.4.6	Chip_POWER_SetVoltage(uint32_t desired_freq)	253
6.38	CHIP: LPC5411X Power Management declarations and functions	254
6.38.1	Detailed Description	254
6.38.2	Macro Definition Documentation	254
6.38.2.1	PMU_BOD_INT	254

6.38.2.2	PMU_BOD_RST	254
6.38.3	Enumeration Type Documentation	255
6.38.3.1	CHIP_PMU_BODRINTVAL_T	255
6.38.3.2	CHIP_PMU_BODRSTLVL_T	255
6.38.4	Function Documentation	255
6.38.4.1	Chip_PMU_DisableBODInt(void)	255
6.38.4.2	Chip_PMU_DisableBODReset(void)	255
6.38.4.3	Chip_PMU_EnableBODInt(void)	256
6.38.4.4	Chip_PMU_EnableBODReset(void)	256
6.38.4.5	Chip_PMU_SetBODLevels(CHIP_PMU_BODRSTLVL_T rstlvl, CHIP_PMU_BODRINTVAL_T intlvl)	256
6.39	CHIP: LPC5411X ROM API declarations and functions	257
6.39.1	Detailed Description	257
6.39.2	Macro Definition Documentation	257
6.39.2.1	IAP_ENTRY_LOCATION	257
6.39.2.2	LPC_ROM_API	257
6.39.2.3	LPC_ROM_API_BASE_LOC	257
6.39.3	Function Documentation	257
6.39.3.1	iap_entry(unsigned int cmd_param[5], unsigned int status_result[4])	257
6.40	CHIP: LPC5411X Real Time clock	258
6.40.1	Detailed Description	258
6.40.2	Macro Definition Documentation	259
6.40.2.1	RTC_CTRL_ALARM1HZ	259
6.40.2.2	RTC_CTRL_ALARMDPD_EN	259
6.40.2.3	RTC_CTRL_MASK	259
6.40.2.4	RTC_CTRL_RTC1KHZ_EN	259
6.40.2.5	RTC_CTRL_RTC_EN	259
6.40.2.6	RTC_CTRL_RTC_OSC_BYPASS	259
6.40.2.7	RTC_CTRL_RTC_OSC_PD	260
6.40.2.8	RTC_CTRL_SWRESET	260
6.40.2.9	RTC_CTRL_WAKE1KHZ	260
6.40.2.10	RTC_CTRL_WAKEDPD_EN	260
6.40.3	Function Documentation	260
6.40.3.1	Chip_RTC_ClearStatus(LPC_RTC_T *pRTC, uint32_t stsMask)	260
6.40.3.2	Chip_RTC_DeInit(LPC_RTC_T *pRTC)	260
6.40.3.3	Chip_RTC_Disable(LPC_RTC_T *pRTC)	261
6.40.3.4	Chip_RTC_Disable1KHZ(LPC_RTC_T *pRTC)	261
6.40.3.5	Chip_RTC_DisableOptions(LPC_RTC_T *pRTC, uint32_t flags)	261
6.40.3.6	Chip_RTC_DisableWakeup(LPC_RTC_T *pRTC, uint32_t ints)	261
6.40.3.7	Chip_RTC_Enable(LPC_RTC_T *pRTC)	262

6.40.3.8	Chip_RTC_Enable1KHZ(LPC_RTC_T *pRTC)	262
6.40.3.9	Chip_RTC_EnableOptions(LPC_RTC_T *pRTC, uint32_t flags)	262
6.40.3.10	Chip_RTC_EnableWakeup(LPC_RTC_T *pRTC, uint32_t ints)	263
6.40.3.11	Chip_RTC_GetAlarm(LPC_RTC_T *pRTC)	263
6.40.3.12	Chip_RTC_GetCount(LPC_RTC_T *pRTC)	263
6.40.3.13	Chip_RTC_GetStatus(LPC_RTC_T *pRTC)	264
6.40.3.14	Chip_RTC_GetWake(LPC_RTC_T *pRTC)	264
6.40.3.15	Chip_RTC_Init(LPC_RTC_T *pRTC)	264
6.40.3.16	Chip_RTC_PowerDown(LPC_RTC_T *pRTC)	264
6.40.3.17	Chip_RTC_PowerUp(LPC_RTC_T *pRTC)	265
6.40.3.18	Chip_RTC_Reset(LPC_RTC_T *pRTC)	265
6.40.3.19	Chip_RTC_SetAlarm(LPC_RTC_T *pRTC, uint32_t count)	265
6.40.3.20	Chip_RTC_SetCount(LPC_RTC_T *pRTC, uint32_t count)	266
6.40.3.21	Chip_RTC_SetWake(LPC_RTC_T *pRTC, uint16_t count)	267
6.41	CHIP: LPC5411X SPI driver	268
6.41.1	Detailed Description	268
6.41.2	Macro Definition Documentation	272
6.41.2.1	Chip_SPI_FlushFIFOs	272
6.41.2.2	Chip_SPI_ReadFIFO	272
6.41.2.3	Chip_SPI_ReadFIFOdata	272
6.41.2.4	Chip_SPI_WriteFIFOcd	272
6.41.2.5	Chip_SPI_WriteFIFOdata	272
6.41.2.6	SPI_CFG_BITMASK	272
6.41.2.7	SPI_CFG_CPHA_FIRST	272
6.41.2.8	SPI_CFG_CPHA_SECOND	272
6.41.2.9	SPI_CFG_CPOL_HI	272
6.41.2.10	SPI_CFG_CPOL_LO	273
6.41.2.11	SPI_CFG_LBM_EN	273
6.41.2.12	SPI_CFG_LSB_FIRST_EN	273
6.41.2.13	SPI_CFG_MASTER_EN	273
6.41.2.14	SPI_CFG_MSB_FIRST_EN	273
6.41.2.15	SPI_CFG_SLAVE_EN	273
6.41.2.16	SPI_CFG_SPI_EN	273
6.41.2.17	SPI_CFG_SPOL_HI	273
6.41.2.18	SPI_CFG_SPOL_LO	273
6.41.2.19	SPI_CFG_SPOLNUM_HI	273
6.41.2.20	SPI_DIV_VAL	273
6.41.2.21	SPI_DLY_BITMASK	273
6.41.2.22	SPI_DLY_FRAME_DELAY	274
6.41.2.23	SPI_DLY_POST_DELAY	274

6.41.2.24 SPI_DLY_PRE_DELAY . . . . .	274
6.41.2.25 SPI_DLY_TRANSFER_DELAY . . . . .	274
6.41.2.26 SPI_FIFO_DEPTH . . . . .	274
6.41.2.27 SPI_FIFOCFG_DMARX . . . . .	274
6.41.2.28 SPI_FIFOCFG_DMATX . . . . .	274
6.41.2.29 SPI_FIFOCFG_EMPTYRX . . . . .	274
6.41.2.30 SPI_FIFOCFG_EMPTYTX . . . . .	274
6.41.2.31 SPI_FIFOCFG_ENABLERX . . . . .	274
6.41.2.32 SPI_FIFOCFG_ENABLETX . . . . .	275
6.41.2.33 SPI_FIFOCFG_WAKERX . . . . .	275
6.41.2.34 SPI_FIFOCFG_WAKETX . . . . .	275
6.41.2.35 SPI_FIFOINT_BITMASK . . . . .	275
6.41.2.36 SPI_FIFOINT_PERINT . . . . .	275
6.41.2.37 SPI_FIFOINT_RXERR . . . . .	275
6.41.2.38 SPI_FIFOINT_RXLVL . . . . .	275
6.41.2.39 SPI_FIFOINT_TXERR . . . . .	275
6.41.2.40 SPI_FIFOINT_TXLVL . . . . .	275
6.41.2.41 SPI_FIFOSTAT_BITMASK . . . . .	275
6.41.2.42 SPI_FIFOSTAT_PERINT . . . . .	276
6.41.2.43 SPI_FIFOSTAT_RXERR . . . . .	276
6.41.2.44 SPI_FIFOSTAT_RXFULL . . . . .	276
6.41.2.45 SPI_FIFOSTAT_RXLVL . . . . .	276
6.41.2.46 SPI_FIFOSTAT_RXNOTEMPTY . . . . .	276
6.41.2.47 SPI_FIFOSTAT_TXEMPTY . . . . .	276
6.41.2.48 SPI_FIFOSTAT_TXERR . . . . .	276
6.41.2.49 SPI_FIFOSTAT_TXLVL . . . . .	276
6.41.2.50 SPI_FIFOSTAT_TXNOTFULL . . . . .	276
6.41.2.51 SPI_FIFOTRIG_BITMASK . . . . .	276
6.41.2.52 SPI_FIFOTRIG_RXLVL . . . . .	276
6.41.2.53 SPI_FIFOTRIG_RXLVL_DEFAULT . . . . .	277
6.41.2.54 SPI_FIFOTRIG_RXLVLENA . . . . .	277
6.41.2.55 SPI_FIFOTRIG_TXLVL . . . . .	277
6.41.2.56 SPI_FIFOTRIG_TXLVL_DEFAULT . . . . .	277
6.41.2.57 SPI_FIFOTRIG_TXLVLENA . . . . .	277
6.41.2.58 SPI_INT_BITMASK . . . . .	277
6.41.2.59 SPI_INT_MSTIDLE . . . . .	277
6.41.2.60 SPI_INT_SSAEN . . . . .	277
6.41.2.61 SPI_INT_SSDEN . . . . .	277
6.41.2.62 SPI_RXDAT_BITMASK . . . . .	277
6.41.2.63 SPI_RXDAT_DATA . . . . .	278



6.41.2.64	SPI_RXDAT_RXSSELN	278
6.41.2.65	SPI_RXDAT_RXSSELN_ACTIVE	278
6.41.2.66	SPI_RXDAT_SOT	278
6.41.2.67	SPI_STAT_BITMASK	278
6.41.2.68	SPI_STAT_EOT	278
6.41.2.69	SPI_STAT_MSTIDLE	278
6.41.2.70	SPI_STAT_SSA	278
6.41.2.71	SPI_STAT_SSD	278
6.41.2.72	SPI_STAT_STALLED	278
6.41.2.73	SPI_TXDAT_ASSERT_SSEL	278
6.41.2.74	SPI_TXDAT_ASSERTNUM_SSEL	278
6.41.2.75	SPI_TXDAT_BITMASK	279
6.41.2.76	SPI_TXDAT_CTRLMASK	279
6.41.2.77	SPI_TXDAT_DATA	279
6.41.2.78	SPI_TXDAT_DATA	279
6.41.2.79	SPI_TXDAT_DEASSERT_ALL	279
6.41.2.80	SPI_TXDAT_DEASSERT_SSEL	279
6.41.2.81	SPI_TXDAT_DEASSERTNUM_SSEL	279
6.41.2.82	SPI_TXDAT_EOF	279
6.41.2.83	SPI_TXDAT_EOT	279
6.41.2.84	SPI_TXDAT_FLEN	279
6.41.2.85	SPI_TXDAT_FLENMASK	279
6.41.2.86	SPI_TXDAT_RXIGNORE	280
6.41.3	Enumeration Type Documentation	280
6.41.3.1	ROM_SPI_CLOCK_MODE_T	280
6.41.3.2	SPI_CLOCK_MODE_T	280
6.41.4	Function Documentation	280
6.41.4.1	Chip_SPI_ClearCFGRegBits(LPC_SPI_T *pSPI, uint32_t bits)	280
6.41.4.2	Chip_SPI_ClearFIFOCfg(LPC_SPI_T *pSPI, uint32_t cfg)	281
6.41.4.3	Chip_SPI_ClearFIFOStatus(LPC_SPI_T *pSPI, uint32_t mask)	281
6.41.4.4	Chip_SPI_ClearStatus(LPC_SPI_T *pSPI, uint32_t Flag)	281
6.41.4.5	Chip_SPI_ConfigureSPI(LPC_SPI_T *pSPI, SPI_CFGSETUP_T *pCFG)	282
6.41.4.6	Chip_SPI_DeInit(LPC_SPI_T *pSPI)	283
6.41.4.7	Chip_SPI_Disable(LPC_SPI_T *pSPI)	283
6.41.4.8	Chip_SPI_DisableFIFOInts(LPC_SPI_T *pSPI, uint32_t intMask)	283
6.41.4.9	Chip_SPI_DisableInts(LPC_SPI_T *pSPI, uint32_t intMask)	283
6.41.4.10	Chip_SPI_Enable(LPC_SPI_T *pSPI)	284
6.41.4.11	Chip_SPI_EnableFIFOInts(LPC_SPI_T *pSPI, uint32_t intMask)	284
6.41.4.12	Chip_SPI_EnableInts(LPC_SPI_T *pSPI, uint32_t intMask)	284
6.41.4.13	Chip_SPI_EnableLSBFirst(LPC_SPI_T *pSPI)	284

6.41.4.14	Chip_SPI_EnableMSBFirst(LPC_SPI_T *pSPI)	285
6.41.4.15	Chip_SPI_EnableSlaveMode(LPC_SPI_T *pSPI)	285
6.41.4.16	Chip_SPI_FlushFifos(LPC_SPI_T *pSPI)	285
6.41.4.17	Chip_SPI_GetEnabledInts(LPC_SPI_T *pSPI)	285
6.41.4.18	Chip_SPI_GetFIFOEnabledInts(LPC_SPI_T *pSPI)	286
6.41.4.19	Chip_SPI_GetFIFOPendingInts(LPC_SPI_T *pSPI)	286
6.41.4.20	Chip_SPI_GetFIFOStatus(LPC_SPI_T *pSPI)	286
6.41.4.21	Chip_SPI_GetFIFOTrigLevel(LPC_SPI_T *pSPI)	287
6.41.4.22	Chip_SPI_GetPendingInts(LPC_SPI_T *pSPI)	288
6.41.4.23	Chip_SPI_GetStatus(LPC_SPI_T *pSPI)	288
6.41.4.24	Chip_SPI_Init(LPC_SPI_T *pSPI)	288
6.41.4.25	Chip_SPI_ReadRawRXFifo(LPC_SPI_T *pSPI)	288
6.41.4.26	Chip_SPI_ReadRXData(LPC_SPI_T *pSPI)	289
6.41.4.27	Chip_SPI_SetCFGRegBits(LPC_SPI_T *pSPI, uint32_t bits)	289
6.41.4.28	Chip_SPI_SetCSPolHigh(LPC_SPI_T *pSPI, uint8_t csNum)	289
6.41.4.29	Chip_SPI_SetCSPolLow(LPC_SPI_T *pSPI, uint8_t csNum)	290
6.41.4.30	Chip_SPI_SetFIFOCfg(LPC_SPI_T *pSPI, uint32_t cfg)	290
6.41.4.31	Chip_SPI_SetFIFOTrigLevel(LPC_SPI_T *pSPI, uint8_t tx_lvl, uint8_t rx_lvl)	290
6.41.4.32	Chip_SPI_SetSPIMode(LPC_SPI_T *pSPI, SPI_CLOCK_MODE_T mode)	291
6.41.4.33	Chip_SPI_SetTXCTRLData(LPC_SPI_T *pSPI, uint16_t ctrl, uint16_t data)	291
6.41.4.34	Chip_SPI_WriteFIFO(LPC_SPI_T *pSPI, uint32_t data)	291
6.41.4.35	Chip_SPI_WriteTXData(LPC_SPI_T *pSPI, uint16_t data)	292
6.42	CHIP: LPC5411X SPI master driver	294
6.42.1	Detailed Description	294
6.42.2	Macro Definition Documentation	295
6.42.2.1	SPIM_XFER_OPT_DMA	295
6.42.2.2	SPIM_XFER_OPT_FRAME_ASSERT	295
6.42.2.3	SPIM_XFER_OPT_FRAME_DLY	295
6.42.3	Enumeration Type Documentation	295
6.42.3.1	SPIM_EVENT_T	295
6.42.3.2	SPIM_XFER_STATE_T	295
6.42.4	Function Documentation	295
6.42.4.1	Chip_SPIM_DelayConfig(LPC_SPI_T *pSPI, SPIM_DELAY_CONFIG_T *pConfig)	295
6.42.4.2	Chip_SPIM_DisableLoopBack(LPC_SPI_T *pSPI)	296
6.42.4.3	Chip_SPIM_EnableLoopBack(LPC_SPI_T *pSPI)	296
6.42.4.4	Chip_SPIM_ForceEndOfTransfer(LPC_SPI_T *pSPI)	296
6.42.4.5	Chip_SPIM_GetClockRate(LPC_SPI_T *pSPI)	297
6.42.4.6	Chip_SPIM_SetClockRate(LPC_SPI_T *pSPI, uint32_t rate)	297
6.42.4.7	Chip_SPIM_Xfer(LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)	297
6.42.4.8	Chip_SPIM_XferBlocking(LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)	298

6.42.4.9	Chip_SPIM_XferFIFO(LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)	298
6.42.4.10	Chip_SPIM_XferHandler(LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)	299
6.43	CHIP: LPC5411X SPI slave driver	300
6.43.1	Detailed Description	300
6.43.2	Enumeration Type Documentation	300
6.43.2.1	SPIS_EVENT_T	300
6.43.3	Function Documentation	300
6.43.3.1	Chip_SPIS_DisableInts(LPC_SPI_T *pSPI)	300
6.43.3.2	Chip_SPIS_EnableInts(LPC_SPI_T *pSPI)	301
6.43.3.3	Chip_SPIS_Init(LPC_SPI_T *pSPI)	301
6.43.3.4	Chip_SPIS_LoadFIFO(LPC_SPI_T *pSPI, SPIS_XFER_T *xfer)	301
6.43.3.5	Chip_SPIS_ReadFIFO(LPC_SPI_T *pSPI, SPIS_XFER_T *xfer)	302
6.43.3.6	Chip_SPIS_XferHandler(LPC_SPI_T *pSPI, SPIS_XFER_T *xfer)	302
6.44	CHIP: LPC5411X State Configurable Timer PWM driver	303
6.44.1	Detailed Description	303
6.44.2	Function Documentation	303
6.44.2.1	Chip_SCTPWM_GetDutyCycle(LPC_SCT_T *pSCT, uint8_t index)	303
6.44.2.2	Chip_SCTPWM_GetTicksPerCycle(LPC_SCT_T *pSCT)	303
6.44.2.3	Chip_SCTPWM_Init(LPC_SCT_T *pSCT)	304
6.44.2.4	Chip_SCTPWM_PercentageToTicks(LPC_SCT_T *pSCT, uint8_t percent)	304
6.44.2.5	Chip_SCTPWM_SetDutyCycle(LPC_SCT_T *pSCT, uint8_t index, uint32_t ticks)	304
6.44.2.6	Chip_SCTPWM_SetOutPin(LPC_SCT_T *pSCT, uint8_t index, uint8_t pin)	305
6.44.2.7	Chip_SCTPWM_SetRate(LPC_SCT_T *pSCT, uint32_t freq)	305
6.44.2.8	Chip_SCTPWM_Start(LPC_SCT_T *pSCT)	305
6.44.2.9	Chip_SCTPWM_Stop(LPC_SCT_T *pSCT)	306
6.45	CHIP: LPC5411X State Configurable Timer driver	307
6.45.1	Detailed Description	307
6.45.2	Macro Definition Documentation	309
6.45.2.1	CONFIG_SCT_nEV	309
6.45.2.2	CONFIG_SCT_nIN	310
6.45.2.3	CONFIG_SCT_nOU	310
6.45.2.4	CONFIG_SCT_nRG	310
6.45.2.5	COUNTUP_TO	310
6.45.2.6	COUNTUP_TO	310
6.45.2.7	COUNTUP_TO_LIMIT_THEN_CLEAR_TO_ZERO	310
6.45.2.8	COUNTUP_TO_LIMIT_THEN_CLEAR_TO_ZERO	310
6.45.2.9	SCT_CONFIG_16BIT_COUNTER	310
6.45.2.10	SCT_CONFIG_32BIT_COUNTER	310
6.45.2.11	SCT_CONFIG_AUTOLIMIT_H	311
6.45.2.12	SCT_CONFIG_AUTOLIMIT_L	311

6.45.2.13 SCT_CONFIG_AUTOLIMIT_U . . . . .	311
6.45.2.14 SCT_CONFIG_CKSEL_FALLING_IN_0 . . . . .	311
6.45.2.15 SCT_CONFIG_CKSEL_FALLING_IN_1 . . . . .	311
6.45.2.16 SCT_CONFIG_CKSEL_FALLING_IN_2 . . . . .	311
6.45.2.17 SCT_CONFIG_CKSEL_FALLING_IN_3 . . . . .	311
6.45.2.18 SCT_CONFIG_CKSEL_FALLING_IN_4 . . . . .	311
6.45.2.19 SCT_CONFIG_CKSEL_FALLING_IN_5 . . . . .	311
6.45.2.20 SCT_CONFIG_CKSEL_FALLING_IN_6 . . . . .	311
6.45.2.21 SCT_CONFIG_CKSEL_FALLING_IN_7 . . . . .	311
6.45.2.22 SCT_CONFIG_CKSEL_RISING_IN_0 . . . . .	311
6.45.2.23 SCT_CONFIG_CKSEL_RISING_IN_1 . . . . .	312
6.45.2.24 SCT_CONFIG_CKSEL_RISING_IN_2 . . . . .	312
6.45.2.25 SCT_CONFIG_CKSEL_RISING_IN_3 . . . . .	312
6.45.2.26 SCT_CONFIG_CKSEL_RISING_IN_4 . . . . .	312
6.45.2.27 SCT_CONFIG_CKSEL_RISING_IN_5 . . . . .	312
6.45.2.28 SCT_CONFIG_CKSEL_RISING_IN_6 . . . . .	312
6.45.2.29 SCT_CONFIG_CKSEL_RISING_IN_7 . . . . .	312
6.45.2.30 SCT_CONFIG_CLKMODE_BUSCLK . . . . .	312
6.45.2.31 SCT_CONFIG_CLKMODE_INCLK . . . . .	312
6.45.2.32 SCT_CONFIG_CLKMODE_INEDGECLK . . . . .	312
6.45.2.33 SCT_CONFIG_CLKMODE_PRESCALED_SCT_INPUT . . . . .	312
6.45.2.34 SCT_CONFIG_CLKMODE_PRESCALED_SYSCLK . . . . .	313
6.45.2.35 SCT_CONFIG_CLKMODE_SCT_INPUT . . . . .	313
6.45.2.36 SCT_CONFIG_CLKMODE_SCTCLK . . . . .	313
6.45.2.37 SCT_CONFIG_CLKMODE_SYSCLK . . . . .	313
6.45.2.38 SCT_CONFIG_NORELOADH . . . . .	313
6.45.2.39 SCT_CONFIG_NORELOADL_U . . . . .	313
6.45.2.40 SCT_CTRL_BIDIR_H . . . . .	313
6.45.2.41 SCT_CTRL_BIDIR_L . . . . .	313
6.45.2.42 SCT_CTRL_CLRCTR_H . . . . .	313
6.45.2.43 SCT_CTRL_CLRCTR_L . . . . .	314
6.45.2.44 SCT_CTRL_HALT_H . . . . .	314
6.45.2.45 SCT_CTRL_HALT_L . . . . .	314
6.45.2.46 SCT_CTRL_PRE_H . . . . .	314
6.45.2.47 SCT_CTRL_PRE_L . . . . .	314
6.45.2.48 SCT_CTRL_STOP_H . . . . .	314
6.45.2.49 SCT_CTRL_STOP_L . . . . .	314
6.45.2.50 SCT_EV_CTRL_COMBMODE_AND . . . . .	314
6.45.2.51 SCT_EV_CTRL_COMBMODE_IO . . . . .	314
6.45.2.52 SCT_EV_CTRL_COMBMODE_MATCH . . . . .	314

6.45.2.53	SCT_EV_CTRL_COMBMODE_OR	315
6.45.2.54	SCT_EV_CTRL_DIRECTION_DOWN	315
6.45.2.55	SCT_EV_CTRL_DIRECTION_INDEPENDENT	315
6.45.2.56	SCT_EV_CTRL_DIRECTION_UP	315
6.45.2.57	SCT_EV_CTRL_HEVENT_H	315
6.45.2.58	SCT_EV_CTRL_HEVENT_L	315
6.45.2.59	SCT_EV_CTRL_IOCOND_FALL	315
6.45.2.60	SCT_EV_CTRL_IOCOND_HIGH	315
6.45.2.61	SCT_EV_CTRL_IOCOND_LOW	315
6.45.2.62	SCT_EV_CTRL_IOCOND_RISE	315
6.45.2.63	SCT_EV_CTRL_IOSEL	315
6.45.2.64	SCT_EV_CTRL_MATCHMEM	315
6.45.2.65	SCT_EV_CTRL_MATCHSEL	316
6.45.2.66	SCT_EV_CTRL_OUTSEL_INPUT	316
6.45.2.67	SCT_EV_CTRL_OUTSEL_OUTPUT	316
6.45.2.68	SCT_EV_CTRL_STATELD	316
6.45.2.69	SCT_EV_CTRL_STATEV	316
6.45.2.70	SCT_RES_CLEAR_OUTPUT	316
6.45.2.71	SCT_RES_NOCHANGE	316
6.45.2.72	SCT_RES_SET_OUTPUT	316
6.45.2.73	SCT_RES_TOGGLE_OUTPUT	316
6.45.3	Enumeration Type Documentation	316
6.45.3.1	CHIP_SCT_EVENT_T	316
6.45.3.2	CHIP_SCT_MATCH_REG_T	317
6.45.4	Function Documentation	317
6.45.4.1	Chip_SCT_ClearControl(LPC_SCT_T *pSCT, uint32_t value)	317
6.45.4.2	Chip_SCT_ClearEventFlag(LPC_SCT_T *pSCT, CHIP_SCT_EVENT_T evt)	318
6.45.4.3	Chip_SCT_Config(LPC_SCT_T *pSCT, uint32_t cfg)	318
6.45.4.4	Chip_SCT_DeInit(LPC_SCT_T *pSCT)	318
6.45.4.5	Chip_SCT_DisableEventInt(LPC_SCT_T *pSCT, CHIP_SCT_EVENT_T evt)	318
6.45.4.6	Chip_SCT_EnableEventInt(LPC_SCT_T *pSCT, CHIP_SCT_EVENT_T evt)	319
6.45.4.7	Chip_SCT_EventControl(LPC_SCT_T *pSCT, uint32_t event_number, uint32_t value)	319
6.45.4.8	Chip_SCT_EventStateMask(LPC_SCT_T *pSCT, uint32_t event_number, uint32_t event_state_mask)	319
6.45.4.9	Chip_SCT_Init(LPC_SCT_T *pSCT)	320
6.45.4.10	Chip_SCT_Limit(LPC_SCT_T *pSCT, uint32_t value)	321
6.45.4.11	Chip_SCT_SetClrControl(LPC_SCT_T *pSCT, uint32_t value, FunctionalState ena)	321
6.45.4.12	Chip_SCT_SetConflictResolution(LPC_SCT_T *pSCT, uint8_t outnum, uint8_t value)	321
6.45.4.13	Chip_SCT_SetControl(LPC_SCT_T *pSCT, uint32_t value)	322

6.45.4.14	Chip_SCT_SetCount(LPC_SCT_T *pSCT, uint32_t count)	322
6.45.4.15	Chip_SCT_SetCountH(LPC_SCT_T *pSCT, uint16_t count)	322
6.45.4.16	Chip_SCT_SetCountL(LPC_SCT_T *pSCT, uint16_t count)	322
6.45.4.17	Chip_SCT_SetMatchCount(LPC_SCT_T *pSCT, CHIP_SCT_MATCH_REG_↵ T n, uint32_t value)	323
6.45.4.18	Chip_SCT_SetMatchReload(LPC_SCT_T *pSCT, CHIP_SCT_MATCH_REG_↵ T n, uint32_t value)	323
6.46	CHIP: LPC5411X System and Control Driver	324
6.46.1	Detailed Description	324
6.46.2	Macro Definition Documentation	326
6.46.2.1	SYSCON_AUTOCGOR_MASK	326
6.46.2.2	SYSCON_AUTOCGOR_RAM0X	327
6.46.2.3	SYSCON_AUTOCGOR_RAM1	327
6.46.2.4	SYSCON_AUTOCGOR_RAM2	327
6.46.2.5	SYSCON_FROCTRL_HSPDCLK	327
6.46.2.6	SYSCON_FROCTRL_MASK	327
6.46.2.7	SYSCON_FROCTRL_SEL96MHZ	327
6.46.2.8	SYSCON_FROCTRL_USBCLKADJ	327
6.46.2.9	SYSCON_FROCTRL_USBMODCHG	327
6.46.2.10	SYSCON_FROCTRL_WRTRIM	327
6.46.2.11	SYSCON_NMISRC_M0_ENABLE	328
6.46.2.12	SYSCON_NMISRC_M4_ENABLE	328
6.46.2.13	SYSCON_PDRUNCFG_LP_VDDFLASH	328
6.46.2.14	SYSCON_PDRUNCFG_PD_ADC0	328
6.46.2.15	SYSCON_PDRUNCFG_PD_BOD_INTR	328
6.46.2.16	SYSCON_PDRUNCFG_PD_BOD_RST	328
6.46.2.17	SYSCON_PDRUNCFG_PD_FLASH	328
6.46.2.18	SYSCON_PDRUNCFG_PD_FRO	328
6.46.2.19	SYSCON_PDRUNCFG_PD_ROM	328
6.46.2.20	SYSCON_PDRUNCFG_PD_SRAM0	329
6.46.2.21	SYSCON_PDRUNCFG_PD_SRAM1	329
6.46.2.22	SYSCON_PDRUNCFG_PD_SRAM2	329
6.46.2.23	SYSCON_PDRUNCFG_PD_SRAMX	329
6.46.2.24	SYSCON_PDRUNCFG_PD_SYS_PLL	329
6.46.2.25	SYSCON_PDRUNCFG_PD_TS	329
6.46.2.26	SYSCON_PDRUNCFG_PD_USB_PHY	329
6.46.2.27	SYSCON_PDRUNCFG_PD_VDDA_ENA	329
6.46.2.28	SYSCON_PDRUNCFG_PD_VDDFLASH	329
6.46.2.29	SYSCON_PDRUNCFG_PD_VDDHV_ENA	330
6.46.2.30	SYSCON_PDRUNCFG_PD_VREFP	330

6.46.2.31	SYSCON_PDRUNCFG_PD_WDT_OSC	330
6.46.2.32	SYSCON_RST_BOD	330
6.46.2.33	SYSCON_RST_EXTRST	330
6.46.2.34	SYSCON_RST_POR	330
6.46.2.35	SYSCON_RST_SYSRST	330
6.46.2.36	SYSCON_RST_WDT	330
6.46.3	Enumeration Type Documentation	330
6.46.3.1	CHIP_SYSCON_BOOT_MODE_REMAP_T	330
6.46.3.2	CHIP_SYSCON_PERIPH_RESET_T	331
6.46.3.3	CHIP_SYSCON_WAKEUP_T	332
6.46.3.4	SYSCON_FLASHTIM_T	333
6.46.4	Function Documentation	333
6.46.4.1	Chip_SYSCON_ClearSystemRSTStatus(uint32_t reset)	333
6.46.4.2	Chip_SYSCON_DisableAutoClocking(uint32_t mask)	333
6.46.4.3	Chip_SYSCON_DisableNMISource(void)	334
6.46.4.4	Chip_SYSCON_DisableWakeup(CHIP_SYSCON_WAKEUP_T periphId)	334
6.46.4.5	Chip_SYSCON_Enable_ASYNC_Syscon(bool enable)	334
6.46.4.6	Chip_SYSCON_EnableAutoClocking(uint32_t mask)	334
6.46.4.7	Chip_SYSCON_EnableNMISource(void)	335
6.46.4.8	Chip_SYSCON_EnableWakeup(CHIP_SYSCON_WAKEUP_T periphId)	335
6.46.4.9	Chip_SYSCON_GetCompFreqMeas(uint32_t refClockRate)	335
6.46.4.10	Chip_SYSCON_GetDeviceID(void)	335
6.46.4.11	Chip_SYSCON_GetMemoryMap(void)	335
6.46.4.12	Chip_SYSCON_GetPORPIOStatus(uint8_t port)	336
6.46.4.13	Chip_SYSCON_GetPowerStates(void)	337
6.46.4.14	Chip_SYSCON_GetRawFreqMeasCapval(void)	337
6.46.4.15	Chip_SYSCON_GetResetPIOStatus(uint8_t port)	337
6.46.4.16	Chip_SYSCON_GetSystemRSTStatus(void)	338
6.46.4.17	Chip_SYSCON_IsFreqMeasComplete(void)	338
6.46.4.18	Chip_SYSCON_Map(CHIP_SYSCON_BOOT_MODE_REMAP_T remap)	338
6.46.4.19	Chip_SYSCON_PeriphReset(CHIP_SYSCON_PERIPH_RESET_T periph)	338
6.46.4.20	Chip_SYSCON_PowerDown(uint32_t powerdownmask)	338
6.46.4.21	Chip_SYSCON_PowerUp(uint32_t powerupmask)	339
6.46.4.22	Chip_SYSCON_SetFLASHAccess(SYSCON_FLASHTIM_T clks)	339
6.46.4.23	Chip_SYSCON_SetNMISource(uint32_t intsrc)	339
6.46.4.24	Chip_SYSCON_SetSYSTCKCAL(uint32_t sysCalVal)	339
6.46.4.25	Chip_SYSCON_SetUSARTFRGCtrl(uint8_t fmul, uint8_t fdiv)	340
6.46.4.26	Chip_SYSCON_StartFreqMeas(void)	340
6.47	CHIP: LPC5411X UART Driver	341
6.47.1	Detailed Description	341

6.47.2	Macro Definition Documentation	344
6.47.2.1	ECHO_DIS	344
6.47.2.2	ECHO_EN	344
6.47.2.3	UART_CFG_AUTOADDR	344
6.47.2.4	UART_CFG_BITMASK	345
6.47.2.5	UART_CFG_CLKPOL	345
6.47.2.6	UART_CFG_CTSEN	345
6.47.2.7	UART_CFG_DATALEN_7	345
6.47.2.8	UART_CFG_DATALEN_8	345
6.47.2.9	UART_CFG_DATALEN_9	345
6.47.2.10	UART_CFG_ENABLE	345
6.47.2.11	UART_CFG_IOMODE	345
6.47.2.12	UART_CFG_LINMODE	345
6.47.2.13	UART_CFG_LOOP	346
6.47.2.14	UART_CFG_MODE32K	346
6.47.2.15	UART_CFG_OEPOL	346
6.47.2.16	UART_CFG_OESEL	346
6.47.2.17	UART_CFG_OETA	346
6.47.2.18	UART_CFG_PARITY_EVEN	346
6.47.2.19	UART_CFG_PARITY_NONE	346
6.47.2.20	UART_CFG_PARITY_ODD	346
6.47.2.21	UART_CFG_RXPOL	346
6.47.2.22	UART_CFG_STOPLEN_1	347
6.47.2.23	UART_CFG_STOPLEN_2	347
6.47.2.24	UART_CFG_SYNCEN	347
6.47.2.25	UART_CFG_SYNCMST	347
6.47.2.26	UART_CFG_TXPOL	347
6.47.2.27	UART_CTRL_ADDRDET	347
6.47.2.28	UART_CTRL_AUTOBAUD	347
6.47.2.29	UART_CTRL_CC	347
6.47.2.30	UART_CTRL_CLRCONRX	347
6.47.2.31	UART_CTRL_TXBRKEN	348
6.47.2.32	UART_CTRL_TXDIS	348
6.47.2.33	UART_FIFO_DEPTH	348
6.47.2.34	UART_FIFOCFG_BITMASK	348
6.47.2.35	UART_FIFOCFG_DMARX	348
6.47.2.36	UART_FIFOCFG_DMATX	348
6.47.2.37	UART_FIFOCFG_EMPTYRX	348
6.47.2.38	UART_FIFOCFG_EMPTYTX	348
6.47.2.39	UART_FIFOCFG_ENBLERX	348



6.47.2.40 UART_FIFOCFG_ENABLETX . . . . .	349
6.47.2.41 UART_FIFOCFG_WAKERX . . . . .	349
6.47.2.42 UART_FIFOCFG_WAKETX . . . . .	349
6.47.2.43 UART_FIFOINT_BITMASK . . . . .	349
6.47.2.44 UART_FIFOINT_PERINT . . . . .	349
6.47.2.45 UART_FIFOINT_RXERR . . . . .	349
6.47.2.46 UART_FIFOINT_RXLVL . . . . .	349
6.47.2.47 UART_FIFOINT_TXERR . . . . .	349
6.47.2.48 UART_FIFOINT_TXLVL . . . . .	349
6.47.2.49 UART_FIFOSTAT_BITMASK . . . . .	349
6.47.2.50 UART_FIFOSTAT_PERIPH . . . . .	350
6.47.2.51 UART_FIFOSTAT_RXERR . . . . .	350
6.47.2.52 UART_FIFOSTAT_RXFULL . . . . .	350
6.47.2.53 UART_FIFOSTAT_RXLVL . . . . .	350
6.47.2.54 UART_FIFOSTAT_RXNOTEMPTY . . . . .	350
6.47.2.55 UART_FIFOSTAT_TXEMPTY . . . . .	350
6.47.2.56 UART_FIFOSTAT_TXERR . . . . .	350
6.47.2.57 UART_FIFOSTAT_TXLVL . . . . .	350
6.47.2.58 UART_FIFOSTAT_TXNOTFULL . . . . .	350
6.47.2.59 UART_FIFOTRIG_BITMASK . . . . .	351
6.47.2.60 UART_FIFOTRIG_RXLVL . . . . .	351
6.47.2.61 UART_FIFOTRIG_RXLVLENA . . . . .	351
6.47.2.62 UART_FIFOTRIG_TXLVL . . . . .	351
6.47.2.63 UART_FIFOTRIG_TXLVLENA . . . . .	351
6.47.2.64 UART_INT_ABERR . . . . .	351
6.47.2.65 UART_INT_DELTACTS . . . . .	351
6.47.2.66 UART_INT_DELTARXBRK . . . . .	351
6.47.2.67 UART_INT_FRAMERR . . . . .	351
6.47.2.68 UART_INT_PARITYERR . . . . .	352
6.47.2.69 UART_INT_RXNOISE . . . . .	352
6.47.2.70 UART_INT_START . . . . .	352
6.47.2.71 UART_INT_TXDIS . . . . .	352
6.47.2.72 UART_INT_TXIDLE . . . . .	352
6.47.2.73 UART_STAT_ABERR . . . . .	352
6.47.2.74 UART_STAT_CTS . . . . .	352
6.47.2.75 UART_STAT_DELTACTS . . . . .	352
6.47.2.76 UART_STAT_DELTARXBRK . . . . .	352
6.47.2.77 UART_STAT_FRM_ERRINT . . . . .	353
6.47.2.78 UART_STAT_PAR_ERRINT . . . . .	353
6.47.2.79 UART_STAT_RXBRK . . . . .	353

6.47.2.80	UART_STAT_RXIDLE . . . . .	353
6.47.2.81	UART_STAT_RXNOISEINT . . . . .	353
6.47.2.82	UART_STAT_START . . . . .	353
6.47.2.83	UART_STAT_TXDISINT . . . . .	353
6.47.2.84	UART_STAT_TXIDLE . . . . .	353
6.47.3	Function Documentation . . . . .	353
6.47.3.1	Chip_UART_AutoBaud(LPC_USART_T *pUART) . . . . .	353
6.47.3.2	Chip_UART_ClearFIFOCfg(LPC_USART_T *pUART, uint32_t cfg) . . . . .	354
6.47.3.3	Chip_UART_ClearFIFOStatus(LPC_USART_T *pUART, uint32_t mask) . . . . .	354
6.47.3.4	Chip_UART_ClearStatus(LPC_USART_T *pUART, uint32_t stsMask) . . . . .	354
6.47.3.5	Chip_UART_ConfigData(LPC_USART_T *pUART, uint32_t config) . . . . .	355
6.47.3.6	Chip_UART_ConfigDMA(LPC_USART_T *pUART) . . . . .	356
6.47.3.7	Chip_UART_DeInit(LPC_USART_T *pUART) . . . . .	356
6.47.3.8	Chip_UART_Disable(LPC_USART_T *pUART) . . . . .	356
6.47.3.9	Chip_UART_DisableFIFOInts(LPC_USART_T *pUART, uint32_t intMask) . . . . .	357
6.47.3.10	Chip_UART_Enable(LPC_USART_T *pUART) . . . . .	358
6.47.3.11	Chip_UART_EnableFIFOInts(LPC_USART_T *pUART, uint32_t intMask) . . . . .	358
6.47.3.12	Chip_UART_FlushFIFOs(LPC_USART_T *pUART) . . . . .	358
6.47.3.13	Chip_UART_GetFIFOEnabledInts(LPC_USART_T *pUART) . . . . .	358
6.47.3.14	Chip_UART_GetFIFOPendingInts(LPC_USART_T *pUART) . . . . .	359
6.47.3.15	Chip_UART_GetFIFOStatus(LPC_USART_T *pUART) . . . . .	359
6.47.3.16	Chip_UART_GetIntsEnabled(LPC_USART_T *pUART) . . . . .	359
6.47.3.17	Chip_UART_GetIntStatus(LPC_USART_T *pUART) . . . . .	360
6.47.3.18	Chip_UART_GetStatus(LPC_USART_T *pUART) . . . . .	361
6.47.3.19	Chip_UART_Init(LPC_USART_T *pUART) . . . . .	361
6.47.3.20	Chip_UART_IntDisable(LPC_USART_T *pUART, uint32_t intMask) . . . . .	361
6.47.3.21	Chip_UART_IntEnable(LPC_USART_T *pUART, uint32_t intMask) . . . . .	362
6.47.3.22	Chip_UART_IRQHandlerDMA(LPC_USART_T *pUART, UART_STATISTICS↔ _T *statistics) . . . . .	362
6.47.3.23	Chip_UART_IRQHandlerRB(LPC_USART_T *pUART, UART_STATISTICS↔ _T *statistics, RINGBUFF_T *pRXRB, RINGBUFF_T *pTXRB) . . . . .	362
6.47.3.24	Chip_UART_Read(LPC_USART_T *pUART, void *data, int numBytes) . . . . .	363
6.47.3.25	Chip_UART_ReadBlocking(LPC_USART_T *pUART, void *data, int numBytes) . . . . .	363
6.47.3.26	Chip_UART_ReadByte(LPC_USART_T *pUART) . . . . .	364
6.47.3.27	Chip_UART_ReadRB(LPC_USART_T *pUART, RINGBUFF_T *pRB, void *data, int bytes) . . . . .	365
6.47.3.28	Chip_UART_RXIntHandlerRB(LPC_USART_T *pUART, RINGBUFF_T *pRB) . . . . .	365
6.47.3.29	Chip_UART_Send(LPC_USART_T *pUART, const void *data, int numBytes) . . . . .	365
6.47.3.30	Chip_UART_SendBlocking(LPC_USART_T *pUART, const void *data, int num↔ Bytes) . . . . .	366
6.47.3.31	Chip_UART_SendByte(LPC_USART_T *pUART, uint8_t data) . . . . .	366

6.47.3.32	Chip_UART_SendRB(LPC_USART_T *pUART, RINGBUFF_T *pRB, const void *data, int count)	367
6.47.3.33	Chip_UART_SetBaud(LPC_USART_T *pUART, uint32_t baudrate)	368
6.47.3.34	Chip_UART_SetFIFOCfg(LPC_USART_T *pUART, uint32_t cfg)	368
6.47.3.35	Chip_UART_SetFIFOTrigLevel(LPC_USART_T *pUART, uint8_t tx_lvl, uint8_t rx_lvl)	368
6.47.3.36	Chip_UART_TXDisable(LPC_USART_T *pUART)	369
6.47.3.37	Chip_UART_TXEnable(LPC_USART_T *pUART)	369
6.47.3.38	Chip_UART_TXIntHandlerRB(LPC_USART_T *pUART, RINGBUFF_T *pRB)	369
6.48	CHIP: LPC5411X Windowed Watchdog driver	371
6.48.1	Detailed Description	371
6.48.2	Macro Definition Documentation	372
6.48.2.1	WWDT_WDMOD_BITMASK	372
6.48.2.2	WWDT_WDMOD_LOCK	372
6.48.2.3	WWDT_WDMOD_WDEN	372
6.48.2.4	WWDT_WDMOD_WDINT	372
6.48.2.5	WWDT_WDMOD_WDPROTECT	372
6.48.2.6	WWDT_WDMOD_WDRESET	372
6.48.2.7	WWDT_WDMOD_WDTOF	372
6.48.3	Function Documentation	372
6.48.3.1	Chip_WWDT_ClearStatusFlag(LPC_WWDT_T *pWWDT, uint32_t status)	372
6.48.3.2	Chip_WWDT_DeInit(LPC_WWDT_T *pWWDT)	373
6.48.3.3	Chip_WWDT_Feed(LPC_WWDT_T *pWWDT)	373
6.48.3.4	Chip_WWDT_GetCurrentCount(LPC_WWDT_T *pWWDT)	373
6.48.3.5	Chip_WWDT_GetStatus(LPC_WWDT_T *pWWDT)	374
6.48.3.6	Chip_WWDT_GetWarning(LPC_WWDT_T *pWWDT)	374
6.48.3.7	Chip_WWDT_GetWindow(LPC_WWDT_T *pWWDT)	374
6.48.3.8	Chip_WWDT_Init(LPC_WWDT_T *pWWDT)	374
6.48.3.9	Chip_WWDT_SetOption(LPC_WWDT_T *pWWDT, uint32_t options)	375
6.48.3.10	Chip_WWDT_SetTimeOut(LPC_WWDT_T *pWWDT, uint32_t timeout)	375
6.48.3.11	Chip_WWDT_SetWarning(LPC_WWDT_T *pWWDT, uint32_t timeout)	375
6.48.3.12	Chip_WWDT_SetWindow(LPC_WWDT_T *pWWDT, uint32_t timeout)	376
6.48.3.13	Chip_WWDT_Start(LPC_WWDT_T *pWWDT)	376
6.48.3.14	Chip_WWDT_UnsetOption(LPC_WWDT_T *pWWDT, uint32_t options)	376
6.49	CHIP: LPC5411X flexcomm API	378
6.49.1	Detailed Description	378
6.49.2	Macro Definition Documentation	379
6.49.2.1	ERR_FLEXCOMM_FUNCNOTSUPPORTED	379
6.49.2.2	ERR_FLEXCOMM_INVALIDBASE	379
6.49.2.3	ERR_FLEXCOMM_NOTFREE	379

6.49.2.4	FLEXCOMM_ID_I2C	379
6.49.2.5	FLEXCOMM_ID_I2S	379
6.49.2.6	FLEXCOMM_ID_SPI	379
6.49.2.7	FLEXCOMM_ID_USART	379
6.49.2.8	FLEXCOMM_LOCK	379
6.49.2.9	FLEXCOMM_PSEL_OFFSET	379
6.49.3	Typedef Documentation	380
6.49.3.1	LPC_FLEXCOMM_T	380
6.49.4	Enumeration Type Documentation	380
6.49.4.1	FLEXCOMM_PERIPH_T	380
6.49.5	Function Documentation	380
6.49.5.1	Chip_FLEXCOMM_DeInit(LPC_FLEXCOMM_T *pFCOMM)	380
6.49.5.2	Chip_FLEXCOMM_GetFunc(LPC_FLEXCOMM_T *pFCOMM)	380
6.49.5.3	Chip_FLEXCOMM_GetIndex(LPC_FLEXCOMM_T *pFCOMM)	381
6.49.5.4	Chip_FLEXCOMM_Init(LPC_FLEXCOMM_T *pFCOMM, FLEXCOMM_PERIPH_T periph)	382
6.49.5.5	Chip_FLEXCOMM_IsLocked(LPC_FLEXCOMM_T *pFCOMM)	382
6.49.5.6	Chip_FLEXCOMM_Lock(LPC_FLEXCOMM_T *pFCOMM)	382
6.49.5.7	Chip_FLEXCOMM_SetPeriph(LPC_FLEXCOMM_T *pFCOMM, FLEXCOMM_PERIPH_T periph, int lock)	383
6.50	CHIP: LPC5411X support functions	385
6.50.1	Detailed Description	385
6.50.2	Function Documentation	385
6.50.2.1	Chip_SetupExtInClocking(uint32_t iFreq)	385
6.50.2.2	Chip_SetupFROClocking(uint32_t iFreq)	385
6.50.2.3	Chip_SetupIrcClocking(uint32_t iFreq)	386
6.50.2.4	Chip_SystemInit(void)	386
6.50.2.5	Chip_USB_Init(void)	386
6.50.2.6	Chip_USB_TrimOff(int enable)	386
6.50.2.7	SystemCoreClockUpdate(void)	387
6.50.3	Variable Documentation	387
6.50.3.1	SystemCoreClock	387
6.51	CHIP: LPC5411x Chip driver build time options	388
6.51.1	Detailed Description	388
6.51.2	Variable Documentation	388
6.51.2.1	ExtClockIn	388
6.52	CHIP: LPC5411x I2C driver	389
6.52.1	Detailed Description	389
6.52.2	Macro Definition Documentation	393
6.52.2.1	I2C_CFG_MASK	393

6.52.2.2	I2C_CFG_MASK . . . . .	393
6.52.2.3	I2C_CFG_MONCLKSTR . . . . .	393
6.52.2.4	I2C_CFG_MONCLKSTR . . . . .	393
6.52.2.5	I2C_CFG_MONEN . . . . .	393
6.52.2.6	I2C_CFG_MONEN . . . . .	393
6.52.2.7	I2C_CFG_MSTEN . . . . .	393
6.52.2.8	I2C_CFG_MSTEN . . . . .	393
6.52.2.9	I2C_CFG_SLVEN . . . . .	394
6.52.2.10	I2C_CFG_SLVEN . . . . .	394
6.52.2.11	I2C_CFG_TIMEOUTEN . . . . .	394
6.52.2.12	I2C_CFG_TIMEOUTEN . . . . .	394
6.52.2.13	I2C_INTENCLR_EVENTTIMEOUT . . . . .	394
6.52.2.14	I2C_INTENCLR_EVENTTIMEOUT . . . . .	394
6.52.2.15	I2C_INTENCLR_MONIDLE . . . . .	394
6.52.2.16	I2C_INTENCLR_MONIDLE . . . . .	394
6.52.2.17	I2C_INTENCLR_MONOV . . . . .	394
6.52.2.18	I2C_INTENCLR_MONOV . . . . .	395
6.52.2.19	I2C_INTENCLR_MONRDY . . . . .	395
6.52.2.20	I2C_INTENCLR_MONRDY . . . . .	395
6.52.2.21	I2C_INTENCLR_MSTPENDING . . . . .	395
6.52.2.22	I2C_INTENCLR_MSTPENDING . . . . .	395
6.52.2.23	I2C_INTENCLR_MSTRARBLOSS . . . . .	395
6.52.2.24	I2C_INTENCLR_MSTRARBLOSS . . . . .	395
6.52.2.25	I2C_INTENCLR_MSTSTSTPERR . . . . .	395
6.52.2.26	I2C_INTENCLR_MSTSTSTPERR . . . . .	395
6.52.2.27	I2C_INTENCLR_SCLTIMEOUT . . . . .	396
6.52.2.28	I2C_INTENCLR_SCLTIMEOUT . . . . .	396
6.52.2.29	I2C_INTENCLR_SLVDESEL . . . . .	396
6.52.2.30	I2C_INTENCLR_SLVDESEL . . . . .	396
6.52.2.31	I2C_INTENCLR_SLVNOTSTR . . . . .	396
6.52.2.32	I2C_INTENCLR_SLVNOTSTR . . . . .	396
6.52.2.33	I2C_INTENCLR_SLVPENDING . . . . .	396
6.52.2.34	I2C_INTENCLR_SLVPENDING . . . . .	396
6.52.2.35	I2C_INTENSET_EVENTTIMEOUT . . . . .	396
6.52.2.36	I2C_INTENSET_EVENTTIMEOUT . . . . .	397
6.52.2.37	I2C_INTENSET_MONIDLE . . . . .	397
6.52.2.38	I2C_INTENSET_MONIDLE . . . . .	397
6.52.2.39	I2C_INTENSET_MONOV . . . . .	397
6.52.2.40	I2C_INTENSET_MONOV . . . . .	397
6.52.2.41	I2C_INTENSET_MONRDY . . . . .	397

6.52.2.42 I2C_INTENSET_MONRDY . . . . .	397
6.52.2.43 I2C_INTENSET_MSTPENDING . . . . .	397
6.52.2.44 I2C_INTENSET_MSTPENDING . . . . .	397
6.52.2.45 I2C_INTENSET_MSTRARBLOSS . . . . .	398
6.52.2.46 I2C_INTENSET_MSTRARBLOSS . . . . .	398
6.52.2.47 I2C_INTENSET_MSTSTSTPERR . . . . .	398
6.52.2.48 I2C_INTENSET_MSTSTSTPERR . . . . .	398
6.52.2.49 I2C_INTENSET_SCLTIMEOUT . . . . .	398
6.52.2.50 I2C_INTENSET_SCLTIMEOUT . . . . .	398
6.52.2.51 I2C_INTENSET_SLVDESEL . . . . .	398
6.52.2.52 I2C_INTENSET_SLVDESEL . . . . .	398
6.52.2.53 I2C_INTENSET_SLVNOTSTR . . . . .	398
6.52.2.54 I2C_INTENSET_SLVNOTSTR . . . . .	399
6.52.2.55 I2C_INTENSET_SLVPENDING . . . . .	399
6.52.2.56 I2C_INTENSET_SLVPENDING . . . . .	399
6.52.2.57 I2C_INTSTAT_EVENTTIMEOUT . . . . .	399
6.52.2.58 I2C_INTSTAT_EVENTTIMEOUT . . . . .	399
6.52.2.59 I2C_INTSTAT_MONIDLE . . . . .	399
6.52.2.60 I2C_INTSTAT_MONIDLE . . . . .	399
6.52.2.61 I2C_INTSTAT_MONOV . . . . .	399
6.52.2.62 I2C_INTSTAT_MONOV . . . . .	399
6.52.2.63 I2C_INTSTAT_MONRDY . . . . .	400
6.52.2.64 I2C_INTSTAT_MONRDY . . . . .	400
6.52.2.65 I2C_INTSTAT_MSTPENDING . . . . .	400
6.52.2.66 I2C_INTSTAT_MSTPENDING . . . . .	400
6.52.2.67 I2C_INTSTAT_MSTRARBLOSS . . . . .	400
6.52.2.68 I2C_INTSTAT_MSTRARBLOSS . . . . .	400
6.52.2.69 I2C_INTSTAT_MSTSTSTPERR . . . . .	400
6.52.2.70 I2C_INTSTAT_MSTSTSTPERR . . . . .	400
6.52.2.71 I2C_INTSTAT_SCLTIMEOUT . . . . .	400
6.52.2.72 I2C_INTSTAT_SCLTIMEOUT . . . . .	401
6.52.2.73 I2C_INTSTAT_SLVDESEL . . . . .	401
6.52.2.74 I2C_INTSTAT_SLVDESEL . . . . .	401
6.52.2.75 I2C_INTSTAT_SLVNOTSTR . . . . .	401
6.52.2.76 I2C_INTSTAT_SLVNOTSTR . . . . .	401
6.52.2.77 I2C_INTSTAT_SLVPENDING . . . . .	401
6.52.2.78 I2C_INTSTAT_SLVPENDING . . . . .	401
6.52.2.79 I2C_MONRXDAT_DATA . . . . .	401
6.52.2.80 I2C_MONRXDAT_DATA . . . . .	401
6.52.2.81 I2C_MONRXDAT_MONNACK . . . . .	402

6.52.2.82 I2C_MONRXDAT_MONNACK . . . . .	402
6.52.2.83 I2C_MONRXDAT_MONRESTART . . . . .	402
6.52.2.84 I2C_MONRXDAT_MONRESTART . . . . .	402
6.52.2.85 I2C_MONRXDAT_MONSTART . . . . .	402
6.52.2.86 I2C_MONRXDAT_MONSTART . . . . .	402
6.52.2.87 I2C_MSTCTL_MSTCONTINUE . . . . .	402
6.52.2.88 I2C_MSTCTL_MSTCONTINUE . . . . .	402
6.52.2.89 I2C_MSTCTL_MSTDMA . . . . .	402
6.52.2.90 I2C_MSTCTL_MSTDMA . . . . .	403
6.52.2.91 I2C_MSTCTL_MSTSTART . . . . .	403
6.52.2.92 I2C_MSTCTL_MSTSTART . . . . .	403
6.52.2.93 I2C_MSTCTL_MSTSTOP . . . . .	403
6.52.2.94 I2C_MSTCTL_MSTSTOP . . . . .	403
6.52.2.95 I2C_MSTDAT_DATAMASK . . . . .	403
6.52.2.96 I2C_MSTDAT_DATAMASK . . . . .	403
6.52.2.97 I2C_MSTTIME_MSTSCHIGH . . . . .	403
6.52.2.98 I2C_MSTTIME_MSTSCHIGH . . . . .	403
6.52.2.99 I2C_MSTTIME_MSTSCLOW . . . . .	404
6.52.2.100 I2C_MSTTIME_MSTSCLOW . . . . .	404
6.52.2.101 I2C_SLVADR_MASK . . . . .	404
6.52.2.102 I2C_SLVADR_MASK . . . . .	404
6.52.2.103 I2C_SLVADR_SADISABLE . . . . .	404
6.52.2.104 I2C_SLVADR_SADISABLE . . . . .	404
6.52.2.105 I2C_SLVADR_SLVADR . . . . .	404
6.52.2.106 I2C_SLVADR_SLVADR . . . . .	404
6.52.2.107 I2C_SLVCTL_SLVCONTINUE . . . . .	404
6.52.2.108 I2C_SLVCTL_SLVCONTINUE . . . . .	405
6.52.2.109 I2C_SLVCTL_SLVDMA . . . . .	405
6.52.2.110 I2C_SLVCTL_SLVDMA . . . . .	405
6.52.2.111 I2C_SLVCTL_SLVNACK . . . . .	405
6.52.2.112 I2C_SLVCTL_SLVNACK . . . . .	405
6.52.2.113 I2C_SLVDAT_DATAMASK . . . . .	405
6.52.2.114 I2C_SLVDAT_DATAMASK . . . . .	405
6.52.2.115 I2C_SLVQUAL_QUALMODE0 . . . . .	405
6.52.2.116 I2C_SLVQUAL_QUALMODE0 . . . . .	405
6.52.2.117 I2C_SLVQUAL_SLVQUAL0 . . . . .	406
6.52.2.118 I2C_SLVQUAL_SLVQUAL0 . . . . .	406
6.52.2.119 I2C_STAT_EVENTTIMEOUT . . . . .	406
6.52.2.120 I2C_STAT_EVENTTIMEOUT . . . . .	406
6.52.2.121 I2C_STAT_MONACTIVE . . . . .	406

6.52.2.122C_STAT_MONACTIVE . . . . .	406
6.52.2.123C_STAT_MONIDLE . . . . .	406
6.52.2.124C_STAT_MONIDLE . . . . .	406
6.52.2.125C_STAT_MONOV . . . . .	406
6.52.2.126C_STAT_MONOV . . . . .	407
6.52.2.127C_STAT_MONRDY . . . . .	407
6.52.2.128C_STAT_MONRDY . . . . .	407
6.52.2.129C_STAT_MSTCODE_IDLE . . . . .	407
6.52.2.130C_STAT_MSTCODE_IDLE . . . . .	407
6.52.2.131C_STAT_MSTCODE_NACKADR . . . . .	407
6.52.2.132C_STAT_MSTCODE_NACKADR . . . . .	407
6.52.2.133C_STAT_MSTCODE_NACKDAT . . . . .	407
6.52.2.134C_STAT_MSTCODE_NACKDAT . . . . .	407
6.52.2.135C_STAT_MSTCODE_RXREADY . . . . .	408
6.52.2.136C_STAT_MSTCODE_RXREADY . . . . .	408
6.52.2.137C_STAT_MSTCODE_TXREADY . . . . .	408
6.52.2.138C_STAT_MSTCODE_TXREADY . . . . .	408
6.52.2.139C_STAT_MSTPENDING . . . . .	408
6.52.2.140C_STAT_MSTPENDING . . . . .	408
6.52.2.141C_STAT_MSTRARBLOSS . . . . .	408
6.52.2.142C_STAT_MSTRARBLOSS . . . . .	408
6.52.2.143C_STAT_MSTSTATE . . . . .	408
6.52.2.144C_STAT_MSTSTATE . . . . .	409
6.52.2.145C_STAT_MSTSTSTPERR . . . . .	409
6.52.2.146C_STAT_MSTSTSTPERR . . . . .	409
6.52.2.147C_STAT_SCLTIMEOUT . . . . .	409
6.52.2.148C_STAT_SCLTIMEOUT . . . . .	409
6.52.2.149C_STAT_SLVCODE_ADDR . . . . .	409
6.52.2.150C_STAT_SLVCODE_ADDR . . . . .	409
6.52.2.151C_STAT_SLVCODE_RX . . . . .	409
6.52.2.152C_STAT_SLVCODE_RX . . . . .	409
6.52.2.153C_STAT_SLVCODE_TX . . . . .	410
6.52.2.154C_STAT_SLVCODE_TX . . . . .	410
6.52.2.155C_STAT_SLVDESEL . . . . .	410
6.52.2.156C_STAT_SLVDESEL . . . . .	410
6.52.2.157C_STAT_SLVIDX . . . . .	410
6.52.2.158C_STAT_SLVIDX . . . . .	410
6.52.2.159C_STAT_SLVNOTSTR . . . . .	410
6.52.2.160C_STAT_SLVNOTSTR . . . . .	410
6.52.2.161C_STAT_SLVPENDING . . . . .	410



6.52.2.1622C_STAT_SLVPENDING . . . . .	411
6.52.2.1632C_STAT_SLVSEL . . . . .	411
6.52.2.1642C_STAT_SLVSEL . . . . .	411
6.52.2.1652C_STAT_SLVSTATE . . . . .	411
6.52.2.1662C_STAT_SLVSTATE . . . . .	411
6.52.2.1672C_TIMEOUT_VAL . . . . .	411
6.52.2.1682C_TIMEOUT_VAL . . . . .	411
6.52.3 Function Documentation . . . . .	411
6.52.3.1 Chip_I2C_ClearInt(LPC_I2C_T *pl2C, uint32_t intClr) . . . . .	411
6.52.3.2 Chip_I2C_DeInit(LPC_I2C_T *pl2C) . . . . .	412
6.52.3.3 Chip_I2C_DisableInt(LPC_I2C_T *pl2C, uint32_t intClr) . . . . .	412
6.52.3.4 Chip_I2C_EnableInt(LPC_I2C_T *pl2C, uint32_t intEn) . . . . .	412
6.52.3.5 Chip_I2C_GetClockDiv(LPC_I2C_T *pl2C) . . . . .	412
6.52.3.6 Chip_I2C_GetPendingInt(LPC_I2C_T *pl2C) . . . . .	413
6.52.3.7 Chip_I2C_Init(LPC_I2C_T *pl2C) . . . . .	413
6.52.3.8 Chip_I2C_SetClockDiv(LPC_I2C_T *pl2C, uint32_t clkdiv) . . . . .	413
6.52.4 Variable Documentation . . . . .	414
6.52.4.1 CLKDIV . . . . .	414
6.52.4.2 INTENCLR . . . . .	414
6.52.4.3 INTENSET . . . . .	414
6.52.4.4 INTSTAT . . . . .	414
6.52.4.5 LPC_I2C_T . . . . .	414
6.52.4.6 MONRXDAT . . . . .	414
6.52.4.7 MSTCTL . . . . .	414
6.52.4.8 MSTDAT . . . . .	415
6.52.4.9 MSTTIME . . . . .	415
6.52.4.10 PID . . . . .	415
6.52.4.11 PSELID . . . . .	415
6.52.4.12 RESERVED0 . . . . .	415
6.52.4.13 RESERVED1 . . . . .	415
6.52.4.14 RESERVED2 . . . . .	415
6.52.4.15 SLVADR . . . . .	415
6.52.4.16 SLVCTL . . . . .	415
6.52.4.17 SLVDAT . . . . .	415
6.52.4.18 SLVQUAL0 . . . . .	416
6.52.4.19 STAT . . . . .	416
6.52.4.20 TIMEOUT . . . . .	416
6.53 CHIP: RTC tick to (a more) Universal Time conversion functions . . . . .	417
6.53.1 Detailed Description . . . . .	417
6.53.2 Macro Definition Documentation . . . . .	417

6.53.2.1	TM_DAYOFWEEK . . . . .	417
6.53.2.2	TM_YEAR_BASE . . . . .	417
6.53.3	Function Documentation . . . . .	417
6.53.3.1	ConvertRtcTime(uint32_t rtcTick, struct tm *pTime) . . . . .	417
6.53.3.2	ConvertTimeRtc(struct tm *pTime, uint32_t *rtcTick) . . . . .	417
6.54	CHIP: Simple ring buffer implementation . . . . .	419
6.54.1	Detailed Description . . . . .	419
6.54.2	Macro Definition Documentation . . . . .	419
6.54.2.1	RB_VHEAD . . . . .	419
6.54.2.2	RB_VTAIL . . . . .	419
6.54.3	Function Documentation . . . . .	420
6.54.3.1	RingBuffer_Flush(RINGBUFF_T *RingBuff) . . . . .	420
6.54.3.2	RingBuffer_GetCount(RINGBUFF_T *RingBuff) . . . . .	420
6.54.3.3	RingBuffer_GetFree(RINGBUFF_T *RingBuff) . . . . .	420
6.54.3.4	RingBuffer_GetSize(RINGBUFF_T *RingBuff) . . . . .	420
6.54.3.5	RingBuffer_Init(RINGBUFF_T *RingBuff, void *buffer, int itemSize, int count, void *(*cpyFunc)(void *dst, const void *src, uint32_t len)) . . . . .	421
6.54.3.6	RingBuffer_Insert(RINGBUFF_T *RingBuff, const void *data) . . . . .	422
6.54.3.7	RingBuffer_InsertMult(RINGBUFF_T *RingBuff, const void *data, int num) . . . . .	422
6.54.3.8	RingBuffer_IsEmpty(RINGBUFF_T *RingBuff) . . . . .	422
6.54.3.9	RingBuffer_IsFull(RINGBUFF_T *RingBuff) . . . . .	423
6.54.3.10	RingBuffer_Pop(RINGBUFF_T *RingBuff, void *data) . . . . .	423
6.54.3.11	RingBuffer_PopMult(RINGBUFF_T *RingBuff, void *data, int num) . . . . .	423
6.55	CHIP: Stopwatch primitives. . . . .	424
6.55.1	Detailed Description . . . . .	424
6.55.2	Function Documentation . . . . .	424
6.55.2.1	StopWatch_DelayMs(uint32_t mS) . . . . .	424
6.55.2.2	StopWatch_DelayTicks(uint32_t ticks) . . . . .	424
6.55.2.3	StopWatch_DelayUs(uint32_t uS) . . . . .	425
6.55.2.4	StopWatch_Elapsed(uint32_t startTime) . . . . .	425
6.55.2.5	StopWatch_Init(void) . . . . .	425
6.55.2.6	StopWatch_MsToTicks(uint32_t mS) . . . . .	425
6.55.2.7	StopWatch_Start(void) . . . . .	426
6.55.2.8	StopWatch_TicksPerSecond(void) . . . . .	426
6.55.2.9	StopWatch_TicksToMs(uint32_t ticks) . . . . .	426
6.55.2.10	StopWatch_TicksToUs(uint32_t ticks) . . . . .	426
6.55.2.11	StopWatch_UsToTicks(uint32_t uS) . . . . .	426
6.56	CHIP_5411X: LPC5411X M0 core peripheral interrupt numbers . . . . .	428
6.56.1	Detailed Description . . . . .	428
6.56.2	Enumeration Type Documentation . . . . .	428

6.56.2.1	LPC5411X_M0_IRQn_Type	428
6.57	CHIP_5411X: LPC5411X M4 core peripheral interrupt numbers	430
6.57.1	Detailed Description	430
6.57.2	Enumeration Type Documentation	430
6.57.2.1	LPC5411X_IRQn_Type	430
6.58	CMSIS Core Instruction Interface	432
6.59	CMSIS Core Register Access Functions	433
6.60	CMSIS Global Defines	434
6.61	CMSIS SIMD Intrinsics	435
6.62	CMSIS support	436
6.63	Chip specific drivers	437
6.63.1	Detailed Description	437
6.64	Code Red LPCXpresso support in LPCOpen	438
6.65	Common FreeRTOS functions shared with multiple platforms	439
6.66	Common components used with chip drivers	440
6.66.1	Detailed Description	440
6.67	Community support for LPCOpen	441
6.68	Copyright (C) 2013 NXP Semiconductors. All rights reserved.	442
6.69	Core Debug Registers (CoreDebug)	443
6.69.1	Detailed Description	443
6.69.2	Macro Definition Documentation	444
6.69.2.1	CoreDebug_DCRSR_REGSEL_Msk	444
6.69.2.2	CoreDebug_DCRSR_REGSEL_Pos	444
6.69.2.3	CoreDebug_DCRSR_REGWnR_Msk	444
6.69.2.4	CoreDebug_DCRSR_REGWnR_Pos	444
6.69.2.5	CoreDebug_DEMCR_MON_EN_Msk	444
6.69.2.6	CoreDebug_DEMCR_MON_EN_Pos	444
6.69.2.7	CoreDebug_DEMCR_MON_PEND_Msk	445
6.69.2.8	CoreDebug_DEMCR_MON_PEND_Pos	445
6.69.2.9	CoreDebug_DEMCR_MON_REQ_Msk	445
6.69.2.10	CoreDebug_DEMCR_MON_REQ_Pos	445
6.69.2.11	CoreDebug_DEMCR_MON_STEP_Msk	445
6.69.2.12	CoreDebug_DEMCR_MON_STEP_Pos	445
6.69.2.13	CoreDebug_DEMCR_TRCENA_Msk	445
6.69.2.14	CoreDebug_DEMCR_TRCENA_Pos	445
6.69.2.15	CoreDebug_DEMCR_VC_BUSERR_Msk	445
6.69.2.16	CoreDebug_DEMCR_VC_BUSERR_Pos	446
6.69.2.17	CoreDebug_DEMCR_VC_CHKERR_Msk	446
6.69.2.18	CoreDebug_DEMCR_VC_CHKERR_Pos	446
6.69.2.19	CoreDebug_DEMCR_VC_CORERESET_Msk	446

6.69.2.20	CoreDebug_DEMCR_VC_CORERESET_Pos	446
6.69.2.21	CoreDebug_DEMCR_VC_HARDERR_Msk	446
6.69.2.22	CoreDebug_DEMCR_VC_HARDERR_Pos	446
6.69.2.23	CoreDebug_DEMCR_VC_INTERR_Msk	446
6.69.2.24	CoreDebug_DEMCR_VC_INTERR_Pos	446
6.69.2.25	CoreDebug_DEMCR_VC_MMERR_Msk	447
6.69.2.26	CoreDebug_DEMCR_VC_MMERR_Pos	447
6.69.2.27	CoreDebug_DEMCR_VC_NOCERR_Msk	447
6.69.2.28	CoreDebug_DEMCR_VC_NOCERR_Pos	447
6.69.2.29	CoreDebug_DEMCR_VC_STATERR_Msk	447
6.69.2.30	CoreDebug_DEMCR_VC_STATERR_Pos	447
6.69.2.31	CoreDebug_DHCSR_C_DEBUGEN_Msk	447
6.69.2.32	CoreDebug_DHCSR_C_DEBUGEN_Pos	447
6.69.2.33	CoreDebug_DHCSR_C_HALT_Msk	447
6.69.2.34	CoreDebug_DHCSR_C_HALT_Pos	448
6.69.2.35	CoreDebug_DHCSR_C_MASKINTS_Msk	448
6.69.2.36	CoreDebug_DHCSR_C_MASKINTS_Pos	448
6.69.2.37	CoreDebug_DHCSR_C_SNAPSTALL_Msk	448
6.69.2.38	CoreDebug_DHCSR_C_SNAPSTALL_Pos	448
6.69.2.39	CoreDebug_DHCSR_C_STEP_Msk	448
6.69.2.40	CoreDebug_DHCSR_C_STEP_Pos	448
6.69.2.41	CoreDebug_DHCSR_DBGKEY_Msk	448
6.69.2.42	CoreDebug_DHCSR_DBGKEY_Pos	448
6.69.2.43	CoreDebug_DHCSR_S_HALT_Msk	449
6.69.2.44	CoreDebug_DHCSR_S_HALT_Pos	449
6.69.2.45	CoreDebug_DHCSR_S_LOCKUP_Msk	449
6.69.2.46	CoreDebug_DHCSR_S_LOCKUP_Pos	449
6.69.2.47	CoreDebug_DHCSR_S_REGRDY_Msk	449
6.69.2.48	CoreDebug_DHCSR_S_REGRDY_Pos	449
6.69.2.49	CoreDebug_DHCSR_S_RESET_ST_Msk	449
6.69.2.50	CoreDebug_DHCSR_S_RESET_ST_Pos	449
6.69.2.51	CoreDebug_DHCSR_S_RETIRE_ST_Msk	449
6.69.2.52	CoreDebug_DHCSR_S_RETIRE_ST_Pos	450
6.69.2.53	CoreDebug_DHCSR_S_SLEEP_Msk	450
6.69.2.54	CoreDebug_DHCSR_S_SLEEP_Pos	450
6.70	Core Definitions	451
6.70.1	Detailed Description	451
6.70.2	Macro Definition Documentation	451
6.70.2.1	CoreDebug	451
6.70.2.2	CoreDebug_BASE	451

6.70.2.3	DWT	451
6.70.2.4	DWT_BASE	452
6.70.2.5	ITM	452
6.70.2.6	ITM_BASE	452
6.70.2.7	NVIC	452
6.70.2.8	NVIC	452
6.70.2.9	NVIC_BASE	452
6.70.2.10	NVIC_BASE	452
6.70.2.11	SCB	452
6.70.2.12	SCB	452
6.70.2.13	SCB_BASE	453
6.70.2.14	SCB_BASE	453
6.70.2.15	SCnSCB	453
6.70.2.16	SCS_BASE	453
6.70.2.17	SCS_BASE	453
6.70.2.18	SysTick	453
6.70.2.19	SysTick	453
6.70.2.20	SysTick_BASE	453
6.70.2.21	SysTick_BASE	453
6.70.2.22	TPI	454
6.70.2.23	TPI_BASE	454
6.71	Data Watchpoint and Trace (DWT)	455
6.71.1	Detailed Description	455
6.71.2	Macro Definition Documentation	456
6.71.2.1	DWT_CPICNT_CPICNT_Msk	456
6.71.2.2	DWT_CPICNT_CPICNT_Pos	456
6.71.2.3	DWT_CTRL_CPIEVTENA_Msk	456
6.71.2.4	DWT_CTRL_CPIEVTENA_Pos	456
6.71.2.5	DWT_CTRL_CYCCNTENA_Msk	457
6.71.2.6	DWT_CTRL_CYCCNTENA_Pos	457
6.71.2.7	DWT_CTRL_CYCEVTENA_Msk	457
6.71.2.8	DWT_CTRL_CYCEVTENA_Pos	457
6.71.2.9	DWT_CTRL_CYCTAP_Msk	457
6.71.2.10	DWT_CTRL_CYCTAP_Pos	457
6.71.2.11	DWT_CTRL_EXCEVTENA_Msk	457
6.71.2.12	DWT_CTRL_EXCEVTENA_Pos	457
6.71.2.13	DWT_CTRL_EXCTRCENA_Msk	457
6.71.2.14	DWT_CTRL_EXCTRCENA_Pos	458
6.71.2.15	DWT_CTRL_FOLDEVTENA_Msk	458
6.71.2.16	DWT_CTRL_FOLDEVTENA_Pos	458

6.71.2.17 DWT_CTRL_LSUEVTENA_Msk . . . . .	458
6.71.2.18 DWT_CTRL_LSUEVTENA_Pos . . . . .	458
6.71.2.19 DWT_CTRL_NOCYCCNT_Msk . . . . .	458
6.71.2.20 DWT_CTRL_NOCYCCNT_Pos . . . . .	458
6.71.2.21 DWT_CTRL_NOEXTTRIG_Msk . . . . .	458
6.71.2.22 DWT_CTRL_NOEXTTRIG_Pos . . . . .	458
6.71.2.23 DWT_CTRL_NOPRFCNT_Msk . . . . .	459
6.71.2.24 DWT_CTRL_NOPRFCNT_Pos . . . . .	459
6.71.2.25 DWT_CTRL_NOTRCPKT_Msk . . . . .	459
6.71.2.26 DWT_CTRL_NOTRCPKT_Pos . . . . .	459
6.71.2.27 DWT_CTRL_NUMCOMP_Msk . . . . .	459
6.71.2.28 DWT_CTRL_NUMCOMP_Pos . . . . .	459
6.71.2.29 DWT_CTRL_PCSAMPLENA_Msk . . . . .	459
6.71.2.30 DWT_CTRL_PCSAMPLENA_Pos . . . . .	459
6.71.2.31 DWT_CTRL_POSTINIT_Msk . . . . .	459
6.71.2.32 DWT_CTRL_POSTINIT_Pos . . . . .	460
6.71.2.33 DWT_CTRL_POSTPRESET_Msk . . . . .	460
6.71.2.34 DWT_CTRL_POSTPRESET_Pos . . . . .	460
6.71.2.35 DWT_CTRL_SLEEPEVTENA_Msk . . . . .	460
6.71.2.36 DWT_CTRL_SLEEPEVTENA_Pos . . . . .	460
6.71.2.37 DWT_CTRL_SYNCTAP_Msk . . . . .	460
6.71.2.38 DWT_CTRL_SYNCTAP_Pos . . . . .	460
6.71.2.39 DWT_EXCCNT_EXCCNT_Msk . . . . .	460
6.71.2.40 DWT_EXCCNT_EXCCNT_Pos . . . . .	460
6.71.2.41 DWT_FOLDCNT_FOLDCNT_Msk . . . . .	461
6.71.2.42 DWT_FOLDCNT_FOLDCNT_Pos . . . . .	461
6.71.2.43 DWT_FUNCTION_CYCMATCH_Msk . . . . .	461
6.71.2.44 DWT_FUNCTION_CYCMATCH_Pos . . . . .	461
6.71.2.45 DWT_FUNCTION_DATAVADDR0_Msk . . . . .	461
6.71.2.46 DWT_FUNCTION_DATAVADDR0_Pos . . . . .	461
6.71.2.47 DWT_FUNCTION_DATAVADDR1_Msk . . . . .	461
6.71.2.48 DWT_FUNCTION_DATAVADDR1_Pos . . . . .	461
6.71.2.49 DWT_FUNCTION_DATAVMATCH_Msk . . . . .	461
6.71.2.50 DWT_FUNCTION_DATAVMATCH_Pos . . . . .	462
6.71.2.51 DWT_FUNCTION_DATAVSIZE_Msk . . . . .	462
6.71.2.52 DWT_FUNCTION_DATAVSIZE_Pos . . . . .	462
6.71.2.53 DWT_FUNCTION_EMITRANGE_Msk . . . . .	462
6.71.2.54 DWT_FUNCTION_EMITRANGE_Pos . . . . .	462
6.71.2.55 DWT_FUNCTION_FUNCTION_Msk . . . . .	462
6.71.2.56 DWT_FUNCTION_FUNCTION_Pos . . . . .	462

6.71.2.57 DWT_FUNCTION_LNK1ENA_Msk . . . . .	462
6.71.2.58 DWT_FUNCTION_LNK1ENA_Pos . . . . .	462
6.71.2.59 DWT_FUNCTION_MATCHED_Msk . . . . .	463
6.71.2.60 DWT_FUNCTION_MATCHED_Pos . . . . .	463
6.71.2.61 DWT_LSUCNT_LSUCNT_Msk . . . . .	463
6.71.2.62 DWT_LSUCNT_LSUCNT_Pos . . . . .	463
6.71.2.63 DWT_MASK_MASK_Msk . . . . .	463
6.71.2.64 DWT_MASK_MASK_Pos . . . . .	463
6.71.2.65 DWT_SLEEPCNT_SLEEPCNT_Msk . . . . .	463
6.71.2.66 DWT_SLEEPCNT_SLEEPCNT_Pos . . . . .	463
6.72 Defines and Type Definitions . . . . .	464
6.72.1 Detailed Description . . . . .	464
6.73 Functions and Instructions Reference . . . . .	465
6.73.1 Detailed Description . . . . .	465
6.74 IAR EWARM support in LPCOpen . . . . .	466
6.75 ITM Functions . . . . .	467
6.75.1 Detailed Description . . . . .	467
6.75.2 Macro Definition Documentation . . . . .	467
6.75.2.1 ITM_RXBUFFER_EMPTY . . . . .	467
6.75.3 Function Documentation . . . . .	467
6.75.3.1 ITM_CheckChar(void) . . . . .	467
6.75.3.2 ITM_ReceiveChar(void) . . . . .	467
6.75.3.3 ITM_SendChar(uint32_t ch) . . . . .	468
6.75.4 Variable Documentation . . . . .	468
6.75.4.1 ITM_RxBuffer . . . . .	468
6.76 Instrumentation Trace Macrocell (ITM) . . . . .	469
6.76.1 Detailed Description . . . . .	469
6.76.2 Macro Definition Documentation . . . . .	469
6.76.2.1 ITM_IMCR_INTEGRATION_Msk . . . . .	469
6.76.2.2 ITM_IMCR_INTEGRATION_Pos . . . . .	470
6.76.2.3 ITM_IRR_ATREADYM_Msk . . . . .	470
6.76.2.4 ITM_IRR_ATREADYM_Pos . . . . .	470
6.76.2.5 ITM_IWR_ATVALIDM_Msk . . . . .	470
6.76.2.6 ITM_IWR_ATVALIDM_Pos . . . . .	470
6.76.2.7 ITM_LSR_Access_Msk . . . . .	470
6.76.2.8 ITM_LSR_Access_Pos . . . . .	470
6.76.2.9 ITM_LSR_ByteAcc_Msk . . . . .	470
6.76.2.10 ITM_LSR_ByteAcc_Pos . . . . .	470
6.76.2.11 ITM_LSR_Present_Msk . . . . .	471
6.76.2.12 ITM_LSR_Present_Pos . . . . .	471

6.76.2.13	ITM_TCR_BUSY_Msk . . . . .	471
6.76.2.14	ITM_TCR_BUSY_Pos . . . . .	471
6.76.2.15	ITM_TCR_DWTENA_Msk . . . . .	471
6.76.2.16	ITM_TCR_DWTENA_Pos . . . . .	471
6.76.2.17	ITM_TCR_GTSFREQ_Msk . . . . .	471
6.76.2.18	ITM_TCR_GTSFREQ_Pos . . . . .	471
6.76.2.19	ITM_TCR_ITMENA_Msk . . . . .	471
6.76.2.20	ITM_TCR_ITMENA_Pos . . . . .	472
6.76.2.21	ITM_TCR_SWOENA_Msk . . . . .	472
6.76.2.22	ITM_TCR_SWOENA_Pos . . . . .	472
6.76.2.23	ITM_TCR_SYNCENA_Msk . . . . .	472
6.76.2.24	ITM_TCR_SYNCENA_Pos . . . . .	472
6.76.2.25	ITM_TCR_TraceBusID_Msk . . . . .	472
6.76.2.26	ITM_TCR_TraceBusID_Pos . . . . .	472
6.76.2.27	ITM_TCR_TSENA_Msk . . . . .	472
6.76.2.28	ITM_TCR_TSENA_Pos . . . . .	472
6.76.2.29	ITM_TCR_TSPrescale_Msk . . . . .	473
6.76.2.30	ITM_TCR_TSPrescale_Pos . . . . .	473
6.76.2.31	ITM_TPR_PRIVMASK_Msk . . . . .	473
6.76.2.32	ITM_TPR_PRIVMASK_Pos . . . . .	473
6.77	Keil uVision support in LPCOpen . . . . .	474
6.78	LPC Public Macros . . . . .	475
6.78.1	Detailed Description . . . . .	475
6.78.2	Macro Definition Documentation . . . . .	475
6.78.2.1	_BIT . . . . .	475
6.78.2.2	_BITMASK . . . . .	475
6.78.2.3	_SBF . . . . .	475
6.78.2.4	EXTERN . . . . .	475
6.78.2.5	MAX . . . . .	475
6.78.2.6	MIN . . . . .	475
6.78.2.7	NELEMENTS . . . . .	475
6.78.2.8	NULL . . . . .	475
6.78.2.9	STATIC . . . . .	476
6.79	LPC Public Types . . . . .	477
6.79.1	Detailed Description . . . . .	477
6.79.2	Macro Definition Documentation . . . . .	477
6.79.2.1	ALIGN . . . . .	477
6.79.2.2	INLINE . . . . .	477
6.79.2.3	PARAM_FUNCTIONALSTATE . . . . .	478
6.79.2.4	PARAM_SETSTATE . . . . .	478



6.79.2.5	WEAK	478
6.79.3	Typedef Documentation	478
6.79.3.1	BOOL_16	478
6.79.3.2	BOOL_32	478
6.79.3.3	BOOL_8	478
6.79.3.4	CHAR	478
6.79.3.5	INT_16	478
6.79.3.6	INT_32	478
6.79.3.7	INT_64	478
6.79.3.8	INT_8	479
6.79.3.9	IntStatus	479
6.79.3.10	PFI	479
6.79.3.11	PFV	479
6.79.3.12	SetState	479
6.79.3.13	UNS_16	479
6.79.3.14	UNS_32	479
6.79.3.15	UNS_64	479
6.79.3.16	UNS_8	479
6.79.4	Enumeration Type Documentation	479
6.79.4.1	Bool	479
6.79.4.2	FlagStatus	480
6.79.4.3	FunctionalState	480
6.79.4.4	Status	480
6.79.4.5	TRANSFER_BLOCK_T	480
6.80	LPC5411X multi-core use in LPCOpen	481
6.81	LPC5411x Chip specific drivers	483
6.81.1	Detailed Description	483
6.82	LPCOpen download and installation information	484
6.82.1	Detailed Description	484
6.83	LPCOpen versioning and release history	485
6.84	NVIC Functions	486
6.84.1	Detailed Description	486
6.84.2	Macro Definition Documentation	486
6.84.2.1	_BIT_SHIFT	486
6.84.2.2	_IP_IDX	487
6.84.2.3	_SHP_IDX	487
6.84.3	Function Documentation	487
6.84.3.1	NVIC_ClearPendingIRQ(IRQn_Type IRQn)	487
6.84.3.2	NVIC_DecodePriority(uint32_t Priority, uint32_t PriorityGroup, uint32_t *p← PreemptPriority, uint32_t *pSubPriority)	487

6.84.3.3	NVIC_DisableIRQ(IRQn_Type IRQn)	487
6.84.3.4	NVIC_EnableIRQ(IRQn_Type IRQn)	487
6.84.3.5	NVIC_EncodePriority(uint32_t PriorityGroup, uint32_t PreemptPriority, uint32_t SubPriority)	488
6.84.3.6	NVIC_GetActive(IRQn_Type IRQn)	488
6.84.3.7	NVIC_GetPendingIRQ(IRQn_Type IRQn)	488
6.84.3.8	NVIC_GetPriority(IRQn_Type IRQn)	489
6.84.3.9	NVIC_GetPriorityGrouping(void)	489
6.84.3.10	NVIC_SetPendingIRQ(IRQn_Type IRQn)	489
6.84.3.11	NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)	489
6.84.3.12	NVIC_SetPriorityGrouping(uint32_t PriorityGroup)	490
6.84.3.13	NVIC_SystemReset(void)	490
6.85	NXP LPCXpresso LPC54114 LQFP board	491
6.86	Nested Vectored Interrupt Controller (NVIC)	492
6.86.1	Detailed Description	492
6.86.2	Macro Definition Documentation	492
6.86.2.1	NVIC_STIR_INTID_Msk	492
6.86.2.2	NVIC_STIR_INTID_Pos	492
6.87	RTOS support code	493
6.87.1	Detailed Description	493
6.88	RTOS: FreeRTOS support code	494
6.88.1	Detailed Description	494
6.89	Status and Control Registers	495
6.89.1	Detailed Description	495
6.90	Supported toolchains in LPCOpen	496
6.90.1	Detailed Description	496
6.91	SysTick Functions	497
6.91.1	Detailed Description	497
6.91.2	Function Documentation	497
6.91.2.1	SysTick_Config(uint32_t ticks)	497
6.92	System Control Block (SCB)	498
6.92.1	Detailed Description	498
6.92.2	Macro Definition Documentation	501
6.92.2.1	SCB_AIRCR_ENDIANESS_Msk	501
6.92.2.2	SCB_AIRCR_ENDIANESS_Msk	501
6.92.2.3	SCB_AIRCR_ENDIANESS_Pos	501
6.92.2.4	SCB_AIRCR_ENDIANESS_Pos	501
6.92.2.5	SCB_AIRCR_PRIGROUP_Msk	501
6.92.2.6	SCB_AIRCR_PRIGROUP_Pos	501
6.92.2.7	SCB_AIRCR_SYSRESETREQ_Msk	502

6.92.2.8	SCB_AIRCR_SYSRESETREQ_Msk	502
6.92.2.9	SCB_AIRCR_SYSRESETREQ_Pos	502
6.92.2.10	SCB_AIRCR_SYSRESETREQ_Pos	502
6.92.2.11	SCB_AIRCR_VECTCLRACTIVE_Msk	502
6.92.2.12	SCB_AIRCR_VECTCLRACTIVE_Msk	502
6.92.2.13	SCB_AIRCR_VECTCLRACTIVE_Pos	502
6.92.2.14	SCB_AIRCR_VECTCLRACTIVE_Pos	502
6.92.2.15	SCB_AIRCR_VECTKEY_Msk	502
6.92.2.16	SCB_AIRCR_VECTKEY_Msk	503
6.92.2.17	SCB_AIRCR_VECTKEY_Pos	503
6.92.2.18	SCB_AIRCR_VECTKEY_Pos	503
6.92.2.19	SCB_AIRCR_VECTKEYSTAT_Msk	503
6.92.2.20	SCB_AIRCR_VECTKEYSTAT_Msk	503
6.92.2.21	SCB_AIRCR_VECTKEYSTAT_Pos	503
6.92.2.22	SCB_AIRCR_VECTKEYSTAT_Pos	503
6.92.2.23	SCB_AIRCR_VECTRESET_Msk	503
6.92.2.24	SCB_AIRCR_VECTRESET_Pos	503
6.92.2.25	SCB_CCR_BFHFNMIGN_Msk	504
6.92.2.26	SCB_CCR_BFHFNMIGN_Pos	504
6.92.2.27	SCB_CCR_DIV_0_TRP_Msk	504
6.92.2.28	SCB_CCR_DIV_0_TRP_Pos	504
6.92.2.29	SCB_CCR_NONBASETHRDENA_Msk	504
6.92.2.30	SCB_CCR_NONBASETHRDENA_Pos	504
6.92.2.31	SCB_CCR_STKALIGN_Msk	504
6.92.2.32	SCB_CCR_STKALIGN_Msk	504
6.92.2.33	SCB_CCR_STKALIGN_Pos	504
6.92.2.34	SCB_CCR_STKALIGN_Pos	505
6.92.2.35	SCB_CCR_UNALIGN_TRP_Msk	505
6.92.2.36	SCB_CCR_UNALIGN_TRP_Msk	505
6.92.2.37	SCB_CCR_UNALIGN_TRP_Pos	505
6.92.2.38	SCB_CCR_UNALIGN_TRP_Pos	505
6.92.2.39	SCB_CCR_USERSETMPEND_Msk	505
6.92.2.40	SCB_CCR_USERSETMPEND_Pos	505
6.92.2.41	SCB_CFSR_BUSFAULTSR_Msk	505
6.92.2.42	SCB_CFSR_BUSFAULTSR_Pos	505
6.92.2.43	SCB_CFSR_MEMFAULTSR_Msk	506
6.92.2.44	SCB_CFSR_MEMFAULTSR_Pos	506
6.92.2.45	SCB_CFSR_USGFAULTSR_Msk	506
6.92.2.46	SCB_CFSR_USGFAULTSR_Pos	506
6.92.2.47	SCB_CPUID_ARCHITECTURE_Msk	506

6.92.2.48 SCB_CPUID_ARCHITECTURE_Msk . . . . .	506
6.92.2.49 SCB_CPUID_ARCHITECTURE_Pos . . . . .	506
6.92.2.50 SCB_CPUID_ARCHITECTURE_Pos . . . . .	506
6.92.2.51 SCB_CPUID_IMPLEMENTER_Msk . . . . .	506
6.92.2.52 SCB_CPUID_IMPLEMENTER_Msk . . . . .	507
6.92.2.53 SCB_CPUID_IMPLEMENTER_Pos . . . . .	507
6.92.2.54 SCB_CPUID_IMPLEMENTER_Pos . . . . .	507
6.92.2.55 SCB_CPUID_PARTNO_Msk . . . . .	507
6.92.2.56 SCB_CPUID_PARTNO_Msk . . . . .	507
6.92.2.57 SCB_CPUID_PARTNO_Pos . . . . .	507
6.92.2.58 SCB_CPUID_PARTNO_Pos . . . . .	507
6.92.2.59 SCB_CPUID_REVISION_Msk . . . . .	507
6.92.2.60 SCB_CPUID_REVISION_Msk . . . . .	507
6.92.2.61 SCB_CPUID_REVISION_Pos . . . . .	508
6.92.2.62 SCB_CPUID_REVISION_Pos . . . . .	508
6.92.2.63 SCB_CPUID_VARIANT_Msk . . . . .	508
6.92.2.64 SCB_CPUID_VARIANT_Msk . . . . .	508
6.92.2.65 SCB_CPUID_VARIANT_Pos . . . . .	508
6.92.2.66 SCB_CPUID_VARIANT_Pos . . . . .	508
6.92.2.67 SCB_DFSR_BKPT_Msk . . . . .	508
6.92.2.68 SCB_DFSR_BKPT_Pos . . . . .	508
6.92.2.69 SCB_DFSR_DWTTRAP_Msk . . . . .	508
6.92.2.70 SCB_DFSR_DWTTRAP_Pos . . . . .	509
6.92.2.71 SCB_DFSR_EXTERNAL_Msk . . . . .	509
6.92.2.72 SCB_DFSR_EXTERNAL_Pos . . . . .	509
6.92.2.73 SCB_DFSR_HALTED_Msk . . . . .	509
6.92.2.74 SCB_DFSR_HALTED_Pos . . . . .	509
6.92.2.75 SCB_DFSR_VCATCH_Msk . . . . .	509
6.92.2.76 SCB_DFSR_VCATCH_Pos . . . . .	509
6.92.2.77 SCB_HFSR_DEBUGEVT_Msk . . . . .	509
6.92.2.78 SCB_HFSR_DEBUGEVT_Pos . . . . .	509
6.92.2.79 SCB_HFSR_FORCED_Msk . . . . .	510
6.92.2.80 SCB_HFSR_FORCED_Pos . . . . .	510
6.92.2.81 SCB_HFSR_VECTTBL_Msk . . . . .	510
6.92.2.82 SCB_HFSR_VECTTBL_Pos . . . . .	510
6.92.2.83 SCB_ICSR_ISRPENDING_Msk . . . . .	510
6.92.2.84 SCB_ICSR_ISRPENDING_Msk . . . . .	510
6.92.2.85 SCB_ICSR_ISRPENDING_Pos . . . . .	510
6.92.2.86 SCB_ICSR_ISRPENDING_Pos . . . . .	510
6.92.2.87 SCB_ICSR_ISRPREEMPT_Msk . . . . .	510

6.92.2.88 SCB_ICSR_ISRPREEMPT_Msk	511
6.92.2.89 SCB_ICSR_ISRPREEMPT_Pos	511
6.92.2.90 SCB_ICSR_ISRPREEMPT_Pos	511
6.92.2.91 SCB_ICSR_NMIPENDSET_Msk	511
6.92.2.92 SCB_ICSR_NMIPENDSET_Msk	511
6.92.2.93 SCB_ICSR_NMIPENDSET_Pos	511
6.92.2.94 SCB_ICSR_NMIPENDSET_Pos	511
6.92.2.95 SCB_ICSR_PENDSTCLR_Msk	511
6.92.2.96 SCB_ICSR_PENDSTCLR_Msk	511
6.92.2.97 SCB_ICSR_PENDSTCLR_Pos	512
6.92.2.98 SCB_ICSR_PENDSTCLR_Pos	512
6.92.2.99 SCB_ICSR_PENDSTSET_Msk	512
6.92.2.100 SCB_ICSR_PENDSTSET_Msk	512
6.92.2.101 SCB_ICSR_PENDSTSET_Pos	512
6.92.2.102 SCB_ICSR_PENDSTSET_Pos	512
6.92.2.103 SCB_ICSR_PENDSVCLR_Msk	512
6.92.2.104 SCB_ICSR_PENDSVCLR_Msk	512
6.92.2.105 SCB_ICSR_PENDSVCLR_Pos	512
6.92.2.106 SCB_ICSR_PENDSVCLR_Pos	513
6.92.2.107 SCB_ICSR_PENDSVSET_Msk	513
6.92.2.108 SCB_ICSR_PENDSVSET_Msk	513
6.92.2.109 SCB_ICSR_PENDSVSET_Pos	513
6.92.2.110 SCB_ICSR_PENDSVSET_Pos	513
6.92.2.111 SCB_ICSR_RETTOBASE_Msk	513
6.92.2.112 SCB_ICSR_RETTOBASE_Pos	513
6.92.2.113 SCB_ICSR_VECTACTIVE_Msk	513
6.92.2.114 SCB_ICSR_VECTACTIVE_Msk	513
6.92.2.115 SCB_ICSR_VECTACTIVE_Pos	514
6.92.2.116 SCB_ICSR_VECTACTIVE_Pos	514
6.92.2.117 SCB_ICSR_VECTPENDING_Msk	514
6.92.2.118 SCB_ICSR_VECTPENDING_Msk	514
6.92.2.119 SCB_ICSR_VECTPENDING_Pos	514
6.92.2.120 SCB_ICSR_VECTPENDING_Pos	514
6.92.2.121 SCB_SCR_SEVONPEND_Msk	514
6.92.2.122 SCB_SCR_SEVONPEND_Msk	514
6.92.2.123 SCB_SCR_SEVONPEND_Pos	514
6.92.2.124 SCB_SCR_SEVONPEND_Pos	515
6.92.2.125 SCB_SCR_SLEEPDEEP_Msk	515
6.92.2.126 SCB_SCR_SLEEPDEEP_Msk	515
6.92.2.127 SCB_SCR_SLEEPDEEP_Pos	515

6.92.2.128	SCB_SCR_SLEEPDEEP_Pos	515
6.92.2.129	SCB_SCR_SLEEPONEXIT_Msk	515
6.92.2.130	SCB_SCR_SLEEPONEXIT_Msk	515
6.92.2.131	SCB_SCR_SLEEPONEXIT_Pos	515
6.92.2.132	SCB_SCR_SLEEPONEXIT_Pos	515
6.92.2.133	SCB_SHCSR_BUSFAULTACT_Msk	516
6.92.2.134	SCB_SHCSR_BUSFAULTACT_Pos	516
6.92.2.135	SCB_SHCSR_BUSFAULTENA_Msk	516
6.92.2.136	SCB_SHCSR_BUSFAULTENA_Pos	516
6.92.2.137	SCB_SHCSR_BUSFAULTPENDEDED_Msk	516
6.92.2.138	SCB_SHCSR_BUSFAULTPENDEDED_Pos	516
6.92.2.139	SCB_SHCSR_MEMFAULTACT_Msk	516
6.92.2.140	SCB_SHCSR_MEMFAULTACT_Pos	516
6.92.2.141	SCB_SHCSR_MEMFAULTENA_Msk	516
6.92.2.142	SCB_SHCSR_MEMFAULTENA_Pos	517
6.92.2.143	SCB_SHCSR_MEMFAULTPENDEDED_Msk	517
6.92.2.144	SCB_SHCSR_MEMFAULTPENDEDED_Pos	517
6.92.2.145	SCB_SHCSR_MONITORACT_Msk	517
6.92.2.146	SCB_SHCSR_MONITORACT_Pos	517
6.92.2.147	SCB_SHCSR_PENDSVACT_Msk	517
6.92.2.148	SCB_SHCSR_PENDSVACT_Pos	517
6.92.2.149	SCB_SHCSR_SVCALLACT_Msk	517
6.92.2.150	SCB_SHCSR_SVCALLACT_Pos	517
6.92.2.151	SCB_SHCSR_SVCALLPENDEDED_Msk	518
6.92.2.152	SCB_SHCSR_SVCALLPENDEDED_Msk	518
6.92.2.153	SCB_SHCSR_SVCALLPENDEDED_Pos	518
6.92.2.154	SCB_SHCSR_SVCALLPENDEDED_Pos	518
6.92.2.155	SCB_SHCSR_SYSTICKACT_Msk	518
6.92.2.156	SCB_SHCSR_SYSTICKACT_Pos	518
6.92.2.157	SCB_SHCSR_USGFAULTACT_Msk	518
6.92.2.158	SCB_SHCSR_USGFAULTACT_Pos	518
6.92.2.159	SCB_SHCSR_USGFAULTENA_Msk	518
6.92.2.160	SCB_SHCSR_USGFAULTENA_Pos	519
6.92.2.161	SCB_SHCSR_USGFAULTPENDEDED_Msk	519
6.92.2.162	SCB_SHCSR_USGFAULTPENDEDED_Pos	519
6.92.2.163	SCB_VTOR_TBLOFF_Msk	519
6.92.2.164	SCB_VTOR_TBLOFF_Pos	519
6.93	System Controls not in SCB (SCnSCB)	520
6.93.1	Detailed Description	520
6.93.2	Macro Definition Documentation	520

6.93.2.1	SCnSCB_ACTLR_DISDEFWBUFF_Msk	520
6.93.2.2	SCnSCB_ACTLR_DISDEFWBUFF_Pos	520
6.93.2.3	SCnSCB_ACTLR_DISFOLD_Msk	520
6.93.2.4	SCnSCB_ACTLR_DISFOLD_Pos	520
6.93.2.5	SCnSCB_ACTLR_DISFPCA_Msk	521
6.93.2.6	SCnSCB_ACTLR_DISFPCA_Pos	521
6.93.2.7	SCnSCB_ACTLR_DISMCYCINT_Msk	521
6.93.2.8	SCnSCB_ACTLR_DISMCYCINT_Pos	521
6.93.2.9	SCnSCB_ACTLR_DISOOFPP_Msk	521
6.93.2.10	SCnSCB_ACTLR_DISOOFPP_Pos	521
6.93.2.11	SCnSCB_ICTR_INTLINESNUM_Msk	521
6.93.2.12	SCnSCB_ICTR_INTLINESNUM_Pos	521
6.94	System Tick Timer (SysTick)	522
6.94.1	Detailed Description	522
6.94.2	Macro Definition Documentation	523
6.94.2.1	SysTick_CALIB_NOREF_Msk	523
6.94.2.2	SysTick_CALIB_NOREF_Pos	523
6.94.2.3	SysTick_CALIB_NOREF_Pos	523
6.94.2.4	SysTick_CALIB_NOREF_Pos	523
6.94.2.5	SysTick_CALIB_SKEW_Msk	523
6.94.2.6	SysTick_CALIB_SKEW_Pos	523
6.94.2.7	SysTick_CALIB_SKEW_Pos	523
6.94.2.8	SysTick_CALIB_SKEW_Pos	523
6.94.2.9	SysTick_CALIB_TENMS_Msk	523
6.94.2.10	SysTick_CALIB_TENMS_Pos	524
6.94.2.11	SysTick_CALIB_TENMS_Pos	524
6.94.2.12	SysTick_CALIB_TENMS_Pos	524
6.94.2.13	SysTick_CTRL_CLKSOURCE_Msk	524
6.94.2.14	SysTick_CTRL_CLKSOURCE_Pos	524
6.94.2.15	SysTick_CTRL_CLKSOURCE_Pos	524
6.94.2.16	SysTick_CTRL_CLKSOURCE_Pos	524
6.94.2.17	SysTick_CTRL_COUNTFLAG_Msk	524
6.94.2.18	SysTick_CTRL_COUNTFLAG_Pos	524
6.94.2.19	SysTick_CTRL_COUNTFLAG_Pos	525
6.94.2.20	SysTick_CTRL_COUNTFLAG_Pos	525
6.94.2.21	SysTick_CTRL_ENABLE_Msk	525
6.94.2.22	SysTick_CTRL_ENABLE_Pos	525
6.94.2.23	SysTick_CTRL_ENABLE_Pos	525
6.94.2.24	SysTick_CTRL_ENABLE_Pos	525
6.94.2.25	SysTick_CTRL_TICKINT_Msk	525

6.94.2.26 SysTick_CTRL_TICKINT_Msk . . . . .	525
6.94.2.27 SysTick_CTRL_TICKINT_Pos . . . . .	525
6.94.2.28 SysTick_CTRL_TICKINT_Pos . . . . .	526
6.94.2.29 SysTick_LOAD_RELOAD_Msk . . . . .	526
6.94.2.30 SysTick_LOAD_RELOAD_Msk . . . . .	526
6.94.2.31 SysTick_LOAD_RELOAD_Pos . . . . .	526
6.94.2.32 SysTick_LOAD_RELOAD_Pos . . . . .	526
6.94.2.33 SysTick_VAL_CURRENT_Msk . . . . .	526
6.94.2.34 SysTick_VAL_CURRENT_Msk . . . . .	526
6.94.2.35 SysTick_VAL_CURRENT_Pos . . . . .	526
6.94.2.36 SysTick_VAL_CURRENT_Pos . . . . .	526
6.95 Trace Port Interface (TPI) . . . . .	527
6.95.1 Detailed Description . . . . .	527
6.95.2 Macro Definition Documentation . . . . .	528
6.95.2.1 TPI_ACPR_PRESCALER_Msk . . . . .	528
6.95.2.2 TPI_ACPR_PRESCALER_Pos . . . . .	528
6.95.2.3 TPI_DEVID_AsynCkIn_Msk . . . . .	528
6.95.2.4 TPI_DEVID_AsynCkIn_Pos . . . . .	528
6.95.2.5 TPI_DEVID_MANCVALID_Msk . . . . .	529
6.95.2.6 TPI_DEVID_MANCVALID_Pos . . . . .	529
6.95.2.7 TPI_DEVID_MinBufSz_Msk . . . . .	529
6.95.2.8 TPI_DEVID_MinBufSz_Pos . . . . .	529
6.95.2.9 TPI_DEVID_NrTraceInput_Msk . . . . .	529
6.95.2.10 TPI_DEVID_NrTraceInput_Pos . . . . .	529
6.95.2.11 TPI_DEVID_NRZVALID_Msk . . . . .	529
6.95.2.12 TPI_DEVID_NRZVALID_Pos . . . . .	529
6.95.2.13 TPI_DEVID_PTINVALID_Msk . . . . .	529
6.95.2.14 TPI_DEVID_PTINVALID_Pos . . . . .	530
6.95.2.15 TPI_DEVTYPE_MajorType_Msk . . . . .	530
6.95.2.16 TPI_DEVTYPE_MajorType_Pos . . . . .	530
6.95.2.17 TPI_DEVTYPE_SubType_Msk . . . . .	530
6.95.2.18 TPI_DEVTYPE_SubType_Pos . . . . .	530
6.95.2.19 TPI_FFCR_EnFCont_Msk . . . . .	530
6.95.2.20 TPI_FFCR_EnFCont_Pos . . . . .	530
6.95.2.21 TPI_FFCR_TrigIn_Msk . . . . .	530
6.95.2.22 TPI_FFCR_TrigIn_Pos . . . . .	530
6.95.2.23 TPI_FFSR_FlInProg_Msk . . . . .	531
6.95.2.24 TPI_FFSR_FlInProg_Pos . . . . .	531
6.95.2.25 TPI_FFSR_FtNonStop_Msk . . . . .	531
6.95.2.26 TPI_FFSR_FtNonStop_Pos . . . . .	531



6.95.2.27 TPI_FFSR_FtStopped_Msk . . . . .	531
6.95.2.28 TPI_FFSR_FtStopped_Pos . . . . .	531
6.95.2.29 TPI_FFSR_TCPresent_Msk . . . . .	531
6.95.2.30 TPI_FFSR_TCPresent_Pos . . . . .	531
6.95.2.31 TPI_FIFO0_ETM0_Msk . . . . .	531
6.95.2.32 TPI_FIFO0_ETM0_Pos . . . . .	532
6.95.2.33 TPI_FIFO0_ETM1_Msk . . . . .	532
6.95.2.34 TPI_FIFO0_ETM1_Pos . . . . .	532
6.95.2.35 TPI_FIFO0_ETM2_Msk . . . . .	532
6.95.2.36 TPI_FIFO0_ETM2_Pos . . . . .	532
6.95.2.37 TPI_FIFO0_ETM_ATVALID_Msk . . . . .	532
6.95.2.38 TPI_FIFO0_ETM_ATVALID_Pos . . . . .	532
6.95.2.39 TPI_FIFO0_ETM_bytecount_Msk . . . . .	532
6.95.2.40 TPI_FIFO0_ETM_bytecount_Pos . . . . .	532
6.95.2.41 TPI_FIFO0_ITM_ATVALID_Msk . . . . .	533
6.95.2.42 TPI_FIFO0_ITM_ATVALID_Pos . . . . .	533
6.95.2.43 TPI_FIFO0_ITM_bytecount_Msk . . . . .	533
6.95.2.44 TPI_FIFO0_ITM_bytecount_Pos . . . . .	533
6.95.2.45 TPI_FIFO1_ETM_ATVALID_Msk . . . . .	533
6.95.2.46 TPI_FIFO1_ETM_ATVALID_Pos . . . . .	533
6.95.2.47 TPI_FIFO1_ETM_bytecount_Msk . . . . .	533
6.95.2.48 TPI_FIFO1_ETM_bytecount_Pos . . . . .	533
6.95.2.49 TPI_FIFO1_ITM0_Msk . . . . .	533
6.95.2.50 TPI_FIFO1_ITM0_Pos . . . . .	534
6.95.2.51 TPI_FIFO1_ITM1_Msk . . . . .	534
6.95.2.52 TPI_FIFO1_ITM1_Pos . . . . .	534
6.95.2.53 TPI_FIFO1_ITM2_Msk . . . . .	534
6.95.2.54 TPI_FIFO1_ITM2_Pos . . . . .	534
6.95.2.55 TPI_FIFO1_ITM_ATVALID_Msk . . . . .	534
6.95.2.56 TPI_FIFO1_ITM_ATVALID_Pos . . . . .	534
6.95.2.57 TPI_FIFO1_ITM_bytecount_Msk . . . . .	534
6.95.2.58 TPI_FIFO1_ITM_bytecount_Pos . . . . .	534
6.95.2.59 TPI_ITATBCTR0_ATREADY_Msk . . . . .	535
6.95.2.60 TPI_ITATBCTR0_ATREADY_Pos . . . . .	535
6.95.2.61 TPI_ITATBCTR2_ATREADY_Msk . . . . .	535
6.95.2.62 TPI_ITATBCTR2_ATREADY_Pos . . . . .	535
6.95.2.63 TPI_ITCTRL_Mode_Msk . . . . .	535
6.95.2.64 TPI_ITCTRL_Mode_Pos . . . . .	535
6.95.2.65 TPI_SPPR_TXMODE_Msk . . . . .	535
6.95.2.66 TPI_SPPR_TXMODE_Pos . . . . .	535

6.95.2.67 TPI_TRIGGER_TRIGGER_Msk . . . . .	535
6.95.2.68 TPI_TRIGGER_TRIGGER_Pos . . . . .	536
6.96 USB_D_Core . . . . .	537
6.96.1 Detailed Description . . . . .	537
6.96.2 Macro Definition Documentation . . . . .	537
6.96.2.1 B3VAL . . . . .	537
6.96.2.2 REQUEST_CLASS . . . . .	537
6.96.2.3 REQUEST_DEVICE_TO_HOST . . . . .	538
6.96.2.4 REQUEST_HOST_TO_DEVICE . . . . .	538
6.96.2.5 REQUEST_RESERVED . . . . .	538
6.96.2.6 REQUEST_STANDARD . . . . .	538
6.96.2.7 REQUEST_TO_DEVICE . . . . .	538
6.96.2.8 REQUEST_TO_ENDPOINT . . . . .	538
6.96.2.9 REQUEST_TO_INTERFACE . . . . .	538
6.96.2.10 REQUEST_TO_OTHER . . . . .	538
6.96.2.11 REQUEST_VENDOR . . . . .	538
6.96.2.12 USB_CONFIG_BUS_POWERED . . . . .	539
6.96.2.13 USB_CONFIG_POWER_MA . . . . .	539
6.96.2.14 USB_CONFIG_POWERED_MASK . . . . .	539
6.96.2.15 USB_CONFIG_REMOTE_WAKEUP . . . . .	539
6.96.2.16 USB_CONFIG_SELF_POWERED . . . . .	539
6.96.2.17 USB_CONFIGURATION_DESC_SIZE . . . . .	539
6.96.2.18 USB_CONFIGURATION_DESCRIPTOR_TYPE . . . . .	539
6.96.2.19 USB_DEBUG_DESCRIPTOR_TYPE . . . . .	539
6.96.2.20 USB_DEVICE_CLASS_APP . . . . .	539
6.96.2.21 USB_DEVICE_CLASS_AUDIO . . . . .	540
6.96.2.22 USB_DEVICE_CLASS_COMMUNICATIONS . . . . .	540
6.96.2.23 USB_DEVICE_CLASS_HUB . . . . .	540
6.96.2.24 USB_DEVICE_CLASS_HUMAN_INTERFACE . . . . .	540
6.96.2.25 USB_DEVICE_CLASS_MISCELLANEOUS . . . . .	540
6.96.2.26 USB_DEVICE_CLASS_MONITOR . . . . .	540
6.96.2.27 USB_DEVICE_CLASS_PHYSICAL_INTERFACE . . . . .	540
6.96.2.28 USB_DEVICE_CLASS_POWER . . . . .	540
6.96.2.29 USB_DEVICE_CLASS_PRINTER . . . . .	540
6.96.2.30 USB_DEVICE_CLASS_RESERVED . . . . .	541
6.96.2.31 USB_DEVICE_CLASS_STORAGE . . . . .	541
6.96.2.32 USB_DEVICE_CLASS_VENDOR_SPECIFIC . . . . .	541
6.96.2.33 USB_DEVICE_DESC_SIZE . . . . .	541
6.96.2.34 USB_DEVICE_DESCRIPTOR_TYPE . . . . .	541
6.96.2.35 USB_DEVICE_QUALI_SIZE . . . . .	541

6.96.2.36 USB_DEVICE_QUALIFIER_DESCRIPTOR_TYPE . . . . .	541
6.96.2.37 USB_ENDPOINT_0_HS_MAXP . . . . .	541
6.96.2.38 USB_ENDPOINT_0_LS_MAXP . . . . .	541
6.96.2.39 USB_ENDPOINT_BULK_HS_MAXP . . . . .	541
6.96.2.40 USB_ENDPOINT_DESC_SIZE . . . . .	542
6.96.2.41 USB_ENDPOINT_DESCRIPTOR_TYPE . . . . .	542
6.96.2.42 USB_ENDPOINT_DIRECTION_MASK . . . . .	542
6.96.2.43 USB_ENDPOINT_IN . . . . .	542
6.96.2.44 USB_ENDPOINT_OUT . . . . .	542
6.96.2.45 USB_ENDPOINT_SYNC_ADAPTIVE . . . . .	542
6.96.2.46 USB_ENDPOINT_SYNC_ASYNCHRONOUS . . . . .	542
6.96.2.47 USB_ENDPOINT_SYNC_MASK . . . . .	542
6.96.2.48 USB_ENDPOINT_SYNC_NO_SYNCHRONIZATION . . . . .	542
6.96.2.49 USB_ENDPOINT_SYNC_SYNCHRONOUS . . . . .	543
6.96.2.50 USB_ENDPOINT_TYPE_BULK . . . . .	543
6.96.2.51 USB_ENDPOINT_TYPE_CONTROL . . . . .	543
6.96.2.52 USB_ENDPOINT_TYPE_INTERRUPT . . . . .	543
6.96.2.53 USB_ENDPOINT_TYPE_ISOCHRONOUS . . . . .	543
6.96.2.54 USB_ENDPOINT_TYPE_MASK . . . . .	543
6.96.2.55 USB_ENDPOINT_USAGE_DATA . . . . .	543
6.96.2.56 USB_ENDPOINT_USAGE_FEEDBACK . . . . .	543
6.96.2.57 USB_ENDPOINT_USAGE_IMPLICIT_FEEDBACK . . . . .	543
6.96.2.58 USB_ENDPOINT_USAGE_MASK . . . . .	544
6.96.2.59 USB_ENDPOINT_USAGE_RESERVED . . . . .	544
6.96.2.60 USB_FEATURE_ENDPOINT_STALL . . . . .	544
6.96.2.61 USB_FEATURE_REMOTE_WAKEUP . . . . .	544
6.96.2.62 USB_FEATURE_TEST_MODE . . . . .	544
6.96.2.63 USB_GETSTATUS_ENDPOINT_STALL . . . . .	544
6.96.2.64 USB_GETSTATUS_REMOTE_WAKEUP . . . . .	544
6.96.2.65 USB_GETSTATUS_SELF_POWERED . . . . .	544
6.96.2.66 USB_INTERFACE_ASSOC_DESC_SIZE . . . . .	544
6.96.2.67 USB_INTERFACE_ASSOCIATION_DESCRIPTOR_TYPE . . . . .	545
6.96.2.68 USB_INTERFACE_DESC_SIZE . . . . .	545
6.96.2.69 USB_INTERFACE_DESCRIPTOR_TYPE . . . . .	545
6.96.2.70 USB_INTERFACE_POWER_DESCRIPTOR_TYPE . . . . .	545
6.96.2.71 USB_OTG_DESCRIPTOR_TYPE . . . . .	545
6.96.2.72 USB_OTHER_SPEED_CONF_SIZE . . . . .	545
6.96.2.73 USB_OTHER_SPEED_CONFIG_DESCRIPTOR_TYPE . . . . .	545
6.96.2.74 USB_REQUEST_CLEAR_FEATURE . . . . .	545
6.96.2.75 USB_REQUEST_GET_CONFIGURATION . . . . .	545

6.96.2.76	USB_REQUEST_GET_DESCRIPTOR . . . . .	545
6.96.2.77	USB_REQUEST_GET_INTERFACE . . . . .	546
6.96.2.78	USB_REQUEST_GET_STATUS . . . . .	546
6.96.2.79	USB_REQUEST_SET_ADDRESS . . . . .	546
6.96.2.80	USB_REQUEST_SET_CONFIGURATION . . . . .	546
6.96.2.81	USB_REQUEST_SET_DESCRIPTOR . . . . .	546
6.96.2.82	USB_REQUEST_SET_FEATURE . . . . .	546
6.96.2.83	USB_REQUEST_SET_INTERFACE . . . . .	546
6.96.2.84	USB_REQUEST_SYNC_FRAME . . . . .	546
6.96.2.85	USB_STRING_DESCRIPTOR_TYPE . . . . .	546
6.96.2.86	WBVAL . . . . .	547
6.96.3	Typedef Documentation . . . . .	547
6.96.3.1	USB_HANDLE_T . . . . .	547
6.97	emWin download and installation . . . . .	548
<b>7</b>	<b>Data Structure Documentation</b>	<b>549</b>
7.1	APSR_Type Union Reference . . . . .	549
7.1.1	Detailed Description . . . . .	549
7.1.2	Field Documentation . . . . .	549
7.1.2.1	_reserved0 . . . . .	549
7.1.2.2	_reserved1 . . . . .	550
7.1.2.3	b . . . . .	550
7.1.2.4	b . . . . .	550
7.1.2.5	C . . . . .	550
7.1.2.6	GE . . . . .	550
7.1.2.7	N . . . . .	550
7.1.2.8	Q . . . . .	550
7.1.2.9	V . . . . .	550
7.1.2.10	w . . . . .	550
7.1.2.11	Z . . . . .	551
7.2	BM_T Struct Reference . . . . .	551
7.2.1	Detailed Description . . . . .	551
7.2.2	Field Documentation . . . . .	551
7.2.2.1	Dir . . . . .	551
7.2.2.2	Recipient . . . . .	551
7.2.2.3	Type . . . . .	551
7.3	CONTROL_Type Union Reference . . . . .	551
7.3.1	Detailed Description . . . . .	552
7.3.2	Field Documentation . . . . .	552
7.3.2.1	_reserved0 . . . . .	552

7.3.2.2	b	552
7.3.2.3	b	552
7.3.2.4	FPCA	552
7.3.2.5	nPRIV	552
7.3.2.6	SPSEL	553
7.3.2.7	w	553
7.4	CoreDebug_Type Struct Reference	553
7.4.1	Detailed Description	553
7.4.2	Field Documentation	553
7.4.2.1	DCRDR	553
7.4.2.2	DCRSR	553
7.4.2.3	DEMCR	553
7.4.2.4	DHCSR	554
7.5	DMA_CHDESC_T Struct Reference	554
7.5.1	Detailed Description	554
7.5.2	Field Documentation	554
7.5.2.1	dest	554
7.5.2.2	next	554
7.5.2.3	source	554
7.5.2.4	xfercfg	554
7.6	DMA_DUAL_DESCRIPTOR_T Struct Reference	555
7.6.1	Detailed Description	555
7.7	DMA_PERIPHERAL_CONTEXT_T Struct Reference	555
7.7.1	Detailed Description	555
7.8	DMIC_CHANNEL_CONFIG_T Struct Reference	555
7.8.1	Detailed Description	555
7.9	DMIC_STATISTICS_T Struct Reference	556
7.9.1	Detailed Description	556
7.10	DWT_Type Struct Reference	556
7.10.1	Detailed Description	556
7.10.2	Field Documentation	557
7.10.2.1	COMP0	557
7.10.2.2	COMP1	557
7.10.2.3	COMP2	557
7.10.2.4	COMP3	557
7.10.2.5	CPICNT	557
7.10.2.6	CTRL	557
7.10.2.7	CYCCNT	557
7.10.2.8	EXCCNT	558
7.10.2.9	FOLDCNT	558

7.10.2.10 FUNCTION0 . . . . .	558
7.10.2.11 FUNCTION1 . . . . .	558
7.10.2.12 FUNCTION2 . . . . .	558
7.10.2.13 FUNCTION3 . . . . .	558
7.10.2.14 LSUCNT . . . . .	558
7.10.2.15 MASK0 . . . . .	558
7.10.2.16 MASK1 . . . . .	558
7.10.2.17 MASK2 . . . . .	559
7.10.2.18 MASK3 . . . . .	559
7.10.2.19 PCSR . . . . .	559
7.10.2.20 RESERVED0 . . . . .	559
7.10.2.21 RESERVED1 . . . . .	559
7.10.2.22 RESERVED2 . . . . .	559
7.10.2.23 SLEEPcnt . . . . .	559
7.11 I2CM_XFER_T Struct Reference . . . . .	559
7.11.1 Detailed Description . . . . .	559
7.11.2 Field Documentation . . . . .	560
7.11.2.1 rxBuff . . . . .	560
7.11.2.2 rxSz . . . . .	560
7.11.2.3 slaveAddr . . . . .	560
7.11.2.4 status . . . . .	560
7.11.2.5 txBuff . . . . .	560
7.11.2.6 txSz . . . . .	560
7.12 I2CS_XFER_T Struct Reference . . . . .	560
7.12.1 Detailed Description . . . . .	560
7.12.2 Field Documentation . . . . .	561
7.12.2.1 slaveDone . . . . .	561
7.12.2.2 slaveRecv . . . . .	561
7.12.2.3 slaveSend . . . . .	561
7.12.2.4 slaveStart . . . . .	561
7.13 I2S_AUDIO_FORMAT_T Struct Reference . . . . .	562
7.13.1 Detailed Description . . . . .	562
7.13.2 Field Documentation . . . . .	562
7.13.2.1 ChannelNumber . . . . .	562
7.13.2.2 DataPos . . . . .	562
7.13.2.3 Direction . . . . .	562
7.13.2.4 Divider . . . . .	562
7.13.2.5 FIFOdepth . . . . .	562
7.13.2.6 FrameWidth . . . . .	563
7.13.2.7 LeftJust . . . . .	563

7.13.2.8	Mode	563
7.13.2.9	MSCfg	563
7.13.2.10	PDMDData	563
7.13.2.11	RightLow	563
7.13.2.12	SCKPol	563
7.13.2.13	WordWidth	563
7.13.2.14	WSPol	563
7.14	I2S_STATISTICS_T Struct Reference	564
7.14.1	Detailed Description	564
7.14.2	Field Documentation	564
7.14.2.1	fifo_err_rx	564
7.14.2.2	fifo_err_tx	564
7.14.2.3	i2s_busy	564
7.14.2.4	i2s_data_paused	564
7.14.2.5	i2s_slvfrmerr	564
7.14.2.6	interrupts	565
7.14.2.7	lvl_rx	565
7.14.2.8	lvl_tx	565
7.15	IPSR_Type Union Reference	565
7.15.1	Detailed Description	565
7.15.2	Field Documentation	565
7.15.2.1	_reserved0	565
7.15.2.2	b	566
7.15.2.3	b	566
7.15.2.4	ISR	566
7.15.2.5	w	566
7.16	ITM_Type Struct Reference	566
7.16.1	Detailed Description	566
7.16.2	Field Documentation	567
7.16.2.1	CID0	567
7.16.2.2	CID1	567
7.16.2.3	CID2	567
7.16.2.4	CID3	567
7.16.2.5	IMCR	567
7.16.2.6	IRR	567
7.16.2.7	IWR	568
7.16.2.8	LAR	568
7.16.2.9	LSR	568
7.16.2.10	PID0	568
7.16.2.11	PID1	568

7.16.2.12 PID2	568
7.16.2.13 PID3	568
7.16.2.14 PID4	568
7.16.2.15 PID5	568
7.16.2.16 PID6	569
7.16.2.17 PID7	569
7.16.2.18 PORT	569
7.16.2.19 RESERVED0	569
7.16.2.20 RESERVED1	569
7.16.2.21 RESERVED2	569
7.16.2.22 RESERVED3	569
7.16.2.23 RESERVED4	569
7.16.2.24 RESERVED5	569
7.16.2.25 TCR	569
7.16.2.26 TER	569
7.16.2.27 TPR	570
7.16.2.28 u16	570
7.16.2.29 u32	570
7.16.2.30 u8	570
7.17 LPC_ADC_T Struct Reference	570
7.17.1 Detailed Description	570
7.17.2 Field Documentation	571
7.17.2.1 "@1	571
7.17.2.2 "@3	571
7.17.2.3 CALIBR	571
7.17.2.4 CHAN_THRSEL	571
7.17.2.5 CTRL	571
7.17.2.6 DAT	571
7.17.2.7 FLAGS	571
7.17.2.8 INSEL	571
7.17.2.9 INTEN	571
7.17.2.10 RESERVED0	571
7.17.2.11 SEQ_CTRL	572
7.17.2.12 SEQ_GDAT	572
7.17.2.13 SEQA_CTRL	572
7.17.2.14 SEQA_GDAT	572
7.17.2.15 SEQB_CTRL	572
7.17.2.16 SEQB_GDAT	572
7.17.2.17 STARTUP	572
7.17.2.18 THR_HIGH	572



7.17.2.19 THR_LOW	572
7.18 LPC_ASYNC_SYSCON_T Struct Reference	572
7.18.1 Detailed Description	572
7.18.2 Field Documentation	573
7.18.2.1 ASYNCAPBCLKCTRL	573
7.18.2.2 ASYNCAPBCLKCTRLCLR	573
7.18.2.3 ASYNCAPBCLKCTRLSET	573
7.18.2.4 ASYNCAPBCLKSELA	573
7.18.2.5 ASYNCPRESETCTRLCLR	573
7.18.2.6 ASYNCPRESETCTRLSET	573
7.18.2.7 AYSNCPRESETCTRL	573
7.18.2.8 RESERVED0	574
7.18.2.9 RESERVED1	574
7.19 LPC_CRC_T Struct Reference	574
7.19.1 Detailed Description	574
7.19.2 Field Documentation	574
7.19.2.1 "@18	574
7.19.2.2 MODE	574
7.19.2.3 SEED	574
7.19.2.4 SUM	574
7.19.2.5 WRDATA16	575
7.19.2.6 WRDATA32	575
7.19.2.7 WRDATA8	575
7.20 LPC_DMA_CHANNEL_T Struct Reference	575
7.20.1 Detailed Description	575
7.20.2 Field Documentation	575
7.20.2.1 CFG	575
7.20.2.2 CTLSTAT	575
7.20.2.3 RESERVED	575
7.20.2.4 XFRCFG	576
7.21 LPC_DMA_COMMON_T Struct Reference	576
7.21.1 Detailed Description	576
7.21.2 Field Documentation	576
7.21.2.1 ABORT	576
7.21.2.2 ACTIVE	577
7.21.2.3 BUSY	577
7.21.2.4 ENABLECLR	577
7.21.2.5 ENABLESET	577
7.21.2.6 ERRINT	577
7.21.2.7 INTA	577

7.21.2.8	INTB	577
7.21.2.9	INTENCLR	577
7.21.2.10	INTENSET	577
7.21.2.11	RESERVED0	578
7.21.2.12	RESERVED1	578
7.21.2.13	RESERVED10	578
7.21.2.14	RESERVED2	578
7.21.2.15	RESERVED3	578
7.21.2.16	RESERVED4	578
7.21.2.17	RESERVED5	578
7.21.2.18	RESERVED6	578
7.21.2.19	RESERVED7	578
7.21.2.20	RESERVED8	578
7.21.2.21	RESERVED9	578
7.21.2.22	SETTRIG	578
7.21.2.23	SETVALID	579
7.22	LPC_DMA_T Struct Reference	579
7.22.1	Detailed Description	579
7.22.2	Field Documentation	579
7.22.2.1	CTRL	579
7.22.2.2	DMACH	579
7.22.2.3	DMACOMMON	579
7.22.2.4	INTSTAT	579
7.22.2.5	RESERVED0	580
7.22.2.6	RESERVED2	580
7.22.2.7	SRMBASE	580
7.23	LPC_DMIC_Channel_Type Struct Reference	580
7.23.1	Detailed Description	580
7.24	LPC_DMIC_T Struct Reference	580
7.24.1	Detailed Description	580
7.25	LPC_GPIO_T Struct Reference	581
7.25.1	Detailed Description	581
7.25.2	Field Documentation	581
7.25.2.1	B	581
7.25.2.2	CLR	582
7.25.2.3	DIR	582
7.25.2.4	MASK	582
7.25.2.5	MPIN	582
7.25.2.6	NOT	582
7.25.2.7	PIN	582

7.25.2.8	SET	582
7.25.2.9	W	582
7.26	LPC_GPIOGROUPINT_T Struct Reference	582
7.26.1	Detailed Description	583
7.26.2	Field Documentation	583
7.26.2.1	CTRL	583
7.26.2.2	PORT_ENA	583
7.26.2.3	PORT_POL	583
7.26.2.4	RESERVED0	583
7.26.2.5	RESERVED1	583
7.27	LPC_I2S_T Struct Reference	583
7.27.1	Detailed Description	583
7.27.2	Field Documentation	584
7.27.2.1	CFG1	584
7.27.2.2	CFG2	584
7.27.2.3	DIV	584
7.27.2.4	FIFOCFG	584
7.27.2.5	FIFOINTENCLR	584
7.27.2.6	FIFOINTENSET	585
7.27.2.7	FIFOINTSTAT	585
7.27.2.8	FIFORD	585
7.27.2.9	FIFORD48HNOPOP	585
7.27.2.10	FIFORDNOPOP	585
7.27.2.11	FIFOSTAT	585
7.27.2.12	FIFOTRIG	585
7.27.2.13	FIFOWR	585
7.27.2.14	PID	585
7.27.2.15	PSELID	586
7.27.2.16	RESERVED0	586
7.27.2.17	RESERVED00	586
7.27.2.18	RESERVED0A	586
7.27.2.19	RESERVED5	586
7.27.2.20	STAT	586
7.28	LPC_INMUX_T Struct Reference	586
7.28.1	Detailed Description	586
7.28.2	Field Documentation	587
7.28.2.1	DMA_ITRIG_INMUX	587
7.28.2.2	DMA_OTRIG_INMUX	587
7.28.2.3	FREQMEAS_REF	587
7.28.2.4	FREQMEAS_TARGET	587

7.28.2.5	PINTSEL	587
7.28.2.6	RESERVED1	587
7.28.2.7	RESERVED2	587
7.28.2.8	RESERVED3	587
7.29	LPC_IOCON_T Struct Reference	588
7.29.1	Detailed Description	588
7.29.2	Field Documentation	588
7.29.2.1	PIO	588
7.30	LPC_MBOX_T Struct Reference	588
7.30.1	Detailed Description	588
7.30.2	Field Documentation	588
7.30.2.1	BOX	588
7.30.2.2	MUTEX	588
7.30.2.3	RESERVED1	589
7.30.2.4	RESERVED2	589
7.31	LPC_MBOXIRQ_T Struct Reference	589
7.31.1	Detailed Description	589
7.31.2	Field Documentation	589
7.31.2.1	IRQ	589
7.31.2.2	IRQCLR	589
7.31.2.3	IRQSET	589
7.31.2.4	RESERVED	589
7.32	LPC_MRT_CH_T Struct Reference	590
7.32.1	Detailed Description	590
7.32.2	Field Documentation	590
7.32.2.1	CTRL	590
7.32.2.2	INTVAL	590
7.32.2.3	STAT	590
7.32.2.4	TIMER	590
7.33	LPC_MRT_T Struct Reference	590
7.33.1	Detailed Description	590
7.33.2	Field Documentation	591
7.33.2.1	CHANNEL	591
7.33.2.2	IDLE_CH	591
7.33.2.3	IRQ_FLAG	591
7.33.2.4	MODCFG	591
7.33.2.5	unused	591
7.34	LPC_PIN_INT_T Struct Reference	591
7.34.1	Detailed Description	591
7.34.2	Field Documentation	592

7.34.2.1	CIENTF	592
7.34.2.2	CIENTR	592
7.34.2.3	FALL	592
7.34.2.4	IENF	592
7.34.2.5	IENR	592
7.34.2.6	ISEL	592
7.34.2.7	IST	592
7.34.2.8	PMCFG	593
7.34.2.9	PMCTRL	593
7.34.2.10	PMSRC	593
7.34.2.11	RISE	593
7.34.2.12	SIENF	593
7.34.2.13	SIENR	593
7.35	LPC_PMU_T Struct Reference	593
7.35.1	Detailed Description	593
7.35.2	Field Documentation	594
7.35.2.1	BODCTRL	594
7.35.2.2	RESERVED0	594
7.36	LPC_ROM_API_T Struct Reference	594
7.36.1	Detailed Description	594
7.36.2	Field Documentation	594
7.36.2.1	reserved_adcaltd	594
7.36.2.2	reserved_clib	595
7.36.2.3	reserved_div	595
7.36.2.4	reserved_dmaaltd	595
7.36.2.5	reserved_flexcomm	595
7.36.2.6	reserved_i2cm	595
7.36.2.7	reserved_i2cmon	595
7.36.2.8	reserved_i2cs	595
7.36.2.9	reserved_power	595
7.36.2.10	reserved_spim	595
7.36.2.11	reserved_spis	596
7.36.2.12	reserved_uartalt	596
7.36.2.13	reserved_usart	596
7.36.2.14	usbdApiBase	596
7.37	LPC_RTC_T Struct Reference	596
7.37.1	Detailed Description	596
7.37.2	Field Documentation	596
7.37.2.1	COUNT	596
7.37.2.2	CTRL	597

7.37.2.3 MATCH	597
7.37.2.4 WAKE	597
7.38 LPC_SCT_T Struct Reference	597
7.38.1 Detailed Description	597
7.38.2 Field Documentation	599
7.38.2.1 "@20	599
7.38.2.2 "@22	599
7.38.2.3 "@24	599
7.38.2.4 "@26	599
7.38.2.5 CAP	599
7.38.2.6 CAPCTRL	599
7.38.2.7 CLR	599
7.38.2.8 CONEN	599
7.38.2.9 CONFIG	599
7.38.2.10 CONFLAG	599
7.38.2.11 COUNT_H	599
7.38.2.12 COUNT_L	600
7.38.2.13 COUNT_U	600
7.38.2.14 CTRL	600
7.38.2.15 CTRL_H	600
7.38.2.16 CTRL_L	600
7.38.2.17 CTRL_U	600
7.38.2.18 DMA0REQUEST	600
7.38.2.19 DMA1REQUEST	600
7.38.2.20 EVEN	600
7.38.2.21 EVENT	601
7.38.2.22 EVFLAG	601
7.38.2.23 H	601
7.38.2.24 HALT_H	601
7.38.2.25 HALT_L	601
7.38.2.26 INPUT	601
7.38.2.27 L	601
7.38.2.28 LIMIT_H	601
7.38.2.29 LIMIT_L	601
7.38.2.30 MATCH	601
7.38.2.31 MATCHREL	602
7.38.2.32 MODULECONTENT	602
7.38.2.33 OUT	602
7.38.2.34 OUTPUT	602
7.38.2.35 OUTPUTDIRCTRL	602

7.38.2.36 REGMODE_H . . . . .	602
7.38.2.37 REGMODE_L . . . . .	602
7.38.2.38 RES . . . . .	602
7.38.2.39 RESERVED1 . . . . .	602
7.38.2.40 RESERVED10 . . . . .	602
7.38.2.41 RESERVED2 . . . . .	602
7.38.2.42 RESERVED3 . . . . .	603
7.38.2.43 RESERVED6 . . . . .	603
7.38.2.44 RESERVED9 . . . . .	603
7.38.2.45 SET . . . . .	603
7.38.2.46 START_H . . . . .	603
7.38.2.47 START_L . . . . .	603
7.38.2.48 STATE . . . . .	603
7.38.2.49 STATE_H . . . . .	603
7.38.2.50 STATE_L . . . . .	603
7.38.2.51 STOP_H . . . . .	603
7.38.2.52 STOP_L . . . . .	604
7.38.2.53 U . . . . .	604
7.39 LPC_SPI_T Struct Reference . . . . .	604
7.39.1 Detailed Description . . . . .	604
7.39.2 Field Documentation . . . . .	604
7.39.2.1 CFG . . . . .	604
7.39.2.2 DIV . . . . .	605
7.39.2.3 DLY . . . . .	605
7.39.2.4 FIFOCFG . . . . .	605
7.39.2.5 FIFOINTENCLR . . . . .	605
7.39.2.6 FIFOINTENSET . . . . .	605
7.39.2.7 FIFOINTSTAT . . . . .	605
7.39.2.8 FIFORD . . . . .	605
7.39.2.9 FIFORDNOPOP . . . . .	605
7.39.2.10 FIFOSTAT . . . . .	605
7.39.2.11 FIFOTRIG . . . . .	606
7.39.2.12 FIFOWR . . . . .	606
7.39.2.13 INTENCLR . . . . .	606
7.39.2.14 INTENSET . . . . .	606
7.39.2.15 INTSTAT . . . . .	606
7.39.2.16 PID . . . . .	606
7.39.2.17 PSELID . . . . .	606
7.39.2.18 RESERVED0 . . . . .	606
7.39.2.19 RESERVED1 . . . . .	606

7.39.2.20 STAT . . . . .	607
7.40 LPC_SYSCON_T Struct Reference . . . . .	607
7.40.1 Detailed Description . . . . .	607
7.40.2 Field Documentation . . . . .	609
7.40.2.1 ADCCLKDIV . . . . .	609
7.40.2.2 ADCCLKSEL . . . . .	609
7.40.2.3 AHBCLKCTRL . . . . .	609
7.40.2.4 AHBCLKCTRLCLR . . . . .	609
7.40.2.5 AHBCLKCTRLSET . . . . .	609
7.40.2.6 AHBCLKDIV . . . . .	609
7.40.2.7 AHBMATPRIO . . . . .	609
7.40.2.8 ASYNCAPBCTRL . . . . .	610
7.40.2.9 AUTOCGOR . . . . .	610
7.40.2.10 CLKOUTDIV . . . . .	610
7.40.2.11 CLKOUTSELA . . . . .	610
7.40.2.12 CPBOOT . . . . .	610
7.40.2.13 CPCTRL . . . . .	610
7.40.2.14 CPSTACK . . . . .	610
7.40.2.15 CPSTAT . . . . .	610
7.40.2.16 DEVICE_ID . . . . .	610
7.40.2.17 DMICCLKDIV . . . . .	611
7.40.2.18 DMICCLKSEL . . . . .	611
7.40.2.19 FLASHCFG . . . . .	611
7.40.2.20 FREQMCTRL . . . . .	611
7.40.2.21 FRGCLKSEL . . . . .	611
7.40.2.22 FRGCTRL . . . . .	611
7.40.2.23 FROCTRL . . . . .	611
7.40.2.24 FXCOMCLKSEL . . . . .	611
7.40.2.25 JTAGIDCODE . . . . .	611
7.40.2.26 MAINCLKSELA . . . . .	612
7.40.2.27 MAINCLKSELB . . . . .	612
7.40.2.28 MCLKCLKSEL . . . . .	612
7.40.2.29 MCLKDIV . . . . .	612
7.40.2.30 MCLKIO . . . . .	612
7.40.2.31 NMISRC . . . . .	612
7.40.2.32 PDRUNCFG . . . . .	612
7.40.2.33 PDRUNCFGCLR . . . . .	612
7.40.2.34 PDRUNCFGSET . . . . .	612
7.40.2.35 PIOPORCAP . . . . .	613
7.40.2.36 PIORESCAP . . . . .	613



7.40.2.37 PRESETCTRL . . . . .	613
7.40.2.38 PRESETCTRLCLR . . . . .	613
7.40.2.39 PRESETCTRLSET . . . . .	613
7.40.2.40 RESERVED0 . . . . .	613
7.40.2.41 RESERVED1 . . . . .	613
7.40.2.42 RESERVED10 . . . . .	613
7.40.2.43 RESERVED11 . . . . .	613
7.40.2.44 RESERVED12 . . . . .	613
7.40.2.45 RESERVED13 . . . . .	613
7.40.2.46 RESERVED14 . . . . .	614
7.40.2.47 RESERVED15 . . . . .	614
7.40.2.48 RESERVED16 . . . . .	614
7.40.2.49 RESERVED16A . . . . .	614
7.40.2.50 RESERVED17 . . . . .	614
7.40.2.51 RESERVED18 . . . . .	614
7.40.2.52 RESERVED19 . . . . .	614
7.40.2.53 RESERVED2 . . . . .	614
7.40.2.54 RESERVED20 . . . . .	614
7.40.2.55 RESERVED21 . . . . .	614
7.40.2.56 RESERVED22 . . . . .	614
7.40.2.57 RESERVED23 . . . . .	614
7.40.2.58 RESERVED24 . . . . .	615
7.40.2.59 RESERVED25 . . . . .	615
7.40.2.60 RESERVED26 . . . . .	615
7.40.2.61 RESERVED27 . . . . .	615
7.40.2.62 RESERVED28 . . . . .	615
7.40.2.63 RESERVED29 . . . . .	615
7.40.2.64 RESERVED3 . . . . .	615
7.40.2.65 RESERVED30 . . . . .	615
7.40.2.66 RESERVED31 . . . . .	615
7.40.2.67 RESERVED32 . . . . .	615
7.40.2.68 RESERVED33 . . . . .	615
7.40.2.69 RESERVED34 . . . . .	615
7.40.2.70 RESERVED35 . . . . .	616
7.40.2.71 RESERVED36 . . . . .	616
7.40.2.72 RESERVED37 . . . . .	616
7.40.2.73 RESERVED4 . . . . .	616
7.40.2.74 RESERVED5 . . . . .	616
7.40.2.75 RESERVED6 . . . . .	616
7.40.2.76 RESERVED7 . . . . .	616

7.40.2.77 RESERVED8 . . . . .	616
7.40.2.78 RESERVED9 . . . . .	616
7.40.2.79 RTCOSCCTRL . . . . .	616
7.40.2.80 SPIFICKDIV . . . . .	616
7.40.2.81 SPIFICKSEL . . . . .	617
7.40.2.82 STARTERP . . . . .	617
7.40.2.83 STARTERPCLR . . . . .	617
7.40.2.84 STARTERPSET . . . . .	617
7.40.2.85 SYSMEMREMAP . . . . .	617
7.40.2.86 SYSPLLCLKSEL . . . . .	617
7.40.2.87 SYSPLLCTRL . . . . .	617
7.40.2.88 SYSPLLNDEC . . . . .	617
7.40.2.89 SYSPLLPDEC . . . . .	617
7.40.2.90 SYSPLLSSCTRL . . . . .	618
7.40.2.91 SYSPLLSTAT . . . . .	618
7.40.2.92 SYSRSTSTAT . . . . .	618
7.40.2.93 SYSTCKCAL . . . . .	618
7.40.2.94 SYSTICKCLKDIV . . . . .	618
7.40.2.95 USBCLKCTRL . . . . .	618
7.40.2.96 USBCLKDIV . . . . .	618
7.40.2.97 USBCLKSEL . . . . .	618
7.40.2.98 USBCLKSTAT . . . . .	618
7.40.2.99 WDTOSCCTRL . . . . .	619
7.41 LPC_TIMER_T Struct Reference . . . . .	619
7.41.1 Detailed Description . . . . .	619
7.41.2 Field Documentation . . . . .	619
7.41.2.1 CCR . . . . .	619
7.41.2.2 CR . . . . .	619
7.41.2.3 CTCR . . . . .	619
7.41.2.4 EMR . . . . .	620
7.41.2.5 IR . . . . .	620
7.41.2.6 MCR . . . . .	620
7.41.2.7 MR . . . . .	620
7.41.2.8 PC . . . . .	620
7.41.2.9 PR . . . . .	620
7.41.2.10 PWMCM . . . . .	620
7.41.2.11 RESERVED0 . . . . .	620
7.41.2.12 TC . . . . .	620
7.41.2.13 TCR . . . . .	621
7.42 LPC_USART_T Struct Reference . . . . .	621

7.42.1 Detailed Description	621
7.42.2 Field Documentation	621
7.42.2.1 ADDR	621
7.42.2.2 BRG	621
7.42.2.3 CFG	622
7.42.2.4 CTL	622
7.42.2.5 FIFOCFG	622
7.42.2.6 FIFOINTENCLR	622
7.42.2.7 FIFOINTENSET	622
7.42.2.8 FIFOINTSTAT	622
7.42.2.9 FIFORD	622
7.42.2.10 FIFORDNOPOP	622
7.42.2.11 FIFOSTAT	622
7.42.2.12 FIFOTRIG	623
7.42.2.13 FIFOWR	623
7.42.2.14 INTENCLR	623
7.42.2.15 INTENSET	623
7.42.2.16 INTSTAT	623
7.42.2.17 OSR	623
7.42.2.18 PID	623
7.42.2.19 PSELID	623
7.42.2.20 STAT	623
7.43 LPC_UTICK_T Struct Reference	624
7.43.1 Detailed Description	624
7.43.2 Field Documentation	624
7.43.2.1 CTRL	624
7.43.2.2 STATUS	624
7.44 LPC_WWDT_T Struct Reference	624
7.44.1 Detailed Description	624
7.44.2 Field Documentation	625
7.44.2.1 FEED	625
7.44.2.2 MOD	625
7.44.2.3 RESERVED0	625
7.44.2.4 TC	625
7.44.2.5 TV	625
7.44.2.6 WARNINT	625
7.44.2.7 WINDOW	625
7.45 NVIC_Type Struct Reference	625
7.45.1 Detailed Description	625
7.45.2 Field Documentation	626

7.45.2.1	IABR	626
7.45.2.2	ICER	626
7.45.2.3	ICPR	626
7.45.2.4	IP	626
7.45.2.5	IP	626
7.45.2.6	ISER	626
7.45.2.7	ISPR	627
7.45.2.8	RESERVED0	627
7.45.2.9	RESERVED2	627
7.45.2.10	RESERVED3	627
7.45.2.11	RESERVED4	627
7.45.2.12	RESERVED5	627
7.45.2.13	RSERVED1	627
7.45.2.14	STIR	627
7.46	PINMUX_GRP_T Struct Reference	627
7.46.1	Detailed Description	627
7.46.2	Field Documentation	628
7.46.2.1	modefunc	628
7.46.2.2	pin	628
7.46.2.3	port	628
7.47	PINTABLE_T Struct Reference	628
7.47.1	Detailed Description	628
7.47.2	Field Documentation	628
7.47.2.1	crc32_len	628
7.47.2.2	crc32_val	629
7.47.2.3	hostIrqPortPin	629
7.47.2.4	hostMisoPortPin	629
7.47.2.5	hostMosiPortPin	629
7.47.2.6	hostSckPortPin	629
7.47.2.7	hostSselPortPin	629
7.47.2.8	ifSel	629
7.47.2.9	img_type	629
7.47.2.10	marker	629
7.47.2.11	version	629
7.47.2.12	xorVal	629
7.48	PLL_CONFIG_T Struct Reference	630
7.48.1	Detailed Description	630
7.48.2	Field Documentation	630
7.48.2.1	desiredRate	630
7.48.2.2	flags	630

7.48.2.3	InputRate	630
7.48.2.4	mfDither	630
7.48.2.5	ss_mc	630
7.48.2.6	ss_mf	631
7.48.2.7	ss_mr	631
7.49	PLL_SETUP_T Struct Reference	631
7.49.1	Detailed Description	631
7.49.2	Field Documentation	631
7.49.2.1	flags	631
7.49.2.2	pllRate	631
7.49.2.3	SYSPLLCTRL	632
7.49.2.4	SYSPLLNDCEC	632
7.49.2.5	SYSPLLPDEC	632
7.49.2.6	SYSPLLSSCTRL	632
7.50	REQUEST_TYPE Union Reference	632
7.50.1	Detailed Description	632
7.50.2	Field Documentation	632
7.50.2.1	B	632
7.50.2.2	BM	632
7.51	RINGBUFF_T Struct Reference	633
7.51.1	Detailed Description	633
7.51.2	Field Documentation	633
7.51.2.1	copy	633
7.51.2.2	count	633
7.51.2.3	data	633
7.51.2.4	head	633
7.51.2.5	itemSz	633
7.51.2.6	tail	633
7.52	SCB_Type Struct Reference	634
7.52.1	Detailed Description	634
7.52.2	Field Documentation	634
7.52.2.1	ADR	634
7.52.2.2	AFSR	634
7.52.2.3	AIRCR	634
7.52.2.4	BFAR	635
7.52.2.5	CCR	635
7.52.2.6	CFSR	635
7.52.2.7	CPACR	635
7.52.2.8	CPUID	635
7.52.2.9	DFR	635

7.52.2.10	DFSR	635
7.52.2.11	HFSR	635
7.52.2.12	ICSR	635
7.52.2.13	ISAR	636
7.52.2.14	MMFAR	636
7.52.2.15	MMFR	636
7.52.2.16	PFR	636
7.52.2.17	RESERVED0	636
7.52.2.18	RESERVED1	636
7.52.2.19	SCR	636
7.52.2.20	SHCSR	636
7.52.2.21	SHP	636
7.52.2.22	SHP	636
7.52.2.23	VTOR	637
7.53	SCnSCB_Type Struct Reference	637
7.53.1	Detailed Description	637
7.53.2	Field Documentation	637
7.53.2.1	ACTLR	637
7.53.2.2	ICTR	637
7.53.2.3	RESERVED0	637
7.54	SPI_CFGSETUP_T Struct Reference	637
7.54.1	Detailed Description	637
7.54.2	Field Documentation	638
7.54.2.1	lsbFirst	638
7.54.2.2	master	638
7.54.2.3	mode	638
7.54.2.4	reserved	638
7.55	SPIM_DELAY_CONFIG_T Struct Reference	638
7.55.1	Detailed Description	638
7.55.2	Field Documentation	639
7.55.2.1	FrameDelay	639
7.55.2.2	PostDelay	639
7.55.2.3	PreDelay	639
7.55.2.4	TransferDelay	639
7.56	SPIM_XFER_T Struct Reference	639
7.56.1	Detailed Description	639
7.56.2	Field Documentation	640
7.56.2.1	dataWidth	640
7.56.2.2	eventCB	640
7.56.2.3	option	640

7.56.2.4	rxCount	640
7.56.2.5	rxData	640
7.56.2.6	rxIndex	640
7.56.2.7	sselNum	640
7.56.2.8	state	640
7.56.2.9	txCount	641
7.56.2.10	txData	641
7.56.2.11	txIndex	641
7.56.2.12	usrData	641
7.57	SPIS_XFER_T Struct Reference	641
7.57.1	Detailed Description	641
7.57.2	Field Documentation	642
7.57.2.1	dataWidth	642
7.57.2.2	eventCB	642
7.57.2.3	rxCount	642
7.57.2.4	rxData	642
7.57.2.5	rxIndex	642
7.57.2.6	ss_count	642
7.57.2.7	ss_state	642
7.57.2.8	sselNum	642
7.57.2.9	thresCount	642
7.57.2.10	txCount	643
7.57.2.11	txData	643
7.57.2.12	txIndex	643
7.58	SysTick_Type Struct Reference	643
7.58.1	Detailed Description	643
7.58.2	Field Documentation	643
7.58.2.1	CALIB	643
7.58.2.2	CTRL	643
7.58.2.3	LOAD	644
7.58.2.4	VAL	644
7.59	TPI_Type Struct Reference	644
7.59.1	Detailed Description	644
7.59.2	Field Documentation	645
7.59.2.1	ACPR	645
7.59.2.2	CLAIMCLR	645
7.59.2.3	CLAIMSET	645
7.59.2.4	CSPSR	645
7.59.2.5	DEVID	645
7.59.2.6	DEVTYPE	645

7.59.2.7	FFCR	645
7.59.2.8	FFSR	645
7.59.2.9	FIFO0	645
7.59.2.10	FIFO1	646
7.59.2.11	FSCR	646
7.59.2.12	ITATBCTR0	646
7.59.2.13	ITATBCTR2	646
7.59.2.14	ITCTRL	646
7.59.2.15	RESERVED0	646
7.59.2.16	RESERVED1	646
7.59.2.17	RESERVED2	646
7.59.2.18	RESERVED3	646
7.59.2.19	RESERVED4	646
7.59.2.20	RESERVED5	646
7.59.2.21	RESERVED7	647
7.59.2.22	SPPR	647
7.59.2.23	SSPSR	647
7.59.2.24	TRIGGER	647
7.60	UART_BAUD_T Struct Reference	647
7.60.1	Detailed Description	647
7.60.2	Field Documentation	648
7.60.2.1	baud	648
7.60.2.2	clk	648
7.60.2.3	div	648
7.60.2.4	mul	648
7.60.2.5	ovr	648
7.61	UART_STATISTICS_T Struct Reference	648
7.61.1	Detailed Description	648
7.61.2	Field Documentation	649
7.61.2.1	fifo_err_rx	649
7.61.2.2	fifo_err_tx	649
7.61.2.3	interrupts	649
7.61.2.4	lvl_rx	649
7.61.2.5	lvl_tx	649
7.61.2.6	uart_break	649
7.61.2.7	uart_cts	649
7.61.2.8	uart_err_auto_baud	650
7.61.2.9	uart_err_frame	650
7.61.2.10	uart_err_parity	650
7.61.2.11	uart_err_rx_noise	650



7.61.2.12	uart_start	650
7.62	USB_COMMON_DESCRIPTOR Struct Reference	650
7.62.1	Detailed Description	650
7.62.2	Field Documentation	650
7.62.2.1	bDescriptorType	650
7.62.2.2	bLength	651
7.63	USB_CONFIGURATION_DESCRIPTOR Struct Reference	651
7.63.1	Detailed Description	651
7.63.2	Field Documentation	651
7.63.2.1	bConfigurationValue	651
7.63.2.2	bDescriptorType	651
7.63.2.3	bLength	651
7.63.2.4	bmAttributes	651
7.63.2.5	bMaxPower	652
7.63.2.6	bNumInterfaces	652
7.63.2.7	iConfiguration	652
7.63.2.8	wTotalLength	652
7.64	USB_DEVICE_DESCRIPTOR Struct Reference	652
7.64.1	Detailed Description	652
7.64.2	Field Documentation	653
7.64.2.1	bcdDevice	653
7.64.2.2	bcdUSB	653
7.64.2.3	bDescriptorType	653
7.64.2.4	bDeviceClass	653
7.64.2.5	bDeviceProtocol	653
7.64.2.6	bDeviceSubClass	654
7.64.2.7	bLength	654
7.64.2.8	bMaxPacketSize0	654
7.64.2.9	bNumConfigurations	654
7.64.2.10	idProduct	654
7.64.2.11	idVendor	654
7.64.2.12	iManufacturer	654
7.64.2.13	iProduct	654
7.64.2.14	iSerialNumber	655
7.65	USB_DEVICE_QUALIFIER_DESCRIPTOR Struct Reference	655
7.65.1	Detailed Description	655
7.65.2	Field Documentation	655
7.65.2.1	bcdUSB	655
7.65.2.2	bDescriptorType	655
7.65.2.3	bDeviceClass	655

7.65.2.4	bDeviceProtocol	655
7.65.2.5	bDeviceSubClass	656
7.65.2.6	bLength	656
7.65.2.7	bMaxPacketSize0	656
7.65.2.8	bNumConfigurations	656
7.65.2.9	bReserved	656
7.66	USB_ENDPOINT_DESCRIPTOR Struct Reference	656
7.66.1	Detailed Description	656
7.66.2	Field Documentation	657
7.66.2.1	bDescriptorType	657
7.66.2.2	bEndpointAddress	657
7.66.2.3	bInterval	657
7.66.2.4	bLength	657
7.66.2.5	bmAttributes	657
7.66.2.6	wMaxPacketSize	658
7.67	USB_IAD_DESCRIPTOR Struct Reference	659
7.67.1	Detailed Description	659
7.67.2	Field Documentation	659
7.67.2.1	bDescriptorType	659
7.67.2.2	bFirstInterface	659
7.67.2.3	bFunctionClass	659
7.67.2.4	bFunctionProtocol	659
7.67.2.5	bFunctionSubClass	659
7.67.2.6	bInterfaceCount	660
7.67.2.7	bLength	660
7.67.2.8	iFunction	660
7.68	USB_INTERFACE_DESCRIPTOR Struct Reference	660
7.68.1	Detailed Description	660
7.68.2	Field Documentation	660
7.68.2.1	bAlternateSetting	660
7.68.2.2	bDescriptorType	660
7.68.2.3	bInterfaceClass	661
7.68.2.4	bInterfaceNumber	661
7.68.2.5	bInterfaceProtocol	661
7.68.2.6	bInterfaceSubClass	661
7.68.2.7	bLength	661
7.68.2.8	bNumEndpoints	661
7.68.2.9	iInterface	661
7.69	USB_OTHER_SPEED_CONFIGURATION Struct Reference	662
7.69.1	Detailed Description	662

7.69.2	Field Documentation	662
7.69.2.1	bConfigurationValue	662
7.69.2.2	bDescriptorType	662
7.69.2.3	bLength	662
7.69.2.4	bmAttributes	662
7.69.2.5	bMaxPower	662
7.69.2.6	bNumInterfaces	663
7.69.2.7	IConfiguration	663
7.69.2.8	wTotalLength	663
7.70	USB_SETUP_PACKET Struct Reference	663
7.70.1	Detailed Description	663
7.70.2	Field Documentation	663
7.70.2.1	bmRequestType	663
7.70.2.2	bRequest	664
7.70.2.3	wIndex	664
7.70.2.4	wLength	664
7.70.2.5	wValue	664
7.71	USB_STRING_DESCRIPTOR Struct Reference	664
7.71.1	Detailed Description	664
7.71.2	Field Documentation	664
7.71.2.1	bDescriptorType	664
7.71.2.2	bLength	665
7.71.2.3	bString	665
7.72	WB_T Struct Reference	665
7.72.1	Detailed Description	665
7.72.2	Field Documentation	665
7.72.2.1	H	665
7.72.2.2	L	665
7.73	WORD_BYTE Union Reference	665
7.73.1	Detailed Description	665
7.73.2	Field Documentation	666
7.73.2.1	W	666
7.73.2.2	WB	666
7.74	xPSR_Type Union Reference	666
7.74.1	Detailed Description	666
7.74.2	Field Documentation	667
7.74.2.1	_reserved0	667
7.74.2.2	_reserved1	667
7.74.2.3	b	667
7.74.2.4	b	667

7.74.2.5	C	667
7.74.2.6	GE	667
7.74.2.7	ISR	667
7.74.2.8	IT	667
7.74.2.9	N	668
7.74.2.10	Q	668
7.74.2.11	T	668
7.74.2.12	V	668
7.74.2.13	w	668
7.74.2.14	Z	668
<b>8</b>	<b>File Documentation</b>	<b>669</b>
8.1	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/adc_5411x.h File Reference	669
8.2	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/chip.h File Reference	672
8.3	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/clock_5411x.h File Reference	675
8.4	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/cmsis.h File Reference	679
8.5	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/cmsis_5411x.h File Reference	679
8.6	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/cmsis_5411x_m0.h File Reference	680
8.7	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/core_cm0plus.h File Reference	681
8.7.1	Detailed Description	681
8.7.2	Macro Definition Documentation	684
8.7.2.1	__CM0PLUS_CMSIS_VERSION	684
8.7.2.2	__CM0PLUS_CMSIS_VERSION_MAIN	684
8.7.2.3	__CM0PLUS_CMSIS_VERSION_SUB	684
8.7.2.4	__CORE_CM0PLUS_H_DEPENDANT	684
8.7.2.5	__CORE_CM0PLUS_H_GENERIC	684
8.7.2.6	__CORTEX_M	684
8.7.2.7	__FPU_USED	684
8.7.2.8	__I	684
8.7.2.9	__IO	685
8.7.2.10	__O	685
8.8	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/core_cm4.h File Reference	685
8.8.1	Detailed Description	685
8.8.2	Macro Definition Documentation	694
8.8.2.1	__CM4_CMSIS_VERSION	694
8.8.2.2	__CM4_CMSIS_VERSION_MAIN	694
8.8.2.3	__CM4_CMSIS_VERSION_SUB	694
8.8.2.4	__CORE_CM4_H_DEPENDANT	694
8.8.2.5	__CORE_CM4_H_GENERIC	694
8.8.2.6	__CORTEX_M	695

8.8.2.7	<code>__I</code>	695
8.8.2.8	<code>__IO</code>	695
8.8.2.9	<code>__O</code>	695
8.9	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/core_cm4_simd.h File Reference	695
8.9.1	Detailed Description	695
8.9.2	Macro Definition Documentation	695
8.9.2.1	<code>__CORE_CM4_SIMD_H</code>	695
8.10	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/core_cmFunc.h File Reference	696
8.10.1	Detailed Description	696
8.11	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/core_cmInstr.h File Reference	696
8.11.1	Detailed Description	696
8.12	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/cpuctrl_5411x.h File Reference	696
8.13	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/crc_5411x.h File Reference	697
8.14	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/dma_5411x.h File Reference	698
8.15	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/dma_service_5411x.h File Reference	701
8.16	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/dmic_5411x.h File Reference	702
8.17	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/error.h File Reference	703
8.17.1	Macro Definition Documentation	705
8.17.1.1	<code>COMPILE_TIME_ASSERT</code>	705
8.17.1.2	<code>offsetof</code>	705
8.17.2	Enumeration Type Documentation	705
8.17.2.1	<code>ErrorCode_t</code>	705
8.18	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/flexcomm_5411x.h File Reference	710
8.19	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/fpu_init.h File Reference	711
8.20	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/gpio_5411x.h File Reference	711
8.21	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/gpiogroup_5411x.h File Reference	712
8.22	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/i2c_common_5411x.h File Reference	713
8.23	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/i2cm_5411x.h File Reference	717
8.24	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/i2cs_5411x.h File Reference	718
8.25	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/i2s_5411x.h File Reference	720
8.25.1	Macro Definition Documentation	723
8.25.1.1	<code>DMAREQ_I2S6_RX</code>	723
8.25.1.2	<code>DMAREQ_I2S6_TX</code>	723
8.25.1.3	<code>DMAREQ_I2S7_RX</code>	723
8.25.1.4	<code>DMAREQ_I2S7_TX</code>	723
8.25.1.5	<code>I2S6_IRQHandler</code>	723
8.25.1.6	<code>I2S6_IRQn</code>	723
8.25.1.7	<code>I2S7_IRQHandler</code>	723
8.25.1.8	<code>I2S7_IRQn</code>	723
8.25.1.9	<code>I2S_CFG1_2NDCOUNT</code>	723

8.25.1.10 I2S_CFG1_DATALEN . . . . .	724
8.25.1.11 I2S_CFG1_DATAPAUSE . . . . .	724
8.25.1.12 I2S_CFG1_LEFTJUST . . . . .	724
8.25.1.13 I2S_CFG1_MAINENABLE . . . . .	724
8.25.1.14 I2S_CFG1_MODE . . . . .	724
8.25.1.15 I2S_CFG1_MSTSLVCFG . . . . .	724
8.25.1.16 I2S_CFG1_ONECHANNEL . . . . .	724
8.25.1.17 I2S_CFG1_PDMDATA . . . . .	724
8.25.1.18 I2S_CFG1_RIGHTLOW . . . . .	724
8.25.1.19 I2S_CFG1_SCK_POL . . . . .	725
8.25.1.20 I2S_CFG1_WS_POL . . . . .	725
8.25.1.21 I2S_CFG2_FRAMELEN . . . . .	725
8.25.1.22 I2S_CFG2_POSITION . . . . .	725
8.25.1.23 I2S_FIFO_CFG_DMARX . . . . .	725
8.25.1.24 I2S_FIFO_CFG_DMATX . . . . .	725
8.25.1.25 I2S_FIFO_CFG_EMPTYRX . . . . .	725
8.25.1.26 I2S_FIFO_CFG_EMPTYTX . . . . .	725
8.25.1.27 I2S_FIFO_CFG_ENABLERX . . . . .	725
8.25.1.28 I2S_FIFO_CFG_ENABLETX . . . . .	726
8.25.1.29 I2S_FIFO_CFG_PACK48 . . . . .	726
8.25.1.30 I2S_FIFO_CFG_POPDBG . . . . .	726
8.25.1.31 I2S_FIFO_CFG_SIZE . . . . .	726
8.25.1.32 I2S_FIFO_CFG_TXI2SE0 . . . . .	726
8.25.1.33 I2S_FIFO_CFG_WAKERX . . . . .	726
8.25.1.34 I2S_FIFO_CFG_WAKETX . . . . .	726
8.25.1.35 I2S_FIFO_INT_BITMASK . . . . .	726
8.25.1.36 I2S_FIFO_INT_PERINT . . . . .	726
8.25.1.37 I2S_FIFO_INT_RXERR . . . . .	727
8.25.1.38 I2S_FIFO_INT_RXLVL . . . . .	727
8.25.1.39 I2S_FIFO_INT_TXERR . . . . .	727
8.25.1.40 I2S_FIFO_INT_TXLVL . . . . .	727
8.25.1.41 I2S_FIFO_STAT_PERINT . . . . .	727
8.25.1.42 I2S_FIFO_STAT_RXERR . . . . .	727
8.25.1.43 I2S_FIFO_STAT_RXFULL . . . . .	727
8.25.1.44 I2S_FIFO_STAT_RXLVL . . . . .	727
8.25.1.45 I2S_FIFO_STAT_RXNOTEMPTY . . . . .	727
8.25.1.46 I2S_FIFO_STAT_TXEMPTY . . . . .	728
8.25.1.47 I2S_FIFO_STAT_TXERR . . . . .	728
8.25.1.48 I2S_FIFO_STAT_TXLVL . . . . .	728
8.25.1.49 I2S_FIFO_STAT_TXNOTFULL . . . . .	728

8.25.1.50 I2S_FIFO_TRIG_RXLVL . . . . .	728
8.25.1.51 I2S_FIFO_TRIG_RXLVLENA . . . . .	728
8.25.1.52 I2S_FIFO_TRIG_TXLVL . . . . .	728
8.25.1.53 I2S_FIFO_TRIG_TXLVLENA . . . . .	728
8.25.1.54 I2S_STAT_BUSY . . . . .	728
8.25.1.55 I2S_STAT_LR . . . . .	729
8.25.1.56 I2S_STAT_PAUSED . . . . .	729
8.25.1.57 I2S_STAT_SLVFRMERR . . . . .	729
8.25.1.58 LPC_I2S6 . . . . .	729
8.25.1.59 LPC_I2S6_BASE . . . . .	729
8.25.1.60 LPC_I2S7 . . . . .	729
8.25.1.61 LPC_I2S7_BASE . . . . .	729
8.25.2 Enumeration Type Documentation . . . . .	729
8.25.2.1 I2S_DIR_T . . . . .	729
8.25.2.2 I2S_FIFO_CMD_T . . . . .	729
8.25.2.3 I2S_MODE_T . . . . .	730
8.25.2.4 I2S_MSTSLVCFG_T . . . . .	730
8.25.3 Function Documentation . . . . .	730
8.25.3.1 Chip_I2S_ClearStatus(LPC_I2S_T *pl2S, uint32_t stsMask) . . . . .	730
8.25.3.2 Chip_I2S_ClrFIFOStatus(LPC_I2S_T *pl2S, uint32_t mask) . . . . .	730
8.25.3.3 Chip_I2S_Config(LPC_I2S_T *pl2S, I2S_AUDIO_FORMAT_T *fmt) . . . . .	731
8.25.3.4 Chip_I2S_DeInit(LPC_I2S_T *pl2S) . . . . .	731
8.25.3.5 Chip_I2S_ErrorHandler(LPC_I2S_T *pl2S, I2S_STATISTICS_T *stat) . . . . .	731
8.25.3.6 Chip_I2S_FIFO_ClearStatus(LPC_I2S_T *pl2S, uint32_t mask) . . . . .	732
8.25.3.7 Chip_I2S_FIFO_ClrInterrupt(LPC_I2S_T *pl2S, uint32_t int_val) . . . . .	733
8.25.3.8 Chip_I2S_FIFO_Config(LPC_I2S_T *pl2S, I2S_AUDIO_FORMAT_T *fmt) . . . . .	733
8.25.3.9 Chip_I2S_FIFO_Control(LPC_I2S_T *pl2S, I2S_AUDIO_FORMAT_T *fmt, I2S_FIFO_CMD_T cmd) . . . . .	733
8.25.3.10 Chip_I2S_FIFO_GetPendingInts(LPC_I2S_T *pl2S) . . . . .	733
8.25.3.11 Chip_I2S_FIFO_SetInterrupt(LPC_I2S_T *pl2S, uint32_t int_val) . . . . .	734
8.25.3.12 Chip_I2S_GetFIFORxLevel(LPC_I2S_T *pl2S) . . . . .	734
8.25.3.13 Chip_I2S_GetFIFOStatus(LPC_I2S_T *pl2S) . . . . .	734
8.25.3.14 Chip_I2S_GetFIFOTrigLevel(LPC_I2S_T *pl2S) . . . . .	734
8.25.3.15 Chip_I2S_GetFIFOTxLevel(LPC_I2S_T *pl2S) . . . . .	735
8.25.3.16 Chip_I2S_GetStatus(LPC_I2S_T *pl2S) . . . . .	735
8.25.3.17 Chip_I2S_Init(LPC_I2S_T *pl2S, I2S_AUDIO_FORMAT_T *fmt) . . . . .	735
8.25.3.18 Chip_I2S_Pause(LPC_I2S_T *pl2S) . . . . .	735
8.25.3.19 Chip_I2S_Play(LPC_I2S_T *pl2S) . . . . .	736
8.25.3.20 Chip_I2S_Receive(LPC_I2S_T *pl2S) . . . . .	736
8.25.3.21 Chip_I2S_RX_Init(LPC_I2S_T *pl2S) . . . . .	736

8.25.3.22	Chip_I2S_Send(LPC_I2S_T *pl2S, uint32_t data)	736
8.25.3.23	Chip_I2S_SetFIFOTrigLevel(LPC_I2S_T *pl2S, uint8_t tx_lvl, uint8_t rx_lvl)	737
8.25.3.24	Chip_I2S_Start(LPC_I2S_T *pl2S)	737
8.25.3.25	Chip_I2S_Stop(LPC_I2S_T *pl2S)	737
8.25.3.26	Chip_I2S_TX_Init(LPC_I2S_T *pl2S)	738
8.26	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/iap.h File Reference	738
8.27	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/inmux_5411x.h File Reference	739
8.28	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/iocon_5411x.h File Reference	740
8.29	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/lpc_assert.h File Reference	741
8.29.1	Macro Definition Documentation	741
8.29.1.1	LPC_ASSERT	741
8.30	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/lpc_types.h File Reference	741
8.31	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/mailbox_5411x.h File Reference	742
8.32	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/mrt_5411x.h File Reference	743
8.33	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/packing.h File Reference	745
8.33.1	Macro Definition Documentation	745
8.33.1.1	ALIGNED	745
8.34	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pinint_5411x.h File Reference	745
8.35	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pintable_5411x.h File Reference	747
8.36	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pll_5411x.h File Reference	747
8.37	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pmu_5411x.h File Reference	749
8.38	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/power_lib_5411x.h File Reference	749
8.39	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ring_buffer.h File Reference	750
8.40	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/romapi_5411x.h File Reference	751
8.41	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc rtc_5411x.h File Reference	751
8.42	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/rtc_ut.h File Reference	752
8.43	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/sct_5411x.h File Reference	753
8.44	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/sct_pwm_5411x.h File Reference	756
8.45	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/spi_common_5411x.h File Reference	756
8.46	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/spim_5411x.h File Reference	760
8.47	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/spis_5411x.h File Reference	761
8.48	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/stopwatch.h File Reference	762
8.49	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/syscon_5411x.h File Reference	762
8.50	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/timer_5411x.h File Reference	765
8.51	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/uart_5411x.h File Reference	767
8.52	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/usbd.h File Reference	771
8.52.1	Detailed Description	771
8.53	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/utick_5411x.h File Reference	774
8.54	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/wwdt_5411x.h File Reference	774
8.55	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/adc_5411x.c File Reference	775



8.56	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/chip_5411x.c File Reference	776
8.57	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/clock_5411x.c File Reference	776
8.57.1	Macro Definition Documentation	777
8.57.1.1	WDT_FREQ_LOOKUP	777
8.57.2	Function Documentation	777
8.57.2.1	Chip_Clock_GetAsyncSyscon_ClockRate_NoDiv(void)	777
8.57.2.2	Chip_Clock_GetMCLKClockRate(void)	777
8.57.3	Variable Documentation	777
8.57.3.1	mclk_in_rate	777
8.58	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/dma_5411x.c File Reference	778
8.59	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/dma_service_5411x.c File Reference	778
8.59.1	Macro Definition Documentation	778
8.59.1.1	DMA_XFERCFG_DSTINC	778
8.59.1.2	DMA_XFERCFG_SRCINC	778
8.59.1.3	DMA_XFERCFG_WIDTH	779
8.59.2	Function Documentation	779
8.59.2.1	Chip_DMASERVICE_ErrorHandler(void)	779
8.59.3	Variable Documentation	779
8.59.3.1	dma_pCallback_array	779
8.59.3.2	dma_service_error_cb	779
8.60	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/dmic_5411x.c File Reference	779
8.61	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/flexcomm_5411x.c File Reference	780
8.62	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/fpu_init.c File Reference	780
8.63	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/i2cm_5411x.c File Reference	780
8.64	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/i2cs_5411x.c File Reference	780
8.65	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/i2s_5411x.c File Reference	781
8.65.1	Function Documentation	781
8.65.1.1	Chip_I2S_Config(LPC_I2S_T *pl2S, I2S_AUDIO_FORMAT_T *fmt)	781
8.65.1.2	Chip_I2S_ErrorHandler(LPC_I2S_T *pl2S, I2S_STATISTICS_T *stat)	781
8.65.1.3	Chip_I2S_FIFO_Config(LPC_I2S_T *pl2S, I2S_AUDIO_FORMAT_T *fmt)	782
8.65.1.4	Chip_I2S_FIFO_Control(LPC_I2S_T *pl2S, I2S_AUDIO_FORMAT_T *fmt, I2S_FIFO_CMD_T cmd)	782
8.65.1.5	Chip_I2S_Init(LPC_I2S_T *pl2S, I2S_AUDIO_FORMAT_T *fmt)	782
8.65.1.6	fifo_ctrl_rx(LPC_I2S_T *pl2S, I2S_FIFO_CMD_T cmd)	782
8.65.1.7	fifo_ctrl_tx(LPC_I2S_T *pl2S, I2S_FIFO_CMD_T cmd)	783
8.66	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/iap.c File Reference	783
8.67	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/pll_5411x.c File Reference	783
8.67.1	Macro Definition Documentation	786
8.67.1.1	MVALMAX	786
8.67.1.2	NVALMAX	786

8.67.1.3	PLL0_SSCG_DITHER_VALUE	786
8.67.1.4	PLL0_SSCG_MC_COMP_VALUE	786
8.67.1.5	PLL0_SSCG_MF_FREQ_VALUE	786
8.67.1.6	PLL0_SSCG_MR_DEPTH_VALUE	786
8.67.1.7	PLL_CTRL_BANDSEL_SSCGREG_N	787
8.67.1.8	PLL_CTRL_BANDSEL_SSCGREG_N_P	787
8.67.1.9	PLL_CTRL_BYPASS	787
8.67.1.10	PLL_CTRL_BYPASS_FBDIV2	787
8.67.1.11	PLL_CTRL_BYPASS_FBDIV2_P	787
8.67.1.12	PLL_CTRL_BYPASS_P	787
8.67.1.13	PLL_CTRL_DIRECTI	787
8.67.1.14	PLL_CTRL_DIRECTI_P	787
8.67.1.15	PLL_CTRL_DIRECTO	787
8.67.1.16	PLL_CTRL_DIRECTO_P	787
8.67.1.17	PLL_CTRL_UPLIMOFF	787
8.67.1.18	PLL_CTRL_UPLIMOFF_P	787
8.67.1.19	PLL_LOWER_IN_LIMIT	788
8.67.1.20	PLL_MAX_CCO_FREQ_MHZ	788
8.67.1.21	PLL_MAX_IN_SSMODE	788
8.67.1.22	PLL_MAX_N_DIV	788
8.67.1.23	PLL_MIN_CCO_FREQ_MHZ	788
8.67.1.24	PLL_MIN_IN_SSMODE	788
8.67.1.25	PLL_NDEC_NREQ	788
8.67.1.26	PLL_NDEC_NREQ_P	788
8.67.1.27	PLL_NDEC_VAL_M	788
8.67.1.28	PLL_NDEC_VAL_P	788
8.67.1.29	PLL_NDEC_VAL_SET	788
8.67.1.30	PLL_PDEC_PREQ	788
8.67.1.31	PLL_PDEC_PREQ_P	789
8.67.1.32	PLL_PDEC_VAL_M	789
8.67.1.33	PLL_PDEC_VAL_P	789
8.67.1.34	PLL_PDEC_VAL_SET	789
8.67.1.35	PLL_SSCG0_MDEC_VAL_M	789
8.67.1.36	PLL_SSCG0_MDEC_VAL_P	789
8.67.1.37	PLL_SSCG0_MDEC_VAL_SET	789
8.67.1.38	PLL_SSCG0_MREQ	789
8.67.1.39	PLL_SSCG0_MREQ	789
8.67.1.40	PLL_SSCG0_MREQ_P	789
8.67.1.41	PLL_SSCG0_MREQ_P	789
8.67.1.42	PLL_SSCG0_SEL_EXT_SSCG_N	789

8.67.1.43 PLL_SSCG0_SEL_EXT_SSCG_N . . . . .	790
8.67.1.44 PLL_SSCG0_SEL_EXT_SSCG_N_P . . . . .	790
8.67.1.45 PLL_SSCG0_SEL_EXT_SSCG_N_P . . . . .	790
8.67.1.46 PLL_SSCG1_DITHER . . . . .	790
8.67.1.47 PLL_SSCG1_DITHER_P . . . . .	790
8.67.1.48 PLL_SSCG1_MC_M . . . . .	790
8.67.1.49 PLL_SSCG1_MC_P . . . . .	790
8.67.1.50 PLL_SSCG1_MD_FRACT_M . . . . .	790
8.67.1.51 PLL_SSCG1_MD_FRACT_P . . . . .	790
8.67.1.52 PLL_SSCG1_MD_FRACT_SET . . . . .	790
8.67.1.53 PLL_SSCG1_MD_INT_M . . . . .	790
8.67.1.54 PLL_SSCG1_MD_INT_P . . . . .	791
8.67.1.55 PLL_SSCG1_MD_INT_SET . . . . .	791
8.67.1.56 PLL_SSCG1_MD_REQ . . . . .	791
8.67.1.57 PLL_SSCG1_MD_REQ_P . . . . .	791
8.67.1.58 PLL_SSCG1_MF_M . . . . .	791
8.67.1.59 PLL_SSCG1_MF_P . . . . .	791
8.67.1.60 PLL_SSCG1_MOD_PD_SSCGCLK_N . . . . .	791
8.67.1.61 PLL_SSCG1_MOD_PD_SSCGCLK_N_P . . . . .	791
8.67.1.62 PLL_SSCG1_MR_M . . . . .	791
8.67.1.63 PLL_SSCG1_MR_P . . . . .	791
8.67.1.64 PLL_SSCG_DITHER_VALUE . . . . .	791
8.67.1.65 PLL_SSCG_MC_COMP_VALUE . . . . .	792
8.67.1.66 PLL_SSCG_MF_FREQ_VALUE . . . . .	792
8.67.1.67 PLL_SSCG_MR_DEPTH_VALUE . . . . .	792
8.67.1.68 PVALMAX . . . . .	792
8.67.1.69 SYS_PLL_BANDSEL . . . . .	792
8.67.1.70 SYS_PLL_BYPASS . . . . .	792
8.67.1.71 SYS_PLL_BYPASSCCODIV2 . . . . .	792
8.67.1.72 SYS_PLL_DIRECTI . . . . .	792
8.67.1.73 SYS_PLL_DIRECTO . . . . .	792
8.67.1.74 SYS_PLL_SEL_I . . . . .	792
8.67.1.75 SYS_PLL_SELP . . . . .	793
8.67.1.76 SYS_PLL_SEL_R . . . . .	793
8.67.1.77 SYS_PLL_UPLIMOFF . . . . .	793
8.67.1.78 SYSCON_SYSPLLCTRL_BANDSEL_SSCGREG_N . . . . .	793
8.67.1.79 SYSCON_SYSPLLCTRL_BANDSEL_SSCGREG_N_P . . . . .	793
8.67.1.80 SYSCON_SYSPLLCTRL_BYPASS . . . . .	793
8.67.1.81 SYSCON_SYSPLLCTRL_BYPASS_FBDIV2 . . . . .	793
8.67.1.82 SYSCON_SYSPLLCTRL_BYPASS_FBDIV2_P . . . . .	793

8.67.1.83	SYSCON_SYSPLLCTRL_BYPASS_P	793
8.67.1.84	SYSCON_SYSPLLCTRL_DIRECTI	793
8.67.1.85	SYSCON_SYSPLLCTRL_DIRECTI_P	793
8.67.1.86	SYSCON_SYSPLLCTRL_DIRECTO	794
8.67.1.87	SYSCON_SYSPLLCTRL_DIRECTO_P	794
8.67.1.88	SYSCON_SYSPLLCTRL_SEL_I_M	794
8.67.1.89	SYSCON_SYSPLLCTRL_SEL_I_P	794
8.67.1.90	SYSCON_SYSPLLCTRL_SEL_P_M	794
8.67.1.91	SYSCON_SYSPLLCTRL_SEL_P_P	794
8.67.1.92	SYSCON_SYSPLLCTRL_SEL_R_M	794
8.67.1.93	SYSCON_SYSPLLCTRL_SEL_R_P	794
8.67.1.94	SYSCON_SYSPLLCTRL_UPLIMOFF	794
8.67.1.95	SYSCON_SYSPLLCTRL_UPLIMOFF_P	794
8.67.1.96	SYSCON_SYSPLLSTAT_LOCK	794
8.67.1.97	SYSCON_SYSPLLSTAT_LOCK_P	794
8.67.2	Function Documentation	795
8.67.2.1	Chip_Clock_GetPllConfig(uint32_t finHz, uint32_t foutHz, PLL_SETUP_T *pSetup, bool useFeedbackDiv2, bool useSS)	795
8.67.2.2	Chip_Clock_GetSystemPLLOutFromSetupUpdate(PLL_SETUP_T *pSetup)	795
8.67.2.3	FindGreatestCommonDivisor(uint32_t m, uint32_t n)	795
8.67.2.4	findPllMMult(uint32_t ctrlReg, uint32_t mDecReg)	795
8.67.2.5	findPllPostDiv(uint32_t ctrlReg, uint32_t pDecReg)	795
8.67.2.6	findPllPreDiv(uint32_t ctrlReg, uint32_t nDecReg)	795
8.67.2.7	pllDecodeM(uint32_t MDEC)	795
8.67.2.8	pllDecodeN(uint32_t NDEC)	795
8.67.2.9	pllDecodeP(uint32_t PDEC)	795
8.67.2.10	pllEncodeM(uint32_t M)	795
8.67.2.11	pllEncodeN(uint32_t N)	795
8.67.2.12	pllEncodeP(uint32_t P)	795
8.67.2.13	pllFindSel(uint32_t M, bool bypassFBDIV2, uint32_t *pSelP, uint32_t *pSelI, uint32_t *pSelR)	796
8.67.3	Variable Documentation	796
8.67.3.1	curPllRate	796
8.68	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/ring_buffer.c File Reference	796
8.68.1	Macro Definition Documentation	796
8.68.1.1	RB_INDH	796
8.68.1.2	RB_INDT	796
8.69	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src rtc_ut.c File Reference	796
8.69.1	Macro Definition Documentation	797
8.69.1.1	DAYS PER LEAP YEAR	797

8.69.1.2	<a href="#">DAYSPERWEEK</a>	797
8.69.1.3	<a href="#">DAYSPERYEAR</a>	797
8.69.1.4	<a href="#">HOURSPERDAY</a>	797
8.69.1.5	<a href="#">MINSPERHOUR</a>	797
8.69.1.6	<a href="#">MONETHSPERYEAR</a>	797
8.69.1.7	<a href="#">SECSPERDAY</a>	798
8.69.1.8	<a href="#">SECSPERHOUR</a>	798
8.69.1.9	<a href="#">SECSPERMIN</a>	798
8.69.2	<a href="#">Function Documentation</a>	798
8.69.2.1	<a href="#">GetDMLY(int dayOff, struct tm *pTime)</a>	798
8.69.3	<a href="#">Variable Documentation</a>	798
8.69.3.1	<a href="#">daysPerMonth</a>	798
8.70	<a href="#">C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/sct_5411x.c File Reference</a>	798
8.71	<a href="#">C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/sct_pwm_5411x.c File Reference</a>	798
8.72	<a href="#">C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/spi_common_5411x.c File Reference</a>	799
8.73	<a href="#">C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/spim_5411x.c File Reference</a>	799
8.73.1	<a href="#">Function Documentation</a>	799
8.73.1.1	<a href="#">get_tx_data(SPIM_XFER_T *xfer)</a>	799
8.73.1.2	<a href="#">put_rx_data(SPIM_XFER_T *xfer, uint16_t data)</a>	800
8.73.1.3	<a href="#">spim_do_txx(LPC_SPI_T *pSPI, uint32_t stat, SPIM_XFER_T *xfer)</a>	800
8.73.1.4	<a href="#">spim_get_xfercfg(const SPIM_XFER_T *xfer)</a>	800
8.74	<a href="#">C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/spis_5411x.c File Reference</a>	800
8.74.1	<a href="#">Function Documentation</a>	800
8.74.1.1	<a href="#">get_tx_data(SPIS_XFER_T *xfer)</a>	800
8.74.1.2	<a href="#">put_rx_data(SPIS_XFER_T *xfer, uint16_t data)</a>	800
8.74.1.3	<a href="#">spis_do_txx(LPC_SPI_T *pSPI, uint32_t stat, SPIS_XFER_T *xfer)</a>	800
8.75	<a href="#">C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/stopwatch_5411x.c File Reference</a>	801
8.75.1	<a href="#">Macro Definition Documentation</a>	801
8.75.1.1	<a href="#">LPC_TIMER32_1</a>	801
8.75.2	<a href="#">Variable Documentation</a>	801
8.75.2.1	<a href="#">ticksPerMs</a>	801
8.75.2.2	<a href="#">ticksPerSecond</a>	801
8.75.2.3	<a href="#">ticksPerUs</a>	802
8.76	<a href="#">C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/syscon_5411x.c File Reference</a>	802
8.77	<a href="#">C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/sysinit_5411x.c File Reference</a>	802
8.77.1	<a href="#">Function Documentation</a>	802
8.77.1.1	<a href="#">setupFlashClocks(uint32_t freq)</a>	802
8.78	<a href="#">C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/timer_5411x.c File Reference</a>	803
8.78.1	<a href="#">Macro Definition Documentation</a>	803
8.78.1.1	<a href="#">LAST_TIMER</a>	803

8.79 C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/uart_5411x.c File Reference . . . . .	803
8.79.1 Function Documentation . . . . .	804
8.79.1.1 _CalcErr(uint32_t n, uint32_t d, uint32_t *prev) . . . . .	804
8.79.1.2 _UART_CalcDiv(UART_BAUD_T *ub) . . . . .	804
8.79.1.3 _UART_CalcMul(UART_BAUD_T *ub) . . . . .	804
8.79.1.4 _UART_DivClk(uint32_t pclk, uint32_t m) . . . . .	804
8.79.1.5 _UART_GetHighDiv(uint32_t val, uint8_t strict) . . . . .	804
8.79.1.6 Chip_UART_CalculateBaud(UART_BAUD_T *baud) . . . . .	804
8.79.2 Variable Documentation . . . . .	805
8.79.2.1 fifo_config_dma . . . . .	805
8.79.2.2 fifo_config_int . . . . .	805
8.80 C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/cr_startup_lpc5411x-m0.c File Reference . . . . .	805
8.80.1 Macro Definition Documentation . . . . .	805
8.80.1.1 ALIAS . . . . .	805
8.80.1.2 WEAK . . . . .	805
8.80.2 Function Documentation . . . . .	806
8.80.2.1 __attribute__((section("".after_vectors"))) . . . . .	806
8.80.2.2 __attribute__((section("".after_vectors.reset"))) . . . . .	806
8.80.2.3 HardFault_Handler(void) . . . . .	806
8.80.2.4 IntDefaultHandler(void) . . . . .	806
8.80.2.5 NMI_Handler(void) . . . . .	806
8.80.2.6 PendSV_Handler(void) . . . . .	806
8.80.2.7 ResetISR(void) . . . . .	806
8.80.2.8 SVC_Handler(void) . . . . .	806
8.80.2.9 SysTick_Handler(void) . . . . .	806
8.80.2.10 WDT_BOD_IRQHandler(void DMA_IRQHandler() ALIAS(IntDefaultHandler) void) . . . . .	806
8.80.3 Variable Documentation . . . . .	806
8.80.3.1 __bss_section_table . . . . .	806
8.80.3.2 __bss_section_table_end . . . . .	806
8.80.3.3 __data_section_table . . . . .	806
8.80.3.4 __data_section_table_end . . . . .	806
8.81 C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/cr_startup_lpc5411x.c File Reference . . . . .	806
8.81.1 Macro Definition Documentation . . . . .	807
8.81.1.1 ALIAS . . . . .	807
8.81.1.2 WEAK . . . . .	807
8.81.2 Function Documentation . . . . .	807
8.81.2.1 __attribute__((section("".after_vectors"))) . . . . .	807
8.81.2.2 __attribute__((section("".after_vectors.reset"))) . . . . .	807
8.81.2.3 BusFault_Handler(void) . . . . .	807
8.81.2.4 DebugMon_Handler(void) . . . . .	807

8.81.2.5	HardFault_Handler(void)	807
8.81.2.6	IntDefaultHandler(void)	807
8.81.2.7	MemManage_Handler(void)	807
8.81.2.8	NMI_Handler(void)	807
8.81.2.9	PendSV_Handler(void)	808
8.81.2.10	ResetISR(void)	808
8.81.2.11	SVC_Handler(void)	808
8.81.2.12	SysTick_Handler(void)	808
8.81.2.13	UsageFault_Handler(void)	808
8.81.2.14	WDT_BOD_IRQHandler(void DMA_IRQHandler() ALIAS(IntDefaultHandler) void)	808
8.81.3	Variable Documentation	808
8.81.3.1	__bss_section_table	808
8.81.3.2	__bss_section_table_end	808
8.81.3.3	__data_section_table	808
8.81.3.4	__data_section_table_end	808
8.81.3.5	__SWVtrace_Enabled	808
8.82	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/crp.c File Reference	808
8.83	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/mtb.c File Reference	808
8.84	C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/sysinit.c File Reference	808
8.84.1	Function Documentation	808
8.84.1.1	SystemInit(void)	808
8.85	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/community_support.dox File Reference	808
8.86	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/copyright.dox File Reference	809
8.87	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/installation_insts.dox File Reference	809
8.88	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/toolchains.dox File Reference	809
8.89	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/version_history.dox File Reference	809
8.90	boards_5411x.dox File Reference	809
8.91	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox_lpc5411x/chip_5411x/chip_5411x.dox File Reference	809
8.92	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox_lpc5411x/chip_5411x/irq_vector_names.dox File Reference	809
8.93	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox_lpc5411x/chip_5411x/lpcopen_layout.dox File Reference	809
8.94	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox_lpc5411x/chip_5411x/main_layout.dox File Reference	809
8.95	C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox_lpc5411x/chip_5411x/multicore.dox File Reference	809





## Chapter 1

# LPCOpen Platform for the NXP LPC5411X family of Microcontrollers

### Introduction

This documentation describes the LPCOpen Platform software API and examples for NXP LPC5411X family of microcontrollers.

### LPCOpen summary page on LPCware.com

Visit the [LPCOpen page on LPCware.com](#) for more resources on the LPCOpen platform.

### LPCOpen quickstart guides

Visit the [LPCOpen quickstart guides](#) page to get started building LPCOpen projects.

### LPCOpen software API documentation

Software API documentation provides API descriptions for the various drivers in LPCOpen.

- [Chip Driver Layer](#) Chip specific drivers
- [Board Support Layer](#) Board support code for various support boards
- [Supported RTOSes](#) such as FreeRTOS and uC/OS-III

Copyright (C) 2013 NXP Semiconductors. All rights reserved.



## Chapter 2

# MISRA-C:2004 Compliance Exceptions

CMSIS violates the following MISRA-C:2004 rules:

- Required Rule 8.5, object/function definition in header file.  
Function definitions in header files are used to allow 'inlining'.
- Required Rule 18.4, declaration of union type or object of union type: '{...}'.  
Unions are used for effective representation of core registers.
- Advisory Rule 19.7, Function-like macro defined.  
Function-like macros are used to allow more efficient code.



## Chapter 3

# Module Index

### 3.1 Modules

Here is a list of all modules:

Board specific drivers and support functions . . . . .	17
BOARD: Common board components used with board drivers . . . . .	15
BOARD: LPC5411X boards . . . . .	16
NXP LPCXpresso LPC54114 LQFP board . . . . .	491
CHIP: CHIP_LPC5411X family IRQ vector names and mapped NVIC IRQ numbers . . . . .	44
CHIP: LPC5411X M0 core CMSIS include file . . . . .	197
CHIP: LPC5411X M0 core Cortex CMSIS definitions . . . . .	198
CHIP_5411X: LPC5411X M0 core peripheral interrupt numbers . . . . .	428
CHIP: LPC5411X M4 core CMSIS include file . . . . .	201
CHIP: LPC5411X M4 core Cortex CMSIS definitions . . . . .	202
CHIP_5411X: LPC5411X M4 core peripheral interrupt numbers . . . . .	430
CMSIS Core Instruction Interface . . . . .	432
CMSIS Global Defines . . . . .	434
CMSIS SIMD Intrinsics . . . . .	435
CMSIS support . . . . .	436
Chip specific drivers . . . . .	437
Common components used with chip drivers . . . . .	440
CHIP: Common Chip ISP/IAP commands and return codes . . . . .	46
CHIP: FPU initialization . . . . .	54
CHIP: LPC Common Types . . . . .	55
LPC Public Macros . . . . .	475
LPC Public Types . . . . .	477
CHIP: RTC tick to (a more) Universal Time conversion functions . . . . .	417
CHIP: Simple ring buffer implementation . . . . .	419
CHIP: Stopwatch primitives. . . . .	424
LPC5411x Chip specific drivers . . . . .	483
CHIP: LPC5411X A/D conversion driver . . . . .	18
CHIP: LPC5410x family CMSIS include files . . . . .	56
CHIP: LPC5411X 32-bit Timer driver . . . . .	57
CHIP: LPC5411X CPU multi-core support driver . . . . .	72
CHIP: LPC5411X Clock Driver . . . . .	74
CHIP: LPC5411X Cyclic Redundancy Check Engine driver . . . . .	97
CHIP: LPC5411X DMA Engine driver (legacy) . . . . .	121
CHIP: LPC5411X DMA Controller driver channel specific functions (legacy) . . . . .	105
CHIP: LPC5411X DMA Controller driver common channel functions (legacy) . . . . .	111
CHIP: LPC5411X DMA Controller driver common functions (legacy) . . . . .	118

CHIP: LPC5411X DMA Service driver . . . . .	129
CHIP: LPC5411X DMIC driver . . . . .	133
CHIP: LPC5411X Enhanced boot block support . . . . .	148
CHIP: LPC5411X GPIO driver . . . . .	150
CHIP: LPC5411X GPIO group driver . . . . .	165
CHIP: LPC5411X IOCON register block and driver . . . . .	188
CHIP: LPC5411X Input Mux Registers and Driver . . . . .	194
CHIP: LPC5411X Mailbox M4/M0+ driver . . . . .	205
CHIP: LPC5411X Micro Tick driver . . . . .	210
CHIP: LPC5411X Multi-Rate Timer driver . . . . .	214
CHIP: LPC5411X PLL Driver . . . . .	224
CHIP: LPC5411X Peripheral addresses and register set declarations . . . . .	232
CHIP: LPC5411X Pin Interrupt and Pattern Match driver . . . . .	239
CHIP: LPC5411X Power LIBRARY functions . . . . .	250
CHIP: LPC5411X Power Management declarations and functions . . . . .	254
CHIP: LPC5411X ROM API declarations and functions . . . . .	257
CHIP: LPC5411X Real Time clock . . . . .	258
CHIP: LPC5411X SPI driver . . . . .	268
CHIP: LPC5411X SPI master driver . . . . .	294
CHIP: LPC5411X SPI slave driver . . . . .	300
CHIP: LPC5411X State Configurable Timer PWM driver . . . . .	303
CHIP: LPC5411X State Configurable Timer driver . . . . .	307
CHIP: LPC5411X System and Control Driver . . . . .	324
CHIP: LPC5411X UART Driver . . . . .	341
CHIP: LPC5411X Windowed Watchdog driver . . . . .	371
CHIP: LPC5411X flexcomm API . . . . .	378
CHIP: LPC5411X support functions . . . . .	385
CHIP: LPC5411x Chip driver build time options . . . . .	388
CHIP: LPC5411x I2C driver . . . . .	389
CHIP: LPC5411X I2C master-only driver . . . . .	171
CHIP: LPC5411X I2C slave-only driver . . . . .	179
LPC5411X multi-core use in LPCOpen . . . . .	481
Community support for LPCOpen . . . . .	441
Copyright (C) 2013 NXP Semiconductors. All rights reserved. . . . .	442
Defines and Type Definitions . . . . .	464
Core Debug Registers (CoreDebug) . . . . .	443
Core Definitions . . . . .	451
Data Watchpoint and Trace (DWT) . . . . .	455
Instrumentation Trace Macrocell (ITM) . . . . .	469
Nested Vectored Interrupt Controller (NVIC) . . . . .	492
Status and Control Registers . . . . .	495
System Control Block (SCB) . . . . .	498
System Controls not in SCB (SCnSCB) . . . . .	520
System Tick Timer (SysTick) . . . . .	522
Trace Port Interface (TPI) . . . . .	527
Functions and Instructions Reference . . . . .	465
CMSIS Core Register Access Functions . . . . .	433
ITM Functions . . . . .	467
NVIC Functions . . . . .	486
SysTick Functions . . . . .	497
LPCOpen download and installation information . . . . .	484
emWin download and installation . . . . .	548
LPCOpen versioning and release history . . . . .	485
RTOS support code . . . . .	493
RTOS: FreeRTOS support code . . . . .	494
Common FreeRTOS functions shared with multiple platforms . . . . .	439
Supported toolchains in LPCOpen . . . . .	496

---

Code Red LPCXpresso support in LPCOpen . . . . .	438
IAR EWARM support in LPCOpen . . . . .	466
Keil uVision support in LPCOpen . . . . .	474
USBD_Core . . . . .	537





## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">APSR_Type</a>	Union type to access the Application Program Status Register (APSR) . . . . .	549
<a href="#">BM_T</a>	. . . . .	551
<a href="#">CONTROL_Type</a>	Union type to access the Control Registers (CONTROL) . . . . .	551
<a href="#">CoreDebug_Type</a>	Structure type to access the Core Debug Register (CoreDebug) . . . . .	553
<a href="#">DMA_CHDESC_T</a>	. . . . .	554
<a href="#">DMA_DUAL_DESCRIPTOR_T</a>	. . . . .	555
<a href="#">DMA_PERIPHERAL_CONTEXT_T</a>	. . . . .	555
<a href="#">DMIC_CHANNEL_CONFIG_T</a>	. . . . .	555
<a href="#">DMIC_STATISTICS_T</a>	DMIC statistics structure . . . . .	556
<a href="#">DWT_Type</a>	Structure type to access the Data Watchpoint and Trace Register (DWT) . . . . .	556
<a href="#">I2CM_XFER_T</a>	Master transfer data structure definitions . . . . .	559
<a href="#">I2CS_XFER_T</a>	. . . . .	560
<a href="#">I2S_AUDIO_FORMAT_T</a>	. . . . .	562
<a href="#">I2S_STATISTICS_T</a>	I2S statistics structure . . . . .	564
<a href="#">IPSR_Type</a>	Union type to access the Interrupt Program Status Register (IPSR) . . . . .	565
<a href="#">ITM_Type</a>	Structure type to access the Instrumentation Trace Macrocell Register (ITM) . . . . .	566
<a href="#">LPC_ADC_T</a>	ADC register block structure . . . . .	570
<a href="#">LPC_ASYNC_SYSCON_T</a>	LPC5411X Asynchronous system configuration register block structure . . . . .	572
<a href="#">LPC_CRC_T</a>	CRC register block structure . . . . .	574
<a href="#">LPC_DMA_CHANNEL_T</a>	DMA Controller shared registers structure . . . . .	575
<a href="#">LPC_DMA_COMMON_T</a>	DMA Controller shared registers structure . . . . .	576
<a href="#">LPC_DMA_T</a>	DMA Controller register block structure . . . . .	579
<a href="#">LPC_DMIC_Channel_Type</a>	. . . . .	580

<a href="#">LPC_DMIC_T</a>	580
<a href="#">LPC_GPIO_T</a>	
GPIO port register block structure	581
<a href="#">LPC_GPIOGROUPINT_T</a>	
GPIO grouped interrupt register block structure	582
<a href="#">LPC_I2S_T</a>	
I2S register block structure	583
<a href="#">LPC_INMUX_T</a>	
LPC5411X Input Mux Register Block Structure	586
<a href="#">LPC_IOCON_T</a>	
LPC5411X IO Configuration Unit register block structure	588
<a href="#">LPC_MBOX_T</a>	588
<a href="#">LPC_MBOXIRQ_T</a>	589
<a href="#">LPC_MRT_CH_T</a>	
MRT register block structure	590
<a href="#">LPC_MRT_T</a>	
MRT register block structure	590
<a href="#">LPC_PIN_INT_T</a>	
LPC5411X Pin Interrupt and Pattern Match register block structure	591
<a href="#">LPC_PMU_T</a>	
PMU register block structure	593
<a href="#">LPC_ROM_API_T</a>	
High level ROM API structure	594
<a href="#">LPC_RTC_T</a>	
LPC5411X Real Time clock register block structure	596
<a href="#">LPC_SCT_T</a>	
State Configurable Timer register block structure	597
<a href="#">LPC_SPI_T</a>	
SPI register block structure	604
<a href="#">LPC_SYSCON_T</a>	
LPC5411X Main system configuration register block structure	607
<a href="#">LPC_TIMER_T</a>	
32-bit Standard timer register block structure	619
<a href="#">LPC_USART_T</a>	
UART Registers	621
<a href="#">LPC_UTICK_T</a>	
Micro Tick register block structure	624
<a href="#">LPC_WWDT_T</a>	
Windowed Watchdog register block structure	624
<a href="#">NVIC_Type</a>	
Structure type to access the Nested Vectored Interrupt Controller (NVIC)	625
<a href="#">PINMUX_GRP_T</a>	
Array of IOCON pin definitions passed to <a href="#">Chip_IOCON_SetPinMuxing()</a> must be in this format	627
<a href="#">PINTABLE_T</a>	
LPC5411X Pin table structure used for enhanced boot block support	628
<a href="#">PLL_CONFIG_T</a>	
PLL configuration structure This structure can be used to configure the settings for a PLL setup structure. Fill in the desired configuration for the PLL and call the PLL setup function to fill in a PLL setup structure	630
<a href="#">PLL_SETUP_T</a>	
PLL setup structure This structure can be used to pre-build a PLL setup configuration at run-time and quickly set the PLL to the configuration. It can be populated with the PLL setup function. If powering up or waiting for PLL lock, the PLL input clock source should be configured prior to PLL setup	631
<a href="#">REQUEST_TYPE</a>	632
<a href="#">RINGBUFF_T</a>	
Ring buffer structure	633

SCB_Type	Structure type to access the System Control Block (SCB) . . . . .	634
SCnSCB_Type	Structure type to access the System Control and ID Register not in the SCB . . . . .	637
SPI_CFGSETUP_T	. . . . .	637
SPIM_DELAY_CONFIG_T	SPI Delay Configure Struct . . . . .	638
SPIM_XFER_T	SPI Master transfer data context . . . . .	639
SPIS_XFER_T	. . . . .	641
SysTick_Type	Structure type to access the System Timer (SysTick) . . . . .	643
TPI_Type	Structure type to access the Trace Port Interface Register (TPI) . . . . .	644
UART_BAUD_T	UART Baud rate calculation structure . . . . .	647
UART_STATISTICS_T	UART statistics structure . . . . .	648
USB_COMMON_DESCRIPTOR	. . . . .	650
USB_CONFIGURATION_DESCRIPTOR	. . . . .	651
USB_DEVICE_DESCRIPTOR	. . . . .	652
USB_DEVICE_QUALIFIER_DESCRIPTOR	. . . . .	655
USB_ENDPOINT_DESCRIPTOR	. . . . .	656
USB_IAD_DESCRIPTOR	. . . . .	659
USB_INTERFACE_DESCRIPTOR	. . . . .	660
USB_OTHER_SPEED_CONFIGURATION	. . . . .	662
USB_SETUP_PACKET	. . . . .	663
USB_STRING_DESCRIPTOR	. . . . .	664
WB_T	. . . . .	665
WORD_BYTE	. . . . .	665
xPSR_Type	Union type to access the Special-Purpose Program Status Registers (xPSR) . . . . .	666



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">adc_5411x.h</a> . . . . .	669
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">chip.h</a> . . . . .	672
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">clock_5411x.h</a> . . . . .	675
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">cmsis.h</a> . . . . .	679
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">cmsis_5411x.h</a> . . . . .	679
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">cmsis_5411x_m0.h</a> . . . . .	680
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">core_cm0plus.h</a> CMSIS Cortex-M0+ Core Peripheral Access Layer Header File . . . . .	681
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">core_cm4.h</a> CMSIS Cortex-M4 Core Peripheral Access Layer Header File . . . . .	685
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">core_cm4_simd.h</a> CMSIS Cortex-M4 SIMD Header File . . . . .	695
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">core_cmFunc.h</a> CMSIS Cortex-M Core Function Access Header File . . . . .	696
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">core_cmInstr.h</a> CMSIS Cortex-M Core Instruction Access Header File . . . . .	696
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">cpuctrl_5411x.h</a> . . . . .	696
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">crc_5411x.h</a> . . . . .	697
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">dma_5411x.h</a> . . . . .	698
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">dma_service_5411x.h</a> . . . . .	701
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">dmic_5411x.h</a> . . . . .	702
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">error.h</a> . . . . .	703
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">flexcomm_5411x.h</a> . . . . .	710
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">fpu_init.h</a> . . . . .	711
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">gpio_5411x.h</a> . . . . .	711
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">gpiogroup_5411x.h</a> . . . . .	712
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">i2c_common_5411x.h</a> . . . . .	713
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">i2cm_5411x.h</a> . . . . .	717
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">i2cs_5411x.h</a> . . . . .	718
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">i2s_5411x.h</a> . . . . .	720
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">iap.h</a> . . . . .	738
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">inmux_5411x.h</a> . . . . .	739
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">iocon_5411x.h</a> . . . . .	740
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">lpc_assert.h</a> . . . . .	741
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">lpc_types.h</a> . . . . .	741
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">mailbox_5411x.h</a> . . . . .	742
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">mrt_5411x.h</a> . . . . .	743
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ <a href="#">packing.h</a> . . . . .	745

C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pinint_5411x.h	745
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pintable_5411x.h	747
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pll_5411x.h	747
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pmu_5411x.h	749
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/power_lib_5411x.h	749
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/ring_buffer.h	750
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/romapi_5411x.h	751
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/rtc_5411x.h	751
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/rtc_ut.h	752
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/sct_5411x.h	753
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/sct_pwm_5411x.h	756
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/spi_common_5411x.h	756
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/spim_5411x.h	760
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/spis_5411x.h	761
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/stopwatch.h	762
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/syscon_5411x.h	762
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/timer_5411x.h	765
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/uart_5411x.h	767
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/usbd.h	
Common definitions and declarations for the USB stack	771
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/utick_5411x.h	774
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/wwdt_5411x.h	774
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/adc_5411x.c	775
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/chip_5411x.c	776
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/clock_5411x.c	776
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/dma_5411x.c	778
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/dma_service_5411x.c	778
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/dmic_5411x.c	779
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/flexcomm_5411x.c	780
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/fpu_init.c	780
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/i2cm_5411x.c	780
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/i2cs_5411x.c	780
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/i2s_5411x.c	781
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/iap.c	783
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/pll_5411x.c	783
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/ring_buffer.c	796
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/rtc_ut.c	796
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/sct_5411x.c	798
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/sct_pwm_5411x.c	798
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/spi_common_5411x.c	799
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/spim_5411x.c	799
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/spis_5411x.c	800
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/stopwatch_5411x.c	801
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/syscon_5411x.c	802
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/sysinit_5411x.c	802
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/timer_5411x.c	803
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/src/uart_5411x.c	803
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/cr_startup_lpc5411x-m0.c	805
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/cr_startup_lpc5411x.c	806
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/crp.c	808
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/mtb.c	808
C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/startup/sysinit.c	808

## **Chapter 6**

# **Module Documentation**

### **6.1 BOARD: Common board components used with board drivers**

## 6.2 BOARD: LPC5411X boards

### 6.2.1 Detailed Description

The following LPC5411X boards are supported in LPCOpen.

#### Modules

- [NXP LPCXpresso LPC54114 LQFP board](#)



## 6.3 Board specific drivers and support functions

### 6.3.1 Detailed Description

Board specific setup code includes code for setting up a board's default memory, clocking, muxing, or other interfaces. The default UART port for debug output is also setup as part of the board layer.

#### Modules

- [BOARD: Common board components used with board drivers](#)
- [BOARD: LPC5411X boards](#)

## 6.4 CHIP: LPC5411X A/D conversion driver

### 6.4.1 Detailed Description

#### Data Structures

- struct [LPC\\_ADC\\_T](#)  
*ADC register block structure.*

#### Macros

- #define [ADC\\_MAX\\_SAMPLE\\_RATE](#) 80000000
- #define [ADC\\_MAX\\_CHANNEL\\_NUM](#) 12
- #define [ADC\\_CR\\_CLKDIV\\_MASK](#) (0xFF << 0)  
*ADC register support bitfields and mask.*
- #define [ADC\\_CR\\_CLKDIV\\_BITPOS](#) (0)
- #define [ADC\\_CR\\_ASYNC\\_MODE](#) (1 << 8)
- #define [ADC\\_CR\\_RESOL](#)(n) ((n) << 9)
- #define [ADC\\_CR\\_LPWRMODEBIT](#) (1 << 10)
- #define [ADC\\_CR\\_BYPASS](#) (1 << 11)
- #define [ADC\\_CR\\_TSAMP](#)(n) ((n) << 12)
- #define [ADC\\_CR\\_CALMODEBIT](#) (1 << 30)
- #define [ADC\\_CR\\_BITACC](#)(n) (((n) & 0x1) << 9))
- #define [ADC\\_CR\\_CLKDIV](#)(n) (((n) & 0xFF) << 0))
- #define [ADC\\_SAMPLE\\_RATE\\_CONFIG\\_MASK](#) ([ADC\\_CR\\_CLKDIV](#)(0xFF) | [ADC\\_CR\\_BITACC](#)(0x01))
- #define [ADC\\_SEQ\\_CTRL\\_CHANNEL\\_EN](#)(n) (1 << n)
- #define [ADC\\_SEQ\\_CTRL\\_TRIGGER](#)(n) ((n & 0x3f) << 12)
- #define [ADC\\_SEQ\\_CTRL\\_HWTRIG\\_POLPOS](#) (1 << 18)
- #define [ADC\\_SEQ\\_CTRL\\_HWTRIG\\_SYNCBYPASS](#) (1 << 19)
- #define [ADC\\_SEQ\\_CTRL\\_START](#) (1 << 26)
- #define [ADC\\_SEQ\\_CTRL\\_BURST](#) (1 << 27)
- #define [ADC\\_SEQ\\_CTRL\\_SINGLESTEP](#) (1 << 28)
- #define [ADC\\_SEQ\\_CTRL\\_LOWPRIO](#) (1 << 29)
- #define [ADC\\_SEQ\\_CTRL\\_MODE\\_EOS](#) (1 << 30)
- #define [ADC\\_SEQ\\_CTRL\\_SEQ\\_ENA](#) (1UL << 31)
- #define [ADC\\_SEQ\\_GDAT\\_RESULT\\_MASK](#) (0xFFFF << 4)
- #define [ADC\\_SEQ\\_GDAT\\_RESULT\\_BITPOS](#) (4)
- #define [ADC\\_SEQ\\_GDAT\\_THCMPRANGE\\_MASK](#) (0x3 << 16)
- #define [ADC\\_SEQ\\_GDAT\\_THCMPRANGE\\_BITPOS](#) (16)
- #define [ADC\\_SEQ\\_GDAT\\_THCMPCROSS\\_MASK](#) (0x3 << 18)
- #define [ADC\\_SEQ\\_GDAT\\_THCMPCROSS\\_BITPOS](#) (18)
- #define [ADC\\_SEQ\\_GDAT\\_CHAN\\_MASK](#) (0xF << 26)
- #define [ADC\\_SEQ\\_GDAT\\_CHAN\\_BITPOS](#) (26)
- #define [ADC\\_SEQ\\_GDAT\\_OVERRUN](#) (1 << 30)
- #define [ADC\\_SEQ\\_GDAT\\_DATAVALID](#) (1UL << 31)
- #define [ADC\\_DR\\_RESULT\\_BITPOS](#) (4)
- #define [ADC\\_DR\\_RESULT](#)(n) (((n) >> 4) & 0xFFFF)
- #define [ADC\\_DR\\_THCMPRANGE\\_MASK](#) (0x3 << 16)
- #define [ADC\\_DR\\_THCMPRANGE\\_BITPOS](#) (16)
- #define [ADC\\_DR\\_THCMPRANGE](#)(n) (((n) >> [ADC\\_DR\\_THCMPRANGE\\_BITPOS](#)) & 0x3)
- #define [ADC\\_DR\\_THCMPCROSS\\_MASK](#) (0x3 << 18)
- #define [ADC\\_DR\\_THCMPCROSS\\_BITPOS](#) (18)
- #define [ADC\\_DR\\_THCMPCROSS](#)(n) (((n) >> [ADC\\_DR\\_THCMPCROSS\\_BITPOS](#)) & 0x3)

- #define `ADC_DR_CHAN_MASK` (0xF << 26)
- #define `ADC_DR_CHAN_BITPOS` (26)
- #define `ADC_DR_CHANNEL`(n) (((n) >> `ADC_DR_CHAN_BITPOS`) & 0xF)
- #define `ADC_DR_OVERRUN` (1 << 30)
- #define `ADC_DR_DATAVALID` (1UL << 31)
- #define `ADC_DR_DONE`(n) (((n) >> 31))
- #define `ADC_THR_VAL_MASK` (0xFFF << 4)
- #define `ADC_THR_VAL_POS` (4)
- #define `ADC_THRSEL_CHAN_SEL_THR1`(n) (1 << (n))
- #define `ADC_INTEN_SEQA_ENABLE` (1 << 0)
- #define `ADC_INTEN_SEQB_ENABLE` (1 << 1)
- #define `ADC_INTEN_SEQN_ENABLE`(seq) (1 << (seq))
- #define `ADC_INTEN_OVRRUN_ENABLE` (1 << 2)
- #define `ADC_INTEN_CMP_DISBALE` (0)
- #define `ADC_INTEN_CMP_OUTSIDETH` (1)
- #define `ADC_INTEN_CMP_CROSSSTH` (2)
- #define `ADC_INTEN_CMP_MASK` (3)
- #define `ADC_INTEN_CMP_ENABLE`(isel, ch) (((isel) & `ADC_INTEN_CMP_MASK`) << ((2 \* (ch)) + 3))
- #define `ADC_FLAGS_THCMP_MASK`(ch) (1 << (ch))
- #define `ADC_FLAGS_OVRRUN_MASK`(ch) (1 << (12 + (ch)))
- #define `ADC_FLAGS_SEQA_OVRRUN_MASK` (1 << 24)
- #define `ADC_FLAGS_SEQB_OVRRUN_MASK` (1 << 25)
- #define `ADC_FLAGS_SEQN_OVRRUN_MASK`(seq) (1 << (24 + (seq)))
- #define `ADC_FLAGS_SEQA_INT_MASK` (1 << 28)
- #define `ADC_FLAGS_SEQB_INT_MASK` (1 << 29)
- #define `ADC_FLAGS_SEQN_INT_MASK`(seq) (1 << (28 + (seq)))
- #define `ADC_FLAGS_THCMP_INT_MASK` (1 << 30)
- #define `ADC_FLAGS_OVRRUN_INT_MASK` (1UL << 31)
- #define `ADC_STARTUP_ENABLE` (0x1 << 0)
- #define `ADC_STARTUP_INIT` (0x1 << 1)
- #define `ADC_CALIB` (0x1 << 0)
- #define `ADC_CALREQD` (0x1 << 1)

## Enumerations

- enum `ADC_SEQ_IDX_T` { `ADC_SEQA_IDX` = 0, `ADC_SEQB_IDX` }
- enum `ADC_TSAMP_T` {  
`ADC_TSAMP_2CLK5` = 0, `ADC_TSAMP_3CLK5`, `ADC_TSAMP_4CLK5`, `ADC_TSAMP_5CLK5`,  
`ADC_TSAMP_6CLK5`, `ADC_TSAMP_7CLK5`, `ADC_TSAMP_8CLK5`, `ADC_TSAMP_9CLK5` }  
*ADC sampling time bits 12, 13 and 14.*
- enum `ADC_DR_THCMPRANGE_T` { `ADC_DR_THCMPRANGE_INRANGE`, `ADC_DR_THCMPRANGE_↵RESERVED`, `ADC_DR_THCMPRANGE_BELOW`, `ADC_DR_THCMPRANGE_ABOVE` }
- enum `ADC_DR_THCMPCROSS_T` { `ADC_DR_THCMPCROSS_NOCROSS`, `ADC_DR_THCMPCROSS_↵RESERVED`, `ADC_DR_THCMPCROSS_DOWNWARD`, `ADC_DR_THCMPCROSS_UPWARD` }
- enum `ADC_INTEN_THCMP_T` { `ADC_INTEN_THCMP_DISABLE`, `ADC_INTEN_THCMP_OUTSIDE`, `AD↵C_INTEN_THCMP_CROSSING` }

## Functions

- void [Chip\\_ADC\\_Init](#) ([LPC\\_ADC\\_T](#) \*pADC, uint32\_t flags)  
*Initialize the ADC peripheral.*
- void [Chip\\_ADC\\_DeInit](#) ([LPC\\_ADC\\_T](#) \*pADC)  
*Shutdown ADC.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_SetDivider](#) ([LPC\\_ADC\\_T](#) \*pADC, uint8\_t div)  
*Set ADC divider.*
- void [Chip\\_ADC\\_SetClockRate](#) ([LPC\\_ADC\\_T](#) \*pADC, uint32\_t rate)  
*Set ADC clock rate.*
- [\\_\\_STATIC\\_INLINE](#) uint8\_t [Chip\\_ADC\\_GetDivider](#) ([LPC\\_ADC\\_T](#) \*pADC)  
*Get ADC divider.*
- uint32\_t [Chip\\_ADC\\_Calibration](#) ([LPC\\_ADC\\_T](#) \*pADC)  
*Perform ADC calibration.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_SelectTempSensorInput](#) ([LPC\\_ADC\\_T](#) \*pADC)  
*Selects Temperature sensor as the input for Channel 0.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_SetSequencerBits](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex, uint32\_t bits)  
*Helper function for safely setting ADC sequencer register bits.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_ClearSequencerBits](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex, uint32\_t bits)  
*Helper function for safely clearing ADC sequencer register bits.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_SetupSequencer](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex, uint32\_t options)  
*Sets up ADC conversion sequencer A or B.*
- [\\_\\_STATIC\\_INLINE](#) uint32\_t [Chip\\_ADC\\_GetSequencerCtrl](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex)  
*Get sequenceX control register value.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_EnableSequencer](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex)  
*Enables a sequencer.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_DisableSequencer](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex)  
*Disables a sequencer.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_StartSequencer](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex)  
*Forces a sequencer trigger event (software trigger of ADC)*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_StartBurstSequencer](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex)  
*Starts sequencer burst mode.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_StopBurstSequencer](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex)  
*Stops sequencer burst mode.*
- [\\_\\_STATIC\\_INLINE](#) uint32\_t [Chip\\_ADC\\_GetGlobalDataReg](#) ([LPC\\_ADC\\_T](#) \*pADC, [ADC\\_SEQ\\_IDX\\_T](#) seqIndex)  
*Read a ADC sequence global data register.*
- [\\_\\_STATIC\\_INLINE](#) uint32\_t [Chip\\_ADC\\_GetDataReg](#) ([LPC\\_ADC\\_T](#) \*pADC, uint8\_t index)  
*Read a ADC data register.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_SetThrLowValue](#) ([LPC\\_ADC\\_T](#) \*pADC, uint8\_t thrnum, uint16\_t value)  
*Set Threshold low value in ADC.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_SetThrHighValue](#) ([LPC\\_ADC\\_T](#) \*pADC, uint8\_t thrnum, uint16\_t value)  
*Set Threshold high value in ADC.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_SelectTH0Channels](#) ([LPC\\_ADC\\_T](#) \*pADC, uint32\_t channels)  
*Select threshold 0 values for comparison for selected channels.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_ADC\\_SelectTH1Channels](#) ([LPC\\_ADC\\_T](#) \*pADC, uint32\_t channels)

- Select threshold 1 value for comparison for selected channels.*

  - `__STATIC_INLINE void Chip_ADC_EnableInt (LPC_ADC_T *pADC, uint32_t intMask)`  
*Enable interrupts in ADC (sequencers A/B and overrun)*
  - `__STATIC_INLINE void Chip_ADC_DisableInt (LPC_ADC_T *pADC, uint32_t intMask)`  
*Disable interrupts in ADC (sequencers A/B and overrun)*
  - `__STATIC_INLINE void Chip_ADC_SetThresholdInt (LPC_ADC_T *pADC, uint8_t ch, ADC_INTEN_THCMP_T thInt)`  
*Enable a threshold event interrupt in ADC.*
  - `__STATIC_INLINE uint32_t Chip_ADC_GetFlags (LPC_ADC_T *pADC)`  
*Get flags register in ADC.*
  - `__STATIC_INLINE void Chip_ADC_ClearFlags (LPC_ADC_T *pADC, uint32_t flags)`  
*Clear flags register in ADC.*
  - `__STATIC_INLINE void Chip_ADC_SetTHRSELBits (LPC_ADC_T *pADC, uint32_t mask)`  
*Set Threshold selection bits.*
  - `__STATIC_INLINE void Chip_ADC_ClearTHRSELBits (LPC_ADC_T *pADC, uint32_t mask)`  
*Clear Threshold selection bits.*

## 6.4.2 Macro Definition Documentation

### 6.4.2.1 `#define ADC_CALIB (0x1 << 0)`

Definition at line 195 of file `adc_5411x.h`.

### 6.4.2.2 `#define ADC_CALREQD (0x1 << 1)`

Definition at line 196 of file `adc_5411x.h`.

### 6.4.2.3 `#define ADC_CR_ASYNC_MODE (1 << 8)`

Asynchronous mode enable bit

Definition at line 96 of file `adc_5411x.h`.

### 6.4.2.4 `#define ADC_CR_BITACC( n ) (((n) & 0x1) << 9)`

12-bit or 10-bit ADC accuracy

Definition at line 102 of file `adc_5411x.h`.

### 6.4.2.5 `#define ADC_CR_BYPASS (1 << 11)`

Bypass mode

Definition at line 99 of file `adc_5411x.h`.

### 6.4.2.6 `#define ADC_CR_CALMODEBIT (1 << 30)`

Self calibration cycle enable bit

Definition at line 101 of file `adc_5411x.h`.

**6.4.2.7 #define ADC\_CR\_CLKDIV( *n* ) (((*n*) & 0xFF) << 0)**

The APB clock (PCLK) is divided by (this value plus one) to produce the clock for the A/D

Definition at line 103 of file adc\_5411x.h.

**6.4.2.8 #define ADC\_CR\_CLKDIV\_BITPOS (0)**

Bit position for Clock divider value

Definition at line 95 of file adc\_5411x.h.

**6.4.2.9 #define ADC\_CR\_CLKDIV\_MASK (0xFF << 0)**

ADC register support bitfields and mask.

ADC Control register bit fields Mask for Clock divider value

Definition at line 94 of file adc\_5411x.h.

**6.4.2.10 #define ADC\_CR\_LPWRMODEBIT (1 << 10)**

Low power mode enable bit

Definition at line 98 of file adc\_5411x.h.

**6.4.2.11 #define ADC\_CR\_RESOL( *n* ) ((*n*) << 9)**

2-bits, 6(0x0),8(0x1),10(0x2),12(0x3)-bit mode enable bit

Definition at line 97 of file adc\_5411x.h.

**6.4.2.12 #define ADC\_CR\_TSAMP( *n* ) ((*n*) << 12)**

3-bits, 2.5(0x0),3.5(0x1),4.5(0x2),5.5(0x3),6.5(0x4),7.5(0x5),8.5(0x6),9.5(0x7) ADC clocks sampling time

Definition at line 100 of file adc\_5411x.h.

**6.4.2.13 #define ADC\_DR\_CHAN\_BITPOS (26)**

Channel number bit position

Definition at line 154 of file adc\_5411x.h.

**6.4.2.14 #define ADC\_DR\_CHAN\_MASK (0xF << 26)**

Channel number mask

Definition at line 153 of file adc\_5411x.h.

**6.4.2.15 #define ADC\_DR\_CHANNEL( *n* ) (((*n*) >> ADC\_DR\_CHAN\_BITPOS) & 0xF)**

Channel number bit position

Definition at line 155 of file adc\_5411x.h.

**6.4.2.16 #define ADC\_DR\_DATAVALID (1UL << 31)**

Data valid bit

Definition at line 157 of file adc\_5411x.h.

**6.4.2.17 #define ADC\_DR\_DONE( n ) (((n) >> 31))**

Definition at line 158 of file adc\_5411x.h.

**6.4.2.18 #define ADC\_DR\_OVERRUN (1 << 30)**

Overrun bit

Definition at line 156 of file adc\_5411x.h.

**6.4.2.19 #define ADC\_DR\_RESULT( n ) (((n) >> 4) & 0xFFF)**

Macro for getting the ADC data value

Definition at line 146 of file adc\_5411x.h.

**6.4.2.20 #define ADC\_DR\_RESULT\_BITPOS (4)**

ADC Data register bit fields Result start bit position

Definition at line 145 of file adc\_5411x.h.

**6.4.2.21 #define ADC\_DR\_THCMPCROSS( n ) (((n) >> ADC\_DR\_THCMPCROSS\_BITPOS) & 0x3)**

Definition at line 152 of file adc\_5411x.h.

**6.4.2.22 #define ADC\_DR\_THCMPCROSS\_BITPOS (18)**

Comparison cross bit position

Definition at line 151 of file adc\_5411x.h.

**6.4.2.23 #define ADC\_DR\_THCMPCROSS\_MASK (0x3 << 18)**

Comparison cross mask

Definition at line 150 of file adc\_5411x.h.

**6.4.2.24 #define ADC\_DR\_THCMPRANGE( n ) (((n) >> ADC\_DR\_THCMPRANGE\_BITPOS) & 0x3)**

Definition at line 149 of file adc\_5411x.h.

**6.4.2.25 #define ADC\_DR\_THCMPRANGE\_BITPOS (16)**

Comparison range bit position

Definition at line 148 of file adc\_5411x.h.

**6.4.2.26 #define ADC\_DR\_THCMPRANGE\_MASK (0x3 << 16)**

Comparison range mask

Definition at line 147 of file adc\_5411x.h.

**6.4.2.27 #define ADC\_FLAGS\_OVRRUN\_INT\_MASK (1UL << 31)**

Overrun Interrupt status

Definition at line 188 of file adc\_5411x.h.

**6.4.2.28 #define ADC\_FLAGS\_OVRRUN\_MASK( ch )(1 << (12 + (ch)))**

Overrun status for channel

Definition at line 180 of file adc\_5411x.h.

**6.4.2.29 #define ADC\_FLAGS\_SEQA\_INT\_MASK (1 << 28)**

Seq A Interrupt status

Definition at line 184 of file adc\_5411x.h.

**6.4.2.30 #define ADC\_FLAGS\_SEQA\_OVRRUN\_MASK (1 << 24)**

Seq A Overrun status

Definition at line 181 of file adc\_5411x.h.

**6.4.2.31 #define ADC\_FLAGS\_SEQB\_INT\_MASK (1 << 29)**

Seq B Interrupt status

Definition at line 185 of file adc\_5411x.h.

**6.4.2.32 #define ADC\_FLAGS\_SEQB\_OVRRUN\_MASK (1 << 25)**

Seq B Overrun status

Definition at line 182 of file adc\_5411x.h.

**6.4.2.33 #define ADC\_FLAGS\_SEQN\_INT\_MASK( seq )(1 << (28 + (seq)))**

Seq A/B Interrupt status

Definition at line 186 of file adc\_5411x.h.

**6.4.2.34 #define ADC\_FLAGS\_SEQN\_OVRRUN\_MASK( seq )(1 << (24 + (seq)))**

Seq A/B Overrun status

Definition at line 183 of file adc\_5411x.h.



**6.4.2.35 #define ADC\_FLAGS\_THCMP\_INT\_MASK (1 << 30)**

Threshold comparison Interrupt status

Definition at line 187 of file adc\_5411x.h.

**6.4.2.36 #define ADC\_FLAGS\_THCMP\_MASK( ch ) (1 << (ch))**

ADC Flags register bit fields Threshold comparison status for channel

Definition at line 179 of file adc\_5411x.h.

**6.4.2.37 #define ADC\_INTEN\_CMP\_CROSSTH (2)**

Crossing threshold interrupt value

Definition at line 174 of file adc\_5411x.h.

**6.4.2.38 #define ADC\_INTEN\_CMP\_DISBALE (0)**

Disable comparison interrupt value

Definition at line 172 of file adc\_5411x.h.

**6.4.2.39 #define ADC\_INTEN\_CMP\_ENABLE( isel, ch ) (((isel) & ADC\_INTEN\_CMP\_MASK) << ((2 \* (ch)) + 3))**

Interrupt selection for channel

Definition at line 176 of file adc\_5411x.h.

**6.4.2.40 #define ADC\_INTEN\_CMP\_MASK (3)**

Comparison interrupt value mask

Definition at line 175 of file adc\_5411x.h.

**6.4.2.41 #define ADC\_INTEN\_CMP\_OUTSIDETH (1)**

Outside threshold interrupt value

Definition at line 173 of file adc\_5411x.h.

**6.4.2.42 #define ADC\_INTEN\_OVRRUN\_ENABLE (1 << 2)**

Overrun Interrupt enable bit

Definition at line 171 of file adc\_5411x.h.

**6.4.2.43 #define ADC\_INTEN\_SEQA\_ENABLE (1 << 0)**

ADC Interrupt Enable register bit fields Sequence A Interrupt enable bit

Definition at line 168 of file adc\_5411x.h.

**6.4.2.44 #define ADC\_INTEN\_SEQB\_ENABLE (1 << 1)**

Sequence B Interrupt enable bit

Definition at line 169 of file adc\_5411x.h.

**6.4.2.45 #define ADC\_INTEN\_SEQN\_ENABLE( seq ) (1 << (seq))**

Sequence A/B Interrupt enable bit

Definition at line 170 of file adc\_5411x.h.

**6.4.2.46 #define ADC\_MAX\_CHANNEL\_NUM 12**

Definition at line 88 of file adc\_5411x.h.

**6.4.2.47 #define ADC\_MAX\_SAMPLE\_RATE 8000000**

Maximum sample rate in Hz (12-bit conversions)

Definition at line 87 of file adc\_5411x.h.

**6.4.2.48 #define ADC\_SAMPLE\_RATE\_CONFIG\_MASK (ADC\_CR\_CLKDIV(0xFF) | ADC\_CR\_BITACC(0x01))**

Definition at line 104 of file adc\_5411x.h.

**6.4.2.49 #define ADC\_SEQ\_CTRL\_BURST (1 << 27)**

Repeated conversion enable bit

Definition at line 126 of file adc\_5411x.h.

**6.4.2.50 #define ADC\_SEQ\_CTRL\_CHANNEL\_EN( n ) (1 << n)**

SEQ\_CTRL register bit fields

Definition at line 121 of file adc\_5411x.h.

**6.4.2.51 #define ADC\_SEQ\_CTRL\_HWTRIG\_POLPOS (1 << 18)**

HW trigger polarity - positive edge

Definition at line 123 of file adc\_5411x.h.

**6.4.2.52 #define ADC\_SEQ\_CTRL\_HWTRIG\_SYNCBYPASS (1 << 19)**

HW trigger bypass synchronisation

Definition at line 124 of file adc\_5411x.h.

**6.4.2.53 #define ADC\_SEQ\_CTRL\_LOWPRIO (1 << 29)**

High priority enable bit (regardless of name)

Definition at line 128 of file adc\_5411x.h.

**6.4.2.54 #define ADC\_SEQ\_CTRL\_MODE\_EOS (1 << 30)**

Mode End of sequence enable bit

Definition at line 129 of file adc\_5411x.h.

**6.4.2.55 #define ADC\_SEQ\_CTRL\_SEQ\_ENA (1UL << 31)**

Sequence enable bit

Definition at line 130 of file adc\_5411x.h.

**6.4.2.56 #define ADC\_SEQ\_CTRL\_SINGLESTEP (1 << 28)**

Single step enable bit

Definition at line 127 of file adc\_5411x.h.

**6.4.2.57 #define ADC\_SEQ\_CTRL\_START (1 << 26)**

Start conversion enable bit

Definition at line 125 of file adc\_5411x.h.

**6.4.2.58 #define ADC\_SEQ\_CTRL\_TRIGGER( n ) ((n & 0x3f)<<12)**

Definition at line 122 of file adc\_5411x.h.

**6.4.2.59 #define ADC\_SEQ\_GDAT\_CHAN\_BITPOS (26)**

Channel number bit position

Definition at line 140 of file adc\_5411x.h.

**6.4.2.60 #define ADC\_SEQ\_GDAT\_CHAN\_MASK (0xF << 26)**

Channel number mask

Definition at line 139 of file adc\_5411x.h.

**6.4.2.61 #define ADC\_SEQ\_GDAT\_DATAVALID (1UL << 31)**

Data valid bit

Definition at line 142 of file adc\_5411x.h.

**6.4.2.62 #define ADC\_SEQ\_GDAT\_OVERRUN (1 << 30)**

Overrun bit

Definition at line 141 of file adc\_5411x.h.

**6.4.2.63 #define ADC\_SEQ\_GDAT\_RESULT\_BITPOS (4)**

Result start bit position

Definition at line 134 of file adc\_5411x.h.

**6.4.2.64 #define ADC\_SEQ\_GDAT\_RESULT\_MASK (0xFF << 4)**

ADC global data register bit fields Result value mask

Definition at line 133 of file adc\_5411x.h.

**6.4.2.65 #define ADC\_SEQ\_GDAT\_THCMPCROSS\_BITPOS (18)**

Comparison cross bit position

Definition at line 138 of file adc\_5411x.h.

**6.4.2.66 #define ADC\_SEQ\_GDAT\_THCMPCROSS\_MASK (0x3 << 18)**

Comparison cross mask

Definition at line 137 of file adc\_5411x.h.

**6.4.2.67 #define ADC\_SEQ\_GDAT\_THCMPRANGE\_BITPOS (16)**

Comparison range bit position

Definition at line 136 of file adc\_5411x.h.

**6.4.2.68 #define ADC\_SEQ\_GDAT\_THCMPRANGE\_MASK (0x3 << 16)**

Comparison range mask

Definition at line 135 of file adc\_5411x.h.

**6.4.2.69 #define ADC\_STARTUP\_ENABLE (0x1 << 0)**

ADC Startup register bit fields

Definition at line 191 of file adc\_5411x.h.

**6.4.2.70 #define ADC\_STARTUP\_INIT (0x1 << 1)**

Definition at line 192 of file adc\_5411x.h.

**6.4.2.71 #define ADC\_THR\_VAL\_MASK (0xFF << 4)**

ADC low/high Threshold register bit fields Threshold value bit mask

Definition at line 161 of file adc\_5411x.h.

**6.4.2.72 #define ADC\_THR\_VAL\_POS (4)**

Threshold value bit position

Definition at line 162 of file adc\_5411x.h.

**6.4.2.73 #define ADC\_THRSEL\_CHAN\_SEL\_THR1( n )(1 << (n))**

ADC Threshold select register bit fields Select THR1 register for channel n

Definition at line 165 of file adc\_5411x.h.

**6.4.3 Enumeration Type Documentation****6.4.3.1 enum ADC\_DR\_THCMPCROSS\_T**

ADC sequence global data register threshold comparison cross enumerations

Enumerator

***ADC\_DR\_THCMPCROSS\_NOCROSS***  
***ADC\_DR\_THCMPCROSS\_RESERVED***  
***ADC\_DR\_THCMPCROSS\_DOWNWARD***  
***ADC\_DR\_THCMPCROSS\_UPWARD***

Definition at line 425 of file adc\_5411x.h.

**6.4.3.2 enum ADC\_DR\_THCMPRANGE\_T**

ADC sequence global data register threshold comparison range enumerations

Enumerator

***ADC\_DR\_THCMPRANGE\_INRANGE***  
***ADC\_DR\_THCMPRANGE\_RESERVED***  
***ADC\_DR\_THCMPRANGE\_BELOW***  
***ADC\_DR\_THCMPRANGE\_ABOVE***

Definition at line 417 of file adc\_5411x.h.

**6.4.3.3 enum ADC\_INTEN\_THCMP\_T**

Threshold interrupt event options

Enumerator

***ADC\_INTEN\_THCMP\_DISABLE***  
***ADC\_INTEN\_THCMP\_OUTSIDE***  
***ADC\_INTEN\_THCMP\_CROSSING***

Definition at line 559 of file adc\_5411x.h.

#### 6.4.3.4 enum ADC\_SEQ\_IDX\_T

Sequence index enumerations, used in various parts of the code for register indexing and sequencer selection

Enumerator

***ADC\_SEQA\_IDX***

***ADC\_SEQB\_IDX***

Definition at line 46 of file adc\_5411x.h.

#### 6.4.3.5 enum ADC\_TSAMP\_T

ADC sampling time bits 12, 13 and 14.

Enumerator

***ADC\_TSAMP\_2CLK5***

***ADC\_TSAMP\_3CLK5***

***ADC\_TSAMP\_4CLK5***

***ADC\_TSAMP\_5CLK5***

***ADC\_TSAMP\_6CLK5***

***ADC\_TSAMP\_7CLK5***

***ADC\_TSAMP\_8CLK5***

***ADC\_TSAMP\_9CLK5***

Definition at line 109 of file adc\_5411x.h.

### 6.4.4 Function Documentation

#### 6.4.4.1 uint32\_t Chip\_ADC\_Calibration ( LPC\_ADC\_T \* *pADC* )

Perform ADC calibration.

Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
-------------	--

Returns

LPC\_OK on success, ERR\_TIME\_OUT or ERR\_ADC\_NO\_POWER on failure

Note

Calibration is not done as part of [Chip\\_ADC\\_Init\(\)](#), but is required after the call to [Chip\\_ADC\\_Init\(\)](#) or after returning from a power-down state.

Definition at line 76 of file adc\_5411x.c.

#### 6.4.4.2 \_\_STATIC\_INLINE void Chip\_ADC\_ClearFlags ( LPC\_ADC\_T \* *pADC*, uint32\_t *flags* )

Clear flags register in ADC.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>flags</i>	: An Or'ed values of ADC_FLAGS_* values to clear

## Returns

Flags register value (ORed ADC\_FLAG\* values)

Definition at line 603 of file adc\_5411x.h.

**6.4.4.3** `__STATIC_INLINE void Chip_ADC_ClearSequencerBits ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex, uint32_t bits )`

Helper function for safely clearing ADC sequencer register bits.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to clear bits for
<i>bits</i>	: Or'ed bits of a sequencer register to clear

## Returns

Nothing

## Note

This function will safely clear the ADC sequencer register bits while maintaining bits 20..25 as 0, regardless of the read state of those bits.

Definition at line 306 of file adc\_5411x.h.

**6.4.4.4** `__STATIC_INLINE void Chip_ADC_ClearTHRSELBits ( LPC_ADC_T * pADC, uint32_t mask )`

Clear Threshold selection bits.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>mask</i>	: Threshold selection mask

## Returns

Nothing

Definition at line 624 of file adc\_5411x.h.

**6.4.4.5** `void Chip_ADC_DeInit ( LPC_ADC_T * pADC )`

Shutdown ADC.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
-------------	--

**Returns**

Nothing

**Note**

Disables the ADC clocks and ADC power

Definition at line 65 of file adc\_5411x.c.

#### 6.4.4.6 `__STATIC_INLINE void Chip_ADC_DisableInt ( LPC_ADC_T * pADC, uint32_t intMask )`

Disable interrupts in ADC (sequencers A/B and overrun)

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>intMask</i>	: Interrupt values to be disabled (see notes)

**Returns**

None

**Note**

Select one or more OR'ed values of ADC\_INTEN\_SEQA\_ENABLE, ADC\_INTEN\_SEQB\_ENABLE, and ADC\_INTEN\_OVRUN\_ENABLE to disable the specific ADC interrupts.

Definition at line 552 of file adc\_5411x.h.

#### 6.4.4.7 `__STATIC_INLINE void Chip_ADC_DisableSequencer ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex )`

Disables a sequencer.

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to disable

**Returns**

Nothing

Definition at line 368 of file adc\_5411x.h.

#### 6.4.4.8 `__STATIC_INLINE void Chip_ADC_EnableInt ( LPC_ADC_T * pADC, uint32_t intMask )`

Enable interrupts in ADC (sequencers A/B and overrun)

**Parameters**


---



<i>pADC</i>	: The base of ADC peripheral on the chip
<i>intMask</i>	: Interrupt values to be enabled (see notes)

**Returns**

None

**Note**

Select one or more OR'ed values of ADC\_INTEN\_SEQA\_ENABLE, ADC\_INTEN\_SEQB\_ENABLE, and ADC\_INTEN\_OVRRUN\_ENABLE to enable the specific ADC interrupts.

Definition at line 537 of file adc\_5411x.h.

#### 6.4.4.9 `__STATIC_INLINE void Chip_ADC_EnableSequencer ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex )`

Enables a sequencer.

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to enable

**Returns**

Nothing

Definition at line 357 of file adc\_5411x.h.

#### 6.4.4.10 `__STATIC_INLINE uint32_t Chip_ADC_GetDataReg ( LPC_ADC_T * pADC, uint8_t index )`

Read a ADC data register.

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>index</i>	: Data register to read, 0-11

**Returns**

Current raw value of the ADC data register

**Note**

This function returns the raw value of the data register and clears the overrun and datavalid status for the register. Once this register is read, the following functions can be used to parse the raw value:

```
uint32_t adcDataRawValue = Chip_ADC_GetDataReg(LPC_ADC, ADC_MAX_CHANNEL_NUM); // Get raw
value uint32_t adcDataValue = ADC_DR_RESULT(adcDataRawValue); // Aligned and masked ADC data
value ADC_DR_THCMPRANGE_T adcRange = (ADC_DR_THCMPRANGE_T) ADC_DR_THCMPRANG<
E(adcDataRawValue); // Sample range compared to threshold low/high ADC_DR_THCMPCROSS_T adc<
Range = (ADC_DR_THCMPCROSS_T) ADC_DR_THCMPCROSS(adcDataRawValue); // Sample cross
compared to threshold low uint32_t channel = ADC_DR_CHANNEL(adcDataRawValue); // ADC channel for
this sample/data bool adcDataOverrun = (bool) ((adcDataRawValue & ADC_DR_OVERRUN) != 0); // Data
overrun flag bool adcDataValid = (bool) ((adcDataRawValue & ADC_SEQ_GDAT_DATAVALID) != 0); // Data
valid flag
```

Definition at line 469 of file adc\_5411x.h.

6.4.4.11 `__STATIC_INLINE uint8_t Chip_ADC_GetDivider ( LPC_ADC_T * pADC )`

Get ADC divider.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
-------------	--

## Returns

the current ADC divider

## Note

This function returns the divider that is used to generate the ADC frequency. The returned value must be incremented by 1. The frequency can be determined with the following function:

```
adc_freq = Chip_Clock_GetSystemClockRate() / (Chip_ADC_GetDivider(LPC_ADC) + 1);
```

Definition at line 255 of file adc\_5411x.h.

#### 6.4.4.12 `__STATIC_INLINE uint32_t Chip_ADC_GetFlags ( LPC_ADC_T * pADC )`

Get flags register in ADC.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
-------------	--

## Returns

Flags register value (ORed ADC\_FLAG\* values)

## Note

Mask the return value of this function with the ADC\_FLAGS\_\* definitions to determine the overall ADC interrupt events.

Example:

```
if (Chip_ADC_GetFlags(LPC_ADC) & ADC_FLAGS_THCMP_MASK(3) // Check of threshold comp status for ADC channel 3
```

Definition at line 592 of file adc\_5411x.h.

#### 6.4.4.13 `__STATIC_INLINE uint32_t Chip_ADC_GetGlobalDataReg ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex )`

Read a ADC sequence global data register.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to read

## Returns

Current raw value of the ADC sequence A or B global data register

**Note**

This function returns the raw value of the data register and clears the overrun and datavalid status for the register. Once this register is read, the following functions can be used to parse the raw value:

```
uint32_t adcDataRawValue = Chip_ADC_GetGlobalDataReg(LPC_ADC, ADC_SEQA_IDX); // Get raw value
uint32_t adcDataValue = ADC_DR_RESULT(adcDataRawValue); // Aligned and masked ADC data value
ADC_DR_THCMPRANGE_T adcRange = (ADC_DR_THCMPRANGE_T) ADC_DR_THCMPRANGE(adcDataRawValue); // Sample range compared to threshold low/high
ADC_DR_THMPCROSS_T adcRange = (ADC_DR_THMPCROSS_T) ADC_DR_THMPCROSS(adcDataRawValue); // Sample cross compared to threshold low
uint32_t channel = ADC_DR_CHANNEL(adcDataRawValue); // ADC channel for this sample/data
bool adcDataOverrun = (bool) ((adcDataRawValue & ADC_DR_OVERRUN) != 0); // Data overrun flag
bool adcDataValid = (bool) ((adcDataRawValue & ADC_SEQ_GDAT_DATAVALID) != 0); // Data valid flag
```

Definition at line 448 of file adc\_5411x.h.

#### 6.4.4.14 `__STATIC_INLINE uint32_t Chip_ADC_GetSequencerCtrl ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex )`

Get sequenceX control register value.

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to setup

**Returns**

Sequencer control register value

Definition at line 346 of file adc\_5411x.h.

#### 6.4.4.15 `void Chip_ADC_Init ( LPC_ADC_T * pADC, uint32_t flags )`

Initialize the ADC peripheral.

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>flags</i>	: ADC flags for init (ADC_CR_MODE10BIT and/or ADC_CR_LPWRMODEBIT)

**Returns**

Nothing

**Note**

To select low-power ADC mode, enable the ADC\_CR\_LPWRMODEBIT flag. To select 10-bit conversion mode, enable the ADC\_CR\_MODE10BIT flag.

Example: `Chip_ADC_Init(LPC_ADC, (ADC_CR_MODE10BIT | ADC_CR_LPWRMODEBIT));`

Definition at line 51 of file adc\_5411x.c.

#### 6.4.4.16 `__STATIC_INLINE void Chip_ADC_SelectTempSensorInput ( LPC_ADC_T * pADC )`

Selects Temperature sensor as the input for Channel 0.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
-------------	--

## Returns

Nothing

Definition at line 275 of file adc\_5411x.h.

6.4.4.17 `__STATIC_INLINE void Chip_ADC_SelectTH0Channels ( LPC_ADC_T * pADC, uint32_t channels )`

Select threshold 0 values for comparison for selected channels.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>channels</i>	: An OR'ed value of one or more <a href="#">ADC_THRSEL_CHAN_SEL_THR1(ch)</a> values

## Returns

None

## Note

Select multiple channels to use the threshold 0 comparison.

Example:

```
Chip_ADC_SelectTH0Channels(LPC_ADC, (ADC_THRSEL_CHAN_SEL_THR1(1) | ADC_THRSEL_CHAN_SEL_THR1(2))); // Selects channels 1 and 2 for threshold 0
```

Definition at line 507 of file adc\_5411x.h.

6.4.4.18 `__STATIC_INLINE void Chip_ADC_SelectTH1Channels ( LPC_ADC_T * pADC, uint32_t channels )`

Select threshold 1 value for comparison for selected channels.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>channels</i>	: An OR'ed value of one or more <a href="#">ADC_THRSEL_CHAN_SEL_THR1(ch)</a> values

## Returns

None

## Note

Select multiple channels to use the 1 threshold comparison.

Example:

```
Chip_ADC_SelectTH1Channels(LPC_ADC, (ADC_THRSEL_CHAN_SEL_THR1(4) | ADC_THRSEL_CHAN_SEL_THR1(5))); // Selects channels 4 and 5 for 1 threshold
```

Definition at line 522 of file adc\_5411x.h.

6.4.4.19 `void Chip_ADC_SetClockRate ( LPC_ADC_T * pADC, uint32_t rate )`

Set ADC clock rate.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>rate</i>	: rate in Hz to set ADC clock to (maximum ADC_MAX_SAMPLE_RATE)

## Returns

Nothing

Definition at line 116 of file adc\_5411x.c.

#### 6.4.4.20 `__STATIC_INLINE void Chip_ADC_SetDivider ( LPC_ADC_T * pADC, uint8_t div )`

Set ADC divider.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>div</i>	: ADC divider value to set minus 1

## Returns

Nothing

## Note

The value is used as a divider to generate the ADC clock rate from the ADC input clock. The ADC input clock is based on the system clock. Valid values for this function are from 0 to 255 with 0=divide by 1, 1=divide by 2, 2=divide by 3, etc.

Do not decrement this value by 1.

To set the ADC clock rate to 1MHz, use the following function:

`Chip_ADC_SetDivider(LPC_ADC, (Chip_Clock_GetSystemClockRate() / 1000000) - 1);`

Definition at line 230 of file adc\_5411x.h.

#### 6.4.4.21 `__STATIC_INLINE void Chip_ADC_SetSequencerBits ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex, uint32_t bits )`

Helper function for safely setting ADC sequencer register bits.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to set bits for
<i>bits</i>	: Or'ed bits of a sequencer register to set

## Returns

Nothing

## Note

This function will safely set the ADC sequencer register bits while maintaining bits 20..25 as 0, regardless of the read state of those bits.

Definition at line 291 of file adc\_5411x.h.

6.4.4.22 `__STATIC_INLINE void Chip_ADC_SetThresholdInt ( LPC_ADC_T * pADC, uint8_t ch, ADC_INTEN_THCMP_T thInt )`

Enable a threshold event interrupt in ADC.

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>ch</i>	: ADC channel to set threshold inetrrupt for, 1-8
<i>thInt</i>	: Selected threshold interrupt type

**Returns**

None

Definition at line 572 of file adc\_5411x.h.

6.4.4.23 `__STATIC_INLINE void Chip_ADC_SetThrHighValue ( LPC_ADC_T * pADC, uint8_t thrnum, uint16_t value )`

Set Threshold high value in ADC.

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>thrnum</i>	: Threshold register value (1 for threshold register 1, 0 for threshold register 0)
<i>value</i>	: Threshold high data value (should be 12-bit value)

**Returns**

None

Definition at line 493 of file adc\_5411x.h.

6.4.4.24 `__STATIC_INLINE void Chip_ADC_SetThrLowValue ( LPC_ADC_T * pADC, uint8_t thrnum, uint16_t value )`

Set Threshold low value in ADC.

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>thrnum</i>	: Threshold register value (1 for threshold register 1, 0 for threshold register 0)
<i>value</i>	: Threshold low data value (should be 12-bit value)

**Returns**

None

Definition at line 481 of file adc\_5411x.h.

6.4.4.25 `__STATIC_INLINE void Chip_ADC_SetTHRSELBits ( LPC_ADC_T * pADC, uint32_t mask )`

Set Threshold selection bits.

**Parameters**

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>mask</i>	: Threshold selection mask

**Returns**

Nothing

Definition at line 614 of file adc\_5411x.h.



6.4.4.26 `__STATIC_INLINE void Chip_ADC_SetupSequencer ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex,  
uint32_t options )`

Sets up ADC conversion sequencer A or B.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to setup
<i>options</i>	: OR'ed Sequencer options to setup (see notes)

## Returns

Nothing

## Note

Sets up sequencer options for a conversion sequence. This function should be used to setup the selected channels for the sequence, the sequencer trigger, the trigger polarity, synchronization bypass, priority, and mode. All options are passed to the functions as a OR'ed list of values. This function will disable/clear the sequencer start/burst/single step/enable if they are enabled.

Select the channels by OR'ing in one or more ADC\_SEQ\_CTRL\_CHANSEL(ch) values.

Select the hardware trigger by OR'ing in one ADC\_SEQ\_CTRL\_HWTRIG\_\* value.

Select a positive edge hardware trigger by OR'ing in ADC\_SEQ\_CTRL\_HWTRIG\_POLPOS.

Select trigger bypass synchronisation by OR'ing in ADC\_SEQ\_CTRL\_HWTRIG\_SYNCBYPASS.

Select ADC single step on trigger/start by OR'ing in ADC\_SEQ\_CTRL\_SINGLESTEP.

Select higher priority conversion on the other sequencer by OR'ing in ADC\_SEQ\_CTRL\_LOWPRIO.

Select end of sequence instead of end of conversion interrupt by OR'ing in ADC\_SEQ\_CTRL\_MODE\_EOS.

Example for setting up sequencer A (channels 0-2, trigger on high edge of PIO0\_2, interrupt on end of sequence):

```
Chip_ADC_SetupSequencer(LPC_ADC, ADC_SEQA_IDX, ( ADC_SEQ_CTRL_CHANSEL(0) | ADC_SEQ_
_CTRL_CHANSEL(1) | ADC_SEQ_CTRL_CHANSEL(2) | ADC_SEQ_CTRL_HWTRIG_PIO0_2 | ADC_SE
Q_CTRL_HWTRIG_POLPOS | ADC_SEQ_CTRL_MODE_EOS));
```

Definition at line 335 of file adc\_5411x.h.

#### 6.4.4.27 `__STATIC_INLINE void Chip_ADC_StartBurstSequencer ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex )`

Starts sequencer burst mode.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to start burst on

## Returns

Nothing

## Note

This function sets the BURST bit for the sequencer to force continuous conversion. Use [Chip\\_ADC\\_StopBurstSequencer\(\)](#) to stop the ADC burst sequence. START and BURST bits can not be set at the same time, thus, START bit will be cleared.

Definition at line 398 of file adc\_5411x.h.

#### 6.4.4.28 `__STATIC_INLINE void Chip_ADC_StartSequencer ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex )`

Forces a sequencer trigger event (software trigger of ADC)

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to start

## Returns

Nothing

## Note

This function sets the START bit for the sequencer to force a single conversion sequence or a single step conversion. START and BURST bits can not be set at the same time, thus, BURST bit will be cleared.

Definition at line 382 of file adc\_5411x.h.

6.4.4.29 `__STATIC_INLINE void Chip_ADC_StopBurstSequencer ( LPC_ADC_T * pADC, ADC_SEQ_IDX_T seqIndex )`

Stops sequencer burst mode.

## Parameters

<i>pADC</i>	: The base of ADC peripheral on the chip
<i>seqIndex</i>	: Sequencer to stop burst on

## Returns

Nothing

Definition at line 411 of file adc\_5411x.h.

## 6.5 CHIP: CHIP\_LPC5411X family IRQ vector names and mapped NVIC IRQ numbers

These are the interrupt vector names and the NVIC IRQ number mapping used for the LPC5411X family. You can override any WEAK function with a function in your application. If you do not create a function with one of these names in your application, the vector will be routed to a default handler (dead loop).

### Use example

```
/* Function created that handles PININT3 interrupt */
void PIN_INT3_IRQHandler(void)
{
    Chip_PININT_ClearIntStatus(LPC_PININT, PININTCH3);
    Board_LED_Toggle(0);
}

/* Enable IRQ for PININT3 */
NVIC_EnableIRQ(PIN_INT3_IRQn);
```

Also see [CHIP\\_5411X: LPC5411X M0 core peripheral interrupt numbers](#) Also see [CHIP\\_5411X: LPC5411X M4 core peripheral interrupt numbers](#)

### MX Core default handler names

```
void ResetISR(void); (Reset_IRQn)
WEAK void NMI_Handler(void); (NonMaskableInt_IRQn)
WEAK void HardFault_Handler(void); (HardFault_IRQn)
WEAK void SVC_Handler(void); (SVCall_IRQn)
WEAK void PendSV_Handler(void); (PendSV_IRQn)
WEAK void SysTick_Handler(void); (SysTick_IRQn)
```

### Peripheral vector names and mapped IRQs

```
WEAK void WDT_IRQHandler(void); (WDT_IRQn)
WEAK void BOD_IRQHandler(void); (BOD_IRQn)
WEAK void DMA_IRQHandler(void); (DMA_IRQn)
WEAK void GINT0_IRQHandler(void); (GINT0_IRQn)
WEAK void PIN_INT0_IRQHandler(void); (PIN_INT0_IRQn)
WEAK void PIN_INT1_IRQHandler(void); (PIN_INT1_IRQn)
WEAK void PIN_INT2_IRQHandler(void); (PIN_INT2_IRQn)
WEAK void PIN_INT3_IRQHandler(void); (PIN_INT3_IRQn)
WEAK void UTICK_IRQHandler(void); (UTICK_IRQn)
WEAK void MRT_IRQHandler(void); (MRT_IRQn)
WEAK void CT32B0_IRQHandler(void); (CT32B0_IRQn)
WEAK void CT32B1_IRQHandler(void); (CT32B1_IRQn)
WEAK void CT32B2_IRQHandler(void); (CT32B2_IRQn)
WEAK void CT32B3_IRQHandler(void); (CT32B3_IRQn)
WEAK void CT32B4_IRQHandler(void); (CT32B4_IRQn)
WEAK void SCT0_IRQHandler(void); (SCT0_IRQn)
WEAK void UART0_IRQHandler(void); (UART0_IRQn)
WEAK void UART1_IRQHandler(void); (UART1_IRQn)
WEAK void UART2_IRQHandler(void); (UART2_IRQn)
WEAK void UART3_IRQHandler(void); (UART3_IRQn)
WEAK void I2C0_IRQHandler(void); (I2C0_IRQn)
WEAK void I2C1_IRQHandler(void); (I2C1_IRQn)
WEAK void I2C2_IRQHandler(void); (I2C2_IRQn)
WEAK void SPI0_IRQHandler(void); (SPI0_IRQn)
WEAK void SPI1_IRQHandler(void); (SPI1_IRQn)
WEAK void ADC_SEQA_IRQHandler(void); (ADC_SEQA_IRQn)
WEAK void ADC_SEQB_IRQHandler(void); (ADC_SEQB_IRQn)
WEAK void ADC_THCMP_IRQHandler(void); (ADC_THCMP_IRQn)
WEAK void RTC_IRQHandler(void); (RTC_IRQn)
```

WEAK void IOH\_IRQHandler(void); (IOH\_IRQn)  
WEAK void MAILBOX\_IRQHandler(void); (MAILBOX\_IRQn)  
WEAK void GINT1\_IRQHandler(void); (GINT1\_IRQn)  
WEAK void PIN\_INT4\_IRQHandler(void); (PIN\_INT4\_IRQn)  
WEAK void PIN\_INT5\_IRQHandler(void); (PIN\_INT5\_IRQn)  
WEAK void PIN\_INT6\_IRQHandler(void); (PIN\_INT6\_IRQn)  
WEAK void PIN\_INT7\_IRQHandler(void); (PIN\_INT7\_IRQn)  
WEAK void RIT\_IRQHandler(void); (RIT\_IRQn)

## 6.6 CHIP: Common Chip ISP/IAP commands and return codes

### 6.6.1 Detailed Description

#### Macros

- `#define IAP_PREWRITE_CMD` 50
- `#define IAP_WRITESECTOR_CMD` 51
- `#define IAP_ERASESECTOR_CMD` 52
- `#define IAP_BLANK_CHECK_SECTOR_CMD` 53
- `#define IAP_READID_CMD` 54
- `#define IAP_READ_BOOT_CODE_CMD` 55
- `#define IAP_COMPARE_CMD` 56
- `#define IAP_REINVOKE_ISP_CMD` 57
- `#define IAP_READ_UID_CMD` 58
- `#define IAP_ERASE_PAGE_CMD` 59
- `#define IAP_EEPROM_WRITE` 61
- `#define IAP_EEPROM_READ` 62
- `#define IAP_CMD_SUCCESS` 0
- `#define IAP_INVALID_COMMAND` 1
- `#define IAP_SRC_ADDR_ERROR` 2
- `#define IAP_DST_ADDR_ERROR` 3
- `#define IAP_SRC_ADDR_NOT_MAPPED` 4
- `#define IAP_DST_ADDR_NOT_MAPPED` 5
- `#define IAP_COUNT_ERROR` 6
- `#define IAP_INVALID_SECTOR` 7
- `#define IAP_SECTOR_NOT_BLANK` 8
- `#define IAP_SECTOR_NOT_PREPARED` 9
- `#define IAP_COMPARE_ERROR` 10
- `#define IAP_BUSY` 11
- `#define IAP_PARAM_ERROR` 12
- `#define IAP_ADDR_ERROR` 13
- `#define IAP_ADDR_NOT_MAPPED` 14
- `#define IAP_CMD_LOCKED` 15
- `#define IAP_INVALID_CODE` 16
- `#define IAP_INVALID_BAUD_RATE` 17
- `#define IAP_INVALID_STOP_BIT` 18
- `#define IAP_CRP_ENABLED` 19

#### Typedefs

- `typedef void(* IAP_ENTRY_T)` (unsigned int[5], unsigned int[4])

#### Functions

- `uint8_t Chip_IAP_PreSectorForReadWrite` (uint32\_t strSector, uint32\_t endSector)  
*Prepare sector for write operation.*
- `uint8_t Chip_IAP_CopyRamToFlash` (uint32\_t dstAdd, uint32\_t \*srcAdd, uint32\_t byteswrt)  
*Copy RAM to flash.*
- `uint8_t Chip_IAP_EraseSector` (uint32\_t strSector, uint32\_t endSector)  
*Erase sector.*
- `uint8_t Chip_IAP_BlankCheckSector` (uint32\_t strSector, uint32\_t endSector)  
*Blank check a sector or multiples sector of on-chip flash memory.*

- uint32\_t [Chip\\_IAP\\_ReadPID](#) (void)  
*Read part identification number.*
- uint8\_t [Chip\\_IAP\\_ReadBootCode](#) (void)  
*Read boot code version number.*
- uint8\_t [Chip\\_IAP\\_Compare](#) (uint32\_t dstAdd, uint32\_t srcAdd, uint32\_t bytescmp)  
*Compare the memory contents at two locations.*
- uint8\_t [Chip\\_IAP\\_ReinvokeISP](#) (void)  
*IAP reinvokes ISP to invoke the bootloader in ISP mode.*
- uint32\_t [Chip\\_IAP\\_ReadUID](#) (void)  
*Read the unique ID.*
- uint8\_t [Chip\\_IAP\\_ErasePage](#) (uint32\_t strPage, uint32\_t endPage)  
*Erase a page or multiple pages of on-chip flash memory.*

## 6.6.2 Macro Definition Documentation

### 6.6.2.1 #define IAP\_ADDR\_ERROR 13

Address is not on word boundary

Definition at line 72 of file iap.h.

### 6.6.2.2 #define IAP\_ADDR\_NOT\_MAPPED 14

Address is not mapped in the memory map

Definition at line 73 of file iap.h.

### 6.6.2.3 #define IAP\_BLANK\_CHECK\_SECTOR\_CMD 53

Blank check sector

Definition at line 48 of file iap.h.

### 6.6.2.4 #define IAP\_BUSY 11

Flash programming hardware interface is busy

Definition at line 70 of file iap.h.

### 6.6.2.5 #define IAP\_CMD\_LOCKED 15

Command is locked

Definition at line 74 of file iap.h.

### 6.6.2.6 #define IAP\_CMD\_SUCCESS 0

Command is executed successfully

Definition at line 59 of file iap.h.

### 6.6.2.7 #define IAP\_COMPARE\_CMD 56

Compare two RAM address locations

Definition at line 51 of file iap.h.

**6.6.2.8 #define IAP\_COMPARE\_ERROR 10**

Source and destination data not equal

Definition at line 69 of file iap.h.

**6.6.2.9 #define IAP\_COUNT\_ERROR 6**

Byte count is not multiple of 4 or is not a permitted value

Definition at line 65 of file iap.h.

**6.6.2.10 #define IAP\_CRP\_ENABLED 19**

Code read protection enabled

Definition at line 78 of file iap.h.

**6.6.2.11 #define IAP\_DST\_ADDR\_ERROR 3**

Destination address is not on a correct boundary

Definition at line 62 of file iap.h.

**6.6.2.12 #define IAP\_DST\_ADDR\_NOT\_MAPPED 5**

Destination address is not mapped in the memory map

Definition at line 64 of file iap.h.

**6.6.2.13 #define IAP\_EEPROM\_READ 62**

EEPROM READ command

Definition at line 56 of file iap.h.

**6.6.2.14 #define IAP\_EEPROM\_WRITE 61**

EEPROM Write command

Definition at line 55 of file iap.h.

**6.6.2.15 #define IAP\_ERASE\_PAGE\_CMD 59**

Erase page

Definition at line 54 of file iap.h.

**6.6.2.16 #define IAP\_ERASECTOR\_CMD 52**

Erase Sector command

Definition at line 47 of file iap.h.



**6.6.2.17 #define IAP\_INVALID\_BAUD\_RATE 17**

Invalid baud rate setting

Definition at line 76 of file iap.h.

**6.6.2.18 #define IAP\_INVALID\_CODE 16**

Unlock code is invalid

Definition at line 75 of file iap.h.

**6.6.2.19 #define IAP\_INVALID\_COMMAND 1**

Invalid command

Definition at line 60 of file iap.h.

**6.6.2.20 #define IAP\_INVALID\_SECTOR 7**

Sector number is invalid or end sector number is greater than start sector number

Definition at line 66 of file iap.h.

**6.6.2.21 #define IAP\_INVALID\_STOP\_BIT 18**

Invalid stop bit setting

Definition at line 77 of file iap.h.

**6.6.2.22 #define IAP\_PARAM\_ERROR 12**

Insufficient number of parameters or invalid parameter

Definition at line 71 of file iap.h.

**6.6.2.23 #define IAP\_PREWRITE\_CMD 50**

Prepare sector for write operation command

Definition at line 45 of file iap.h.

**6.6.2.24 #define IAP\_READ\_BOOT\_CODE\_CMD 55**

Read Boot code version

Definition at line 50 of file iap.h.

**6.6.2.25 #define IAP\_READ\_UID\_CMD 58**

Read UID

Definition at line 53 of file iap.h.

#### 6.6.2.26 `#define IAP_REINVOKE_ISP_CMD` 57

Reinvoke ISP

Definition at line 52 of file iap.h.

#### 6.6.2.27 `#define IAP_REPID_CMD` 54

Read PartID command

Definition at line 49 of file iap.h.

#### 6.6.2.28 `#define IAP_SECTOR_NOT_BLANK` 8

Sector is not blank

Definition at line 67 of file iap.h.

#### 6.6.2.29 `#define IAP_SECTOR_NOT_PREPARED` 9

Command to prepare sector for write operation was not executed

Definition at line 68 of file iap.h.

#### 6.6.2.30 `#define IAP_SRC_ADDR_ERROR` 2

Source address is not on word boundary

Definition at line 61 of file iap.h.

#### 6.6.2.31 `#define IAP_SRC_ADDR_NOT_MAPPED` 4

Source address is not mapped in the memory map

Definition at line 63 of file iap.h.

#### 6.6.2.32 `#define IAP_WRISECTOR_CMD` 51

Write Sector command

Definition at line 46 of file iap.h.

### 6.6.3 Typedef Documentation

#### 6.6.3.1 `typedef void(* IAP_ENTRY_T) (unsigned int[5], unsigned int[4])`

Definition at line 81 of file iap.h.

### 6.6.4 Function Documentation

#### 6.6.4.1 `uint8_t Chip_IAP_BlankCheckSector ( uint32_t strSector, uint32_t endSector )`

Blank check a sector or multiples sector of on-chip flash memory.

## Parameters

<i>strSector</i>	: Start sector number
<i>endSector</i>	: End sector number

## Returns

Offset of the first non blank word location if the status code is SECTOR\_NOT\_BLANK

## Note

The end sector must be greater than or equal to start sector number

Definition at line 93 of file iap.c.

6.6.4.2 uint8\_t Chip\_IAP\_Compare ( uint32\_t *dstAdd*, uint32\_t *srcAdd*, uint32\_t *bytescmp* )

Compare the memory contents at two locations.

## Parameters

<i>dstAdd</i>	: Destination of the RAM address of data bytes to be compared
<i>srcAdd</i>	: Source of the RAM address of data bytes to be compared
<i>bytescmp</i>	: Number of bytes to be compared

## Returns

Offset of the first mismatch of the status code is COMPARE\_ERROR

## Note

The addresses should be a word boundary and number of bytes should be a multiply of 4

Definition at line 128 of file iap.c.

6.6.4.3 uint8\_t Chip\_IAP\_CopyRamToFlash ( uint32\_t *dstAdd*, uint32\_t \* *srcAdd*, uint32\_t *byteswrt* )

Copy RAM to flash.

## Parameters

<i>dstAdd</i>	: Destination flash address where data bytes are to be written
<i>srcAdd</i>	: Source flash address where data bytes are to be read
<i>byteswrt</i>	: Number of bytes to be written

## Returns

Status code to indicate the command is executed successfully or not

## Note

The addresses should be a 256 byte boundary and the number of bytes should be 256 | 512 | 1024 | 4096

Definition at line 64 of file iap.c.

6.6.4.4 uint8\_t Chip\_IAP\_ErasePage ( uint32\_t *strPage*, uint32\_t *endPage* )

Erase a page or multiple papers of on-chip flash memory.

**Parameters**

<i>strPage</i>	: Start page number
<i>endPage</i>	: End page number

**Returns**

Status code to indicate the command is executed successfully or not

**Note**

The page number must be greater than or equal to start page number

Definition at line 164 of file iap.c.

**6.6.4.5 uint8\_t Chip\_IAP\_EraseSector ( uint32\_t *strSector*, uint32\_t *endSector* )**

Erase sector.

**Parameters**

<i>strSector</i>	: Start sector number
<i>endSector</i>	: End sector number

**Returns**

Status code to indicate the command is executed successfully or not

**Note**

The end sector must be greater than or equal to start sector number

Definition at line 79 of file iap.c.

**6.6.4.6 uint8\_t Chip\_IAP\_PreSectorForReadWrite ( uint32\_t *strSector*, uint32\_t *endSector* )**

Prepare sector for write operation.

**Parameters**

<i>strSector</i>	: Start sector number
<i>endSector</i>	: End sector number

**Returns**

Status code to indicate the command is executed successfully or not

**Note**

This command must be executed before executing "Copy RAM to flash" or "Erase Sector" command. The end sector must be greater than or equal to start sector number

Definition at line 51 of file iap.c.

**6.6.4.7 uint8\_t Chip\_IAP\_ReadBootCode ( void )**

Read boot code version number.

**Returns**

Boot code version number

Definition at line 117 of file iap.c.

**6.6.4.8 uint32\_t Chip\_IAP\_ReadPID ( void )**

Read part identification number.

**Returns**

Part identification number

Definition at line 106 of file iap.c.

**6.6.4.9 uint32\_t Chip\_IAP\_ReadUID ( void )**

Read the unique ID.

**Returns**

Status code to indicate the command is executed successfully or not

Definition at line 153 of file iap.c.

**6.6.4.10 uint8\_t Chip\_IAP\_ReinvokeISP ( void )**

IAP reinvoke ISP to invoke the bootloader in ISP mode.

**Returns**

none

Definition at line 142 of file iap.c.

## 6.7 CHIP: FPU initialization

### 6.7.1 Detailed Description

Cortex FPU initialization

#### Functions

- void `fpulnit` (void)  
*Early initialization of the FPU.*

### 6.7.2 Function Documentation

#### 6.7.2.1 void `fpulnit` ( void )

Early initialization of the FPU.

#### Returns

Nothing

## 6.8 CHIP: LPC Common Types

### 6.8.1 Detailed Description

#### Modules

- [LPC Public Macros](#)
- [LPC Public Types](#)

## 6.9 CHIP: LPC5410x family CMSIS include files

IRQ vector names and NVIC vector mapping for the LPC5410x device



## 6.10 CHIP: LPC5411X 32-bit Timer driver

### 6.10.1 Detailed Description

#### Data Structures

- struct [LPC\\_TIMER\\_T](#)  
*32-bit Standard timer register block structure*

#### Macros

- `#define` [TIMER\\_IR\\_CLR](#)(n) `_BIT`(n)
- `#define` [TIMER\\_MATCH\\_INT](#)(n) `_BIT`((n) & 0x0F)
- `#define` [TIMER\\_CAP\\_INT](#)(n) `_BIT`((n) & 0x0F + 4))
- `#define` [TIMER\\_ENABLE](#) ((uint32\_t) (1 << 0))
- `#define` [TIMER\\_RESET](#) ((uint32\_t) (1 << 1))
- `#define` [TIMER\\_CTRL\\_MASK](#) ((uint32\_t) 0x03)
- `#define` [TIMER\\_INT\\_ON\\_MATCH](#)(n) `_BIT`((n) \* 3))
- `#define` [TIMER\\_RESET\\_ON\\_MATCH](#)(n) `_BIT`((n) \* 3 + 1))
- `#define` [TIMER\\_STOP\\_ON\\_MATCH](#)(n) `_BIT`((n) \* 3 + 2))
- `#define` [TIMER\\_MCR\\_MASK](#) ((uint32\_t) 0x0FFF)
- `#define` [TIMER\\_CAP\\_RISING](#)(n) `_BIT`((n) \* 3))
- `#define` [TIMER\\_CAP\\_FALLING](#)(n) `_BIT`((n) \* 3 + 1))
- `#define` [TIMER\\_INT\\_ON\\_CAP](#)(n) `_BIT`((n) \* 3 + 2))
- `#define` [TIMER\\_CCR\\_MASK](#) ((uint32\_t) 0x0FFF)
- `#define` [TIMER\\_EMR\\_MASK](#) ((uint32\_t) 0x0FFF)
- `#define` [TIMER\\_CTCR\\_MASK](#) ((uint32\_t) 0x0F)

#### Enumerations

- enum [TIMER\\_PIN\\_MATCH\\_STATE\\_T](#) { [TIMER\\_EXTMATCH\\_DO\\_NOTHING](#) = 0, [TIMER\\_EXTMATCH\\_CLEAR](#) = 1, [TIMER\\_EXTMATCH\\_SET](#) = 2, [TIMER\\_EXTMATCH\\_TOGGLE](#) = 3 }
- *Standard timer initial match pin state and change state.*
- enum [TIMER\\_CAP\\_SRC\\_STATE\\_T](#) { [TIMER\\_CAPSRC\\_RISING\\_PCLK](#) = 0, [TIMER\\_CAPSRC\\_RISING\\_CAPN](#) = 1, [TIMER\\_CAPSRC\\_FALLING\\_CAPN](#) = 2, [TIMER\\_CAPSRC\\_BOTH\\_CAPN](#) = 3 }
- *Standard timer clock and edge for count source.*

#### Functions

- `__STATIC_INLINE` void [Chip\\_TIMER\\_Init](#) ([LPC\\_TIMER\\_T](#) \*pTMR)  
*Initialize a timer.*
- `__STATIC_INLINE` void [Chip\\_TIMER\\_DeInit](#) ([LPC\\_TIMER\\_T](#) \*pTMR)  
*Shutdown a timer.*
- `__STATIC_INLINE` bool [Chip\\_TIMER\\_MatchPending](#) ([LPC\\_TIMER\\_T](#) \*pTMR, int8\_t matchnum)  
*Determine if a match interrupt is pending.*
- `__STATIC_INLINE` bool [Chip\\_TIMER\\_CapturePending](#) ([LPC\\_TIMER\\_T](#) \*pTMR, int8\_t capnum)  
*Determine if a capture interrupt is pending.*
- `__STATIC_INLINE` void [Chip\\_TIMER\\_ClearMatch](#) ([LPC\\_TIMER\\_T](#) \*pTMR, int8\_t matchnum)  
*Clears a (pending) match interrupt.*
- `__STATIC_INLINE` void [Chip\\_TIMER\\_ClearCapture](#) ([LPC\\_TIMER\\_T](#) \*pTMR, int8\_t capnum)  
*Clears a (pending) capture interrupt.*

- `__STATIC_INLINE void Chip_TIMER_Enable (LPC_TIMER_T *pTMR)`  
*Enables the timer (starts count)*
- `__STATIC_INLINE void Chip_TIMER_Disable (LPC_TIMER_T *pTMR)`  
*Disables the timer (stops count)*
- `__STATIC_INLINE uint32_t Chip_TIMER_ReadCount (LPC_TIMER_T *pTMR)`  
*Returns the current timer count.*
- `__STATIC_INLINE uint32_t Chip_TIMER_ReadPrescale (LPC_TIMER_T *pTMR)`  
*Returns the current prescale count.*
- `__STATIC_INLINE void Chip_TIMER_PrescaleSet (LPC_TIMER_T *pTMR, uint32_t prescale)`  
*Sets the prescaler value.*
- `__STATIC_INLINE void Chip_TIMER_SetMatch (LPC_TIMER_T *pTMR, int8_t matchnum, uint32_t matchval)`  
*Sets a timer match value.*
- `__STATIC_INLINE uint32_t Chip_TIMER_ReadCapture (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Reads a capture register.*
- `void Chip_TIMER_Reset (LPC_TIMER_T *pTMR)`  
*Resets the timer terminal and prescale counts to 0.*
- `__STATIC_INLINE void Chip_TIMER_MatchEnableInt (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Enables a match interrupt that fires when the terminal count matches the match counter value.*
- `__STATIC_INLINE void Chip_TIMER_MatchDisableInt (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Disables a match interrupt for a match counter.*
- `__STATIC_INLINE void Chip_TIMER_ResetOnMatchEnable (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*For the specific match counter, enables reset of the terminal count register when a match occurs.*
- `__STATIC_INLINE void Chip_TIMER_ResetOnMatchDisable (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*For the specific match counter, disables reset of the terminal count register when a match occurs.*
- `__STATIC_INLINE void Chip_TIMER_StopOnMatchEnable (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Enable a match timer to stop the terminal count when a match count equals the terminal count.*
- `__STATIC_INLINE void Chip_TIMER_StopOnMatchDisable (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Disable stop on match for a match timer. Disables a match timer to stop the terminal count when a match count equals the terminal count.*
- `__STATIC_INLINE void Chip_TIMER_CaptureRisingEdgeEnable (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Enables capture on on rising edge of selected CAP signal for the selected capture register, enables the selected CAPn.capnum signal to load the capture register with the terminal coount on a rising edge.*
- `__STATIC_INLINE void Chip_TIMER_CaptureRisingEdgeDisable (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Disables capture on on rising edge of selected CAP signal. For the selected capture register, disables the selected CAPn.capnum signal to load the capture register with the terminal coount on a rising edge.*
- `__STATIC_INLINE void Chip_TIMER_CaptureFallingEdgeEnable (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Enables capture on on falling edge of selected CAP signal. For the selected capture register, enables the selected CAPn.capnum signal to load the capture register with the terminal coount on a falling edge.*
- `__STATIC_INLINE void Chip_TIMER_CaptureFallingEdgeDisable (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Disables capture on on falling edge of selected CAP signal. For the selected capture register, disables the selected CAPn.capnum signal to load the capture register with the terminal coount on a falling edge.*
- `__STATIC_INLINE void Chip_TIMER_CaptureEnableInt (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Enables interrupt on capture of selected CAP signal. For the selected capture register, an interrupt will be generated when the enabled rising or falling edge on CAPn.capnum is detected.*
- `__STATIC_INLINE void Chip_TIMER_CaptureDisableInt (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Disables interrupt on capture of selected CAP signal.*
- `void Chip_TIMER_ExtMatchControlSet (LPC_TIMER_T *pTMR, int8_t initial_state, TIMER_PIN_MATCH_STATE_T matchState, int8_t matchnum)`  
*Sets external match control (MATn.matchnum) pin control. For the pin selected with matchnum, sets the function of the pin that occurs on a terminal count match for the match count.*
- `__STATIC_INLINE void Chip_TIMER_TIMER_SetCountClockSrc (LPC_TIMER_T *pTMR, TIMER_CAP_STATE_T capSrc, int8_t capnum)`  
*Sets timer count source and edge with the selected passed from CapSrc. If CapSrc selected a CAPn pin, select the specific CAPn pin with the capnum value.*

## 6.10.2 Macro Definition Documentation

### 6.10.2.1 `#define TIMER_CAP_FALLING( n ) ( _BIT((((n) * 3) + 1))`

Bit location for CAP.n on CRx falling edge, n = 0 to 3

Definition at line 90 of file timer\_5411x.h.

### 6.10.2.2 `#define TIMER_CAP_INT( n ) ( _BIT(((n) & 0x0F) + 4))`

Macro for getting a capture event interrupt bit

Definition at line 69 of file timer\_5411x.h.

### 6.10.2.3 `#define TIMER_CAP_RISING( n ) ( _BIT(((n) * 3))`

Bit location for CAP.n on CRx rising edge, n = 0 to 3

Definition at line 88 of file timer\_5411x.h.

### 6.10.2.4 `#define TIMER_CCR_MASK ((uint32_t) 0xFFFF)`

Capture Control register Mask

Definition at line 94 of file timer\_5411x.h.

### 6.10.2.5 `#define TIMER_CTCR_MASK ((uint32_t) 0x0F)`

Counter Control register Mask

Definition at line 98 of file timer\_5411x.h.

### 6.10.2.6 `#define TIMER_CTRL_MASK ((uint32_t) 0x03)`

Timer Control register Mask

Definition at line 76 of file timer\_5411x.h.

### 6.10.2.7 `#define TIMER_EMR_MASK ((uint32_t) 0xFFFF)`

External Match register Mask

Definition at line 96 of file timer\_5411x.h.

### 6.10.2.8 `#define TIMER_ENABLE ((uint32_t) (1 << 0))`

Timer/counter enable bit

Definition at line 72 of file timer\_5411x.h.

### 6.10.2.9 `#define TIMER_INT_ON_CAP( n ) ( _BIT((((n) * 3) + 2))`

Bit location for CAP.n on CRx interrupt enable, n = 0 to 3

Definition at line 92 of file timer\_5411x.h.

6.10.2.10 **#define** `TIMER_INT_ON_MATCH( n ) ( _BIT(((n) * 3)))`

Bit location for interrupt on MRx match, n = 0 to 3

Definition at line 79 of file timer\_5411x.h.

6.10.2.11 **#define** `TIMER_IR_CLR( n ) _BIT(n)`

Macro to clear interrupt pending

Definition at line 64 of file timer\_5411x.h.

6.10.2.12 **#define** `TIMER_MATCH_INT( n ) ( _BIT((n) & 0x0F))`

Macro for getting a timer match interrupt bit

Definition at line 67 of file timer\_5411x.h.

6.10.2.13 **#define** `TIMER_MCR_MASK ((uint32_t) 0x0FFF)`

Match Control register Mask

Definition at line 85 of file timer\_5411x.h.

6.10.2.14 **#define** `TIMER_RESET ((uint32_t) (1 << 1))`

Timer/counter reset bit

Definition at line 74 of file timer\_5411x.h.

6.10.2.15 **#define** `TIMER_RESET_ON_MATCH( n ) ( _BIT((((n) * 3) + 1)))`

Bit location for reset on MRx match, n = 0 to 3

Definition at line 81 of file timer\_5411x.h.

6.10.2.16 **#define** `TIMER_STOP_ON_MATCH( n ) ( _BIT((((n) * 3) + 2)))`

Bit location for stop on MRx match, n = 0 to 3

Definition at line 83 of file timer\_5411x.h.

## 6.10.3 Enumeration Type Documentation

6.10.3.1 **enum** `TIMER_CAP_SRC_STATE_T`

Standard timer clock and edge for count source.

Enumerator

***TIMER\_CAPSRC\_RISING\_PCLK*** Timer ticks on PCLK rising edge

***TIMER\_CAPSRC\_RISING\_CAPN*** Timer ticks on CAPn.x rising edge

***TIMER\_CAPSRC\_FALLING\_CAPN*** Timer ticks on CAPn.x falling edge

***TIMER\_CAPSRC\_BOTH\_CAPN*** Timer ticks on CAPn.x both edges

Definition at line 464 of file timer\_5411x.h.

6.10.3.2 enum `TIMER_PIN_MATCH_STATE_T`

Standard timer initial match pin state and change state.

## Enumerator

**`TIMER_EXTMATCH_DO_NOTHING`** Timer match state does nothing on match pin

**`TIMER_EXTMATCH_CLEAR`** Timer match state sets match pin low

**`TIMER_EXTMATCH_SET`** Timer match state sets match pin high

**`TIMER_EXTMATCH_TOGGLE`** Timer match state toggles match pin

Definition at line 439 of file `timer_5411x.h`.

## 6.10.4 Function Documentation

6.10.4.1 `__STATIC_INLINE void Chip_TIMER_CaptureDisableInt ( LPC_TIMER_T * pTMR, int8_t capnum )`

Disables interrupt on capture of selected CAP signal.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>capnum</i>	: Capture signal/register to use

## Returns

Nothing

Definition at line 431 of file `timer_5411x.h`.

6.10.4.2 `__STATIC_INLINE void Chip_TIMER_CaptureEnableInt ( LPC_TIMER_T * pTMR, int8_t capnum )`

Enables interrupt on capture of selected CAP signal. For the selected capture register, an interrupt will be generated when the enabled rising or falling edge on CAPn.capnum is detected.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>capnum</i>	: Capture signal/register to use

## Returns

Nothing

Definition at line 420 of file `timer_5411x.h`.

6.10.4.3 `__STATIC_INLINE void Chip_TIMER_CaptureFallingEdgeDisable ( LPC_TIMER_T * pTMR, int8_t capnum )`

Disables capture on on falling edge of selected CAP signal. For the selected capture register, disables the selected CAPn.capnum signal to load the capture register with the terminal count on a falling edge.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>capnum</i>	: Capture signal/register to use

**Returns**

Nothing

Definition at line 407 of file timer\_5411x.h.

#### 6.10.4.4 `__STATIC_INLINE void Chip_TIMER_CaptureFallingEdgeEnable ( LPC_TIMER_T * pTMR, int8_t capnum )`

Enables capture on on falling edge of selected CAP signal. For the selected capture register, enables the selected CAPn.capnum signal to load the capture register with the terminal coount on a falling edge.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
<i>capnum</i>	: Capture signal/register to use

**Returns**

Nothing

Definition at line 394 of file timer\_5411x.h.

#### 6.10.4.5 `__STATIC_INLINE bool Chip_TIMER_CapturePending ( LPC_TIMER_T * pTMR, int8_t capnum )`

Determine if a capture interrupt is pending.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
<i>capnum</i>	: Capture interrupt number to check

**Returns**

false if the interrupt is not pending, otherwise true

**Note**

Determine if the capture interrupt for the passed capture pin is pending.

Definition at line 174 of file timer\_5411x.h.

#### 6.10.4.6 `__STATIC_INLINE void Chip_TIMER_CaptureRisingEdgeDisable ( LPC_TIMER_T * pTMR, int8_t capnum )`

Disables capture on on rising edge of selected CAP signal. For the selected capture register, disables the selected CAPn.capnum signal to load the capture register with the terminal coount on a rising edge.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
<i>capnum</i>	: Capture signal/register to use

**Returns**

Nothing

Definition at line 381 of file timer\_5411x.h.

6.10.4.7 `__STATIC_INLINE void Chip_TIMER_CaptureRisingEdgeEnable ( LPC_TIMER_T * pTMR, int8_t capnum )`

Enables capture on on rising edge of selected CAP signal for the selected capture register, enables the selected CAPn.capnum signal to load the capture register with the terminal coount on a rising edge.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>capnum</i>	: Capture signal/register to use

## Returns

Nothing

Definition at line 368 of file timer\_5411x.h.

#### 6.10.4.8 `__STATIC_INLINE void Chip_TIMER_ClearCapture ( LPC_TIMER_T * pTMR, int8_t capnum )`

Clears a (pending) capture interrupt.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>capnum</i>	: Capture interrupt number to clear

## Returns

Nothing

## Note

Clears a pending timer capture interrupt.

Definition at line 198 of file timer\_5411x.h.

#### 6.10.4.9 `__STATIC_INLINE void Chip_TIMER_ClearMatch ( LPC_TIMER_T * pTMR, int8_t matchnum )`

Clears a (pending) match interrupt.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>matchnum</i>	: Match interrupt number to clear

## Returns

Nothing

## Note

Clears a pending timer match interrupt.

Definition at line 186 of file timer\_5411x.h.

#### 6.10.4.10 `__STATIC_INLINE void Chip_TIMER_DeInit ( LPC_TIMER_T * pTMR )`

Shutdown a timer.



## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
-------------	--

## Returns

Nothing

Definition at line 134 of file timer\_5411x.h.

#### 6.10.4.11 `__STATIC_INLINE void Chip_TIMER_Disable ( LPC_TIMER_T * pTMR )`

Disables the timer (stops count)

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
-------------	--

## Returns

Nothing

## Note

Disables the timer to stop counting.

Definition at line 220 of file timer\_5411x.h.

#### 6.10.4.12 `__STATIC_INLINE void Chip_TIMER_Enable ( LPC_TIMER_T * pTMR )`

Enables the timer (starts count)

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
-------------	--

## Returns

Nothing

## Note

Enables the timer to start counting.

Definition at line 209 of file timer\_5411x.h.

#### 6.10.4.13 `void Chip_TIMER_ExtMatchControlSet ( LPC_TIMER_T * pTMR, int8_t initial_state, TIMER_PIN_MATCH_STATE_T matchState, int8_t matchnum )`

Sets external match control (MATn.matchnum) pin control. For the pin selected with matchnum, sets the function of the pin that occurs on a terminal count match for the match count.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
<i>initial_state</i>	: Initial state of the pin, high(1) or low(0)
<i>matchState</i>	: Selects the match state for the pin
<i>matchnum</i>	: MATn.matchnum signal to use

**Returns**

Nothing

**Note**

For the pin selected with matchnum, sets the function of the pin that occurs on a terminal count match for the match count.

Definition at line 72 of file timer\_5411x.c.

#### 6.10.4.14 `__STATIC_INLINE void Chip_TIMER_Init ( LPC_TIMER_T * pTMR )`

Initialize a timer.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
-------------	--

**Returns**

Nothing

Definition at line 105 of file timer\_5411x.h.

#### 6.10.4.15 `__STATIC_INLINE void Chip_TIMER_MatchDisableInt ( LPC_TIMER_T * pTMR, int8_t matchnum )`

Disables a match interrupt for a match counter.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
<i>matchnum</i>	: Match timer, 0 to 3

**Returns**

Nothing

Definition at line 309 of file timer\_5411x.h.

#### 6.10.4.16 `__STATIC_INLINE void Chip_TIMER_MatchEnableInt ( LPC_TIMER_T * pTMR, int8_t matchnum )`

Enables a match interrupt that fires when the terminal count matches the match counter value.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
-------------	--

<i>matchnum</i>	: Match timer, 0 to 3
-----------------	-----------------------

**Returns**

Nothing

Definition at line 298 of file timer\_5411x.h.

**6.10.4.17** `__STATIC_INLINE bool Chip_TIMER_MatchPending ( LPC_TIMER_T * pTMR, int8_t matchnum )`

Determine if a match interrupt is pending.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
<i>matchnum</i>	: Match interrupt number to check

**Returns**

false if the interrupt is not pending, otherwise true

**Note**

Determine if the match interrupt for the passed timer and match counter is pending.

Definition at line 161 of file timer\_5411x.h.

**6.10.4.18** `__STATIC_INLINE void Chip_TIMER_PrescaleSet ( LPC_TIMER_T * pTMR, uint32_t prescale )`

Sets the prescaler value.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
<i>prescale</i>	: Prescale value to set the prescale register to

**Returns**

Nothing

**Note**

Sets the prescale count value.

Definition at line 254 of file timer\_5411x.h.

**6.10.4.19** `__STATIC_INLINE uint32_t Chip_TIMER_ReadCapture ( LPC_TIMER_T * pTMR, int8_t capnum )`

Reads a capture register.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
<i>capnum</i>	: Capture register to read

**Returns**

The selected capture register value

**Note**

Returns the selected capture register value.

Definition at line 279 of file timer\_5411x.h.

#### 6.10.4.20 `__STATIC_INLINE uint32_t Chip_TIMER_ReadCount ( LPC_TIMER_T * pTMR )`

Returns the current timer count.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
-------------	--

**Returns**

Current timer terminal count value

**Note**

Returns the current timer terminal count.

Definition at line 231 of file timer\_5411x.h.

#### 6.10.4.21 `__STATIC_INLINE uint32_t Chip_TIMER_ReadPrescale ( LPC_TIMER_T * pTMR )`

Returns the current prescale count.

**Parameters**

<i>pTMR</i>	: Pointer to timer IP register address
-------------	--

**Returns**

Current timer prescale count value

**Note**

Returns the current prescale count.

Definition at line 242 of file timer\_5411x.h.

#### 6.10.4.22 `void Chip_TIMER_Reset ( LPC_TIMER_T * pTMR )`

Resets the timer terminal and prescale counts to 0.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
-------------	--

## Returns

Nothing

Definition at line 52 of file timer\_5411x.c.

#### 6.10.4.23 `__STATIC_INLINE void Chip_TIMER_ResetOnMatchDisable ( LPC_TIMER_T * pTMR, int8_t matchnum )`

For the specific match counter, disables reset of the terminal count register when a match occurs.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>matchnum</i>	: Match timer, 0 to 3

## Returns

Nothing

Definition at line 331 of file timer\_5411x.h.

#### 6.10.4.24 `__STATIC_INLINE void Chip_TIMER_ResetOnMatchEnable ( LPC_TIMER_T * pTMR, int8_t matchnum )`

For the specific match counter, enables reset of the terminal count register when a match occurs.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>matchnum</i>	: Match timer, 0 to 3

## Returns

Nothing

Definition at line 320 of file timer\_5411x.h.

#### 6.10.4.25 `__STATIC_INLINE void Chip_TIMER_SetMatch ( LPC_TIMER_T * pTMR, int8_t matchnum, uint32_t matchval )`

Sets a timer match value.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>matchnum</i>	: Match timer to set match count for
<i>matchval</i>	: Match value for the selected match count

## Returns

Nothing

## Note

Sets one of the timer match values.

Definition at line 267 of file timer\_5411x.h.

6.10.4.26 `__STATIC_INLINE void Chip_TIMER_StopOnMatchDisable ( LPC_TIMER_T * pTMR, int8_t matchnum )`

Disable stop on match for a match timer. Disables a match timer to stop the terminal count when a match count equals the terminal count.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>matchnum</i>	: Match timer, 0 to 3

## Returns

Nothing

Definition at line 355 of file timer\_5411x.h.

#### 6.10.4.27 `__STATIC_INLINE void Chip_TIMER_StopOnMatchEnable ( LPC_TIMER_T * pTMR, int8_t matchnum )`

Enable a match timer to stop the terminal count when a match count equals the terminal count.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>matchnum</i>	: Match timer, 0 to 3

## Returns

Nothing

Definition at line 343 of file timer\_5411x.h.

#### 6.10.4.28 `__STATIC_INLINE void Chip_TIMER_TIMER_SetCountClockSrc ( LPC_TIMER_T * pTMR, TIMER_CAP_SRC_STATE_T capSrc, int8_t capnum )`

Sets timer count source and edge with the selected passed from CapSrc. If CapSrc selected a CAPn pin, select the specific CAPn pin with the capnum value.

## Parameters

<i>pTMR</i>	: Pointer to timer IP register address
<i>capSrc</i>	: timer clock source and edge
<i>capnum</i>	: CAPn.capnum pin to use (0 - 2)

## Returns

Nothing

## Note

If CapSrc selected a CAPn pin, select the specific CAPn pin with the capnum value.

Definition at line 480 of file timer\_5411x.h.

## 6.11 CHIP: LPC5411X CPU multi-core support driver

### 6.11.1 Detailed Description

This driver helps with determine which MCU core the software is running, whether the MCU core is in master or slave mode, and provides functions for master and slave core control.

The functions for the driver are provided as part of the [CHIP: LPC5411X Power LIBRARY functions](#) library. For more information on using the LPC5411x LPCOpen package with multi-core, see [LPC5411X multi-core use in LPCOpen](#).

### Enumerations

- enum [CORESELECT\\_T](#) { [CORESELECT\\_M0PLUS](#) = 0, [CORESELECT\\_M4](#) }

### Functions

- \_\_STATIC\_INLINE bool [Chip\\_CPU\\_IsM4Core](#) (void)  
*Determine which MCU this code is running on.*
- void [Chip\\_CPU\\_SelectMasterCore](#) ([CORESELECT\\_T](#) master, [CORESELECT\\_T](#) ownerPower)  
*Select master core and system power control ownership.*
- bool [Chip\\_CPU\\_IsMasterCore](#) (void)  
*Determine if this core is a slave or master.*
- void [Chip\\_CPU\\_CM0Boot](#) (uint32\_t \*coentry, uint32\_t \*costackptr)  
*Setup M0+ boot and reset M0+ core.*
- void [Chip\\_CPU\\_CM4Boot](#) (uint32\_t \*coentry, uint32\_t \*costackptr)  
*Setup M4 boot and reset M4 core.*

### 6.11.2 Enumeration Type Documentation

#### 6.11.2.1 enum CORESELECT\_T

##### Enumerator

**[CORESELECT\\_M0PLUS](#)**  
**[CORESELECT\\_M4](#)**

Definition at line 65 of file `cpuctrl_5411x.h`.

### 6.11.3 Function Documentation

#### 6.11.3.1 void [Chip\\_CPU\\_CM0Boot](#) ( uint32\_t \* *coentry*, uint32\_t \* *costackptr* )

Setup M0+ boot and reset M0+ core.

##### Parameters

<i>coentry</i>	: Pointer to boot entry point for M0+ core
<i>costackptr</i>	: Pointer to where stack should be located for M0+ core

##### Returns

Nothing



**Note**

Will setup boot stack and entry point, enable M0+ clock and then reset M0+ core.

### 6.11.3.2 void Chip\_CPU\_CM4Boot ( uint32\_t \* *coentry*, uint32\_t \* *costackptr* )

Setup M4 boot and reset M4 core.

**Parameters**

<i>coentry</i>	: Pointer to boot entry point for M4 core
<i>costackptr</i>	: Pointer to where stack should be located for M4 core

**Returns**

Nothing

**Note**

Will setup boot stack and entry point, enable M4 clock and then reset M0+ core.

### 6.11.3.3 \_\_STATIC\_INLINE bool Chip\_CPU\_IsM4Core ( void )

Determine which MCU this code is running on.

**Returns**

true if executing on the CM4, or false if executing on the CM0+

Definition at line 55 of file cpuctrl\_5411x.h.

### 6.11.3.4 bool Chip\_CPU\_IsMasterCore ( void )

Determine if this core is a slave or master.

**Returns**

true if this MCU is operating as the master, or false if operating as a slave

### 6.11.3.5 void Chip\_CPU\_SelectMasterCore ( CORESELECT\_T *master*, CORESELECT\_T *ownerPower* )

Select master core and system power control ownership.

**Returns**

Nothing

**Note**

This function can be used to select the master core and which core can powerdown the system. The master core can be re-selected on either the current master or slave core. Power control ownership is used to select which core can place the system in DEEP SLEEP, POWERDOWN, and DEEP POWERDOWN modes. (See [Chip\\_POWER\\_EnterPowerMode](#)). Note both the master and slave cores can used SLEEP mode, but only the master core can use the other modes.

## 6.12 CHIP: LPC5411X Clock Driver

### 6.12.1 Detailed Description

#### Macros

- #define `SYSCON_FRO12MHZ_FREQ` (12000000)
- #define `SYSCON_FRO48MHZ_FREQ` (48000000)
- #define `SYSCON_FRO96MHZ_FREQ` (96000000)
- #define `SYSCON_WDTOSC_FREQ` (500000)
- #define `SYSCON_RTC_FREQ` (32768)
- #define `Chip_Clock_GetIntOscRate()` `SYSCON_FRO12MHZ_FREQ`

#### Enumerations

- enum `WDT_OSC_FREQ_T` {  
`WDT_FREQ_RESERVED`, `WDT_FREQ_400000`, `WDT_FREQ_600000`, `WDT_FREQ_750000`,  
`WDT_FREQ_900000`, `WDT_FREQ_1000000`, `WDT_FREQ_1200000`, `WDT_FREQ_1300000`,  
`WDT_FREQ_1400000`, `WDT_FREQ_1500000`, `WDT_FREQ_1600000`, `WDT_FREQ_1700000`,  
`WDT_FREQ_1800000`, `WDT_FREQ_1900000`, `WDT_FREQ_2000000`, `WDT_FREQ_2050000`,  
`WDT_FREQ_2100000`, `WDT_FREQ_2200000`, `WDT_FREQ_2250000`, `WDT_FREQ_2300000`,  
`WDT_FREQ_2400000`, `WDT_FREQ_2450000`, `WDT_FREQ_2500000`, `WDT_FREQ_2600000`,  
`WDT_FREQ_2650000`, `WDT_FREQ_2700000`, `WDT_FREQ_2800000`, `WDT_FREQ_2850000`,  
`WDT_FREQ_2900000`, `WDT_FREQ_2950000`, `WDT_FREQ_3000000`, `WDT_FREQ_3050000` }  
*WDT Osc frequency value table.*
- enum `CHIP_SYSCON_MAIN_A_CLKSRC_T` { `SYSCON_MAIN_A_CLKSRC_FRO12MHZ` = 0, `SYSCON_↵`  
`_MAIN_A_CLKSRCA_CLKIN`, `SYSCON_MAIN_A_CLKSRCA_WDTOSC`, `SYSCON_MAIN_A_CLKSRCA_↵`  
`_FROHF` }
- enum `CHIP_SYSCON_USBCLKSRC_T` { `SYSCON_USBCLKSRC_FROHF`, `SYSCON_USBCLKSRC_PLL`,  
`SYSCON_USBCLKSRC_DISABLED` = 7 }
- *USB Clock source.*
- enum `CHIP_SYSCON_MCLKSRC_T` { `SYSCON_MCLKSRC_FROHF`, `SYSCON_MCLKSRC_PLL`, `SYSC_↵`  
`ON_MCLKSRC_MCLKIN`, `SYSCON_MCLKSRC_DISABLED` = 7 }
- *MCLK Clock sources.*
- enum `CHIP_SYSCON_MAIN_B_CLKSRC_T` { `SYSCON_MAIN_B_CLKSRC_MAINCLKSELA` = 0, `SYSC_↵`  
`ON_MAIN_B_CLKSRC_PLL` = 2, `SYSCON_MAIN_B_CLKSRC_RTC` }
- enum `CHIP_SYSCON_CLKOUTSRC_T` {  
`SYSCON_CLKOUTSRC_MAINCLK` = 0, `SYSCON_CLKOUTSRC_CLKIN`, `SYSCON_CLKOUTSRC_WD_↵`  
`TOSC`, `SYSCON_CLKOUTSRC_FROHF`,  
`SYSCON_CLKOUTSRC_PLL`, `SYSCON_CLKOUTSRC_FRO12MHZ`, `SYSCON_CLKOUTSRC_RTC`, `SY_↵`  
`SCON_CLKOUTSRC_DISABLED` }
- enum `CHIP_SYSCON_CLOCK_T` {  
`SYSCON_CLOCK_ROM` = 1, `SYSCON_CLOCK_SRAM1` = 3, `SYSCON_CLOCK_SRAM2`, `SYSCON_CL_↵`  
`OCK_SRAMX`,  
`SYSCON_CLOCK_FLASH` = 7, `SYSCON_CLOCK_FMC`, `SYSCON_CLOCK_SPIFI` = 10, `SYSCON_CLO_↵`  
`CK_INPUTMUX`,  
`SYSCON_CLOCK_IOCON` = 13, `SYSCON_CLOCK_GPIO0`, `SYSCON_CLOCK_GPIO1`, `SYSCON_CLO_↵`  
`CK_PINT` = 18,  
`SYSCON_CLOCK_GINT`, `SYSCON_CLOCK_DMA`, `SYSCON_CLOCK_CRC`, `SYSCON_CLOCK_WWDT`,  
`SYSCON_CLOCK_RTC`, `SYSCON_CLOCK_MAILBOX` = 26, `SYSCON_CLOCK_ADC0`, `SYSCON_CLO_↵`  
`CK_MRT` = 32,  
`SYSCON_CLOCK_SCT0` = 32 + 2, `SYSCON_CLOCK_UTICK` = 32 + 10, `SYSCON_CLOCK_FLEXCOMM0`,  
`SYSCON_CLOCK_FLEXCOMM1`,  
`SYSCON_CLOCK_FLEXCOMM2`, `SYSCON_CLOCK_FLEXCOMM3`, `SYSCON_CLOCK_FLEXCOMM4`,

```

SYSCON_CLOCK_FLEXCOMM5,
SYSCON_CLOCK_FLEXCOMM6, SYSCON_CLOCK_FLEXCOMM7, SYSCON_CLOCK_DMIC, SYSCON_
N_CLOCK_TIMER2 = 32 + 22,
SYSCON_CLOCK_USB = 32 + 25, SYSCON_CLOCK_TIMER0, SYSCON_CLOCK_TIMER1, SYSCON_
CLOCK_TIMER3 = 128 + 13,
SYSCON_CLOCK_TIMER4 }
• enum CHIP_SYSCON_FLEXCOMMCLKSELSRC_T {
SYSCON_FLEXCOMMCLKSELSRC_FRO12MHZ = 0, SYSCON_FLEXCOMMCLKSELSRC_FROHF, SY
SCON_FLEXCOMMCLKSELSRC_PLL, SYSCON_FLEXCOMMCLKSELSRC_MCLK,
SYSCON_FLEXCOMMCLKSELSRC_FRG, SYSCON_FLEXCOMMCLKSELSRC_NONE = 7 }
• enum CHIP_SYSCON_ADCCLKSELSRC_T { SYSCON_ADCCLKSELSRC_MAINCLK = 0, SYSCON_AD
CCLKSELSRC_SYSPLOUT, SYSCON_ADCCLKSELSRC_FROHF }
• enum CHIP_ASYNC_SYSCON_SRC_T { SYSCON_ASYNC_MAINCLK = 0, SYSCON_ASYNC_FRO12M
HZ }
• enum CHIP_SYSCON_MAINCLKSRC_T {
SYSCON_MAINCLKSRC_FRO12MHZ = 0, SYSCON_MAINCLKSRC_CLKIN, SYSCON_MAINCLKSRC_
WDTOSC, SYSCON_MAINCLKSRC_FROHF,
SYSCON_MAINCLKSRC_PLOUT = 6, SYSCON_MAINCLKSRC_RTC }
• enum CHIP_SYSCON_FRGCLKSRC_T {
SYSCON_FRGCLKSRC_MAINCLK, SYSCON_FRGCLKSRC_PLL, SYSCON_FRGCLKSRC_FRO12MH
Z, SYSCON_FRGCLKSRC_FROHF,
SYSCON_FRGCLKSRC_NONE = 7 }

```

*Fractional Divider clock sources.*

## Functions

- `__STATIC_INLINE uint32_t Chip_Clock_GetExtClockInRate (void)`  
*Returns the external clock input rate.*
- `__STATIC_INLINE uint32_t Chip_Clock_GetRTCOscRate (void)`  
*Returns the RTC clock rate.*
- `__STATIC_INLINE void Chip_Clock_SetWDTOSCRate (WDT_OSC_FREQ_T freq, uint32_t div)`  
*Set the WDT Oscillator frequency and divider.*
- `uint32_t Chip_Clock_GetWDTOSCRate (void)`  
*Return estimated watchdog oscillator rate.*
- `__STATIC_INLINE uint32_t Chip_Clock_GetFROHFRate (void)`  
*Gets the HF-FRO Frequency rate.*
- `__STATIC_INLINE void Chip_Clock_SetMain_A_ClockSource (CHIP_SYSCON_MAIN_A_CLKSRC_T src)`  
*Set main A system clock source.*
- `__STATIC_INLINE void Chip_Clock_SetUSBClockSource (CHIP_SYSCON_USBCLKSRC_T src, uint32_t div)`  
*Set USB clock source.*
- `__STATIC_INLINE CHIP_SYSCON_USBCLKSRC_T Chip_Clock_GetUSBClockSource (void)`  
*Gets the clock source used by USB.*
- `__STATIC_INLINE uint32_t Chip_Clock_GetUSBClockDiv (void)`  
*Gets the clock divider used by USB.*
- `__STATIC_INLINE void Chip_Clock_SetMCLKClockSource (CHIP_SYSCON_MCLKSRC_T src, uint32_t div)`  
*Set the MCLK clock source.*
- `__STATIC_INLINE uint32_t Chip_Clock_GetMCLKDiv (void)`  
*Get MCLK clock div.*
- `__STATIC_INLINE CHIP_SYSCON_MCLKSRC_T Chip_Clock_GetMCLKSource (void)`  
*Get MCLK clock source.*
- `__STATIC_INLINE void Chip_Clock_SetMCLKDirInput (void)`

- Set MCLK pin direction to INPUT.*

  - `__STATIC_INLINE void Chip_Clock_SetMCLKDirOutput (void)`
- Set MCLK pin direction to OUTPUT.*

  - `__STATIC_INLINE void Chip_Clock_SetMCLKDir (int dir)`
- Set MCLK pin direction to INPUT or OUTPUT.*

  - `__STATIC_INLINE int Chip_Clock_GetMCLKDir (void)`
- Returns the main A clock source.*

  - `__STATIC_INLINE CHIP_SYSCON_MAIN_A_CLKSRC_T Chip_Clock_GetMain_A_ClockSource (void)`
- Return main A clock rate.*

  - `uint32_t Chip_Clock_GetMain_A_ClockRate (void)`
- Set main B system clock source.*

  - `__STATIC_INLINE void Chip_Clock_SetMain_B_ClockSource (CHIP_SYSCON_MAIN_B_CLKSRC_T src)`
- Returns the main B clock source.*

  - `__STATIC_INLINE CHIP_SYSCON_MAIN_B_CLKSRC_T Chip_Clock_GetMain_B_ClockSource (void)`
- Return main B clock rate.*

  - `uint32_t Chip_Clock_GetMain_B_ClockRate (void)`
- Set CLKOUT clock source and divider.*

  - `__STATIC_INLINE void Chip_Clock_SetCLKOUTSource (CHIP_SYSCON_CLKOUTSRC_T src, uint32_t div)`
- Get CLKOUT clock source.*

  - `__STATIC_INLINE CHIP_SYSCON_CLKOUTSRC_T Chip_Clock_GetCLKOUTSource (void)`
- Get CLKOUT clock divider.*

  - `__STATIC_INLINE uint32_t Chip_Clock_GetCLKOUTDiv (void)`
- Enable a system or peripheral clock.*

  - `void Chip_Clock_EnablePeriphClock (CHIP_SYSCON_CLOCK_T clk)`
- Disable a system or peripheral clock.*

  - `void Chip_Clock_DisablePeriphClock (CHIP_SYSCON_CLOCK_T clk)`
- Set system tick clock divider (external CLKIN as SYSTICK reference only)*

  - `__STATIC_INLINE void Chip_Clock_SetSysTickClockDiv (uint32_t div)`
- Returns system tick clock divider.*

  - `__STATIC_INLINE uint32_t Chip_Clock_GetSysTickClockDiv (void)`
- Returns the system tick rate as used with the system tick divider.*

  - `uint32_t Chip_Clock_GetSysTickClockRate (void)`
- Set system clock divider.*

  - `__STATIC_INLINE void Chip_Clock_SetSysClockDiv (uint32_t div)`
- Get system clock divider.*

  - `__STATIC_INLINE uint32_t Chip_Clock_GetSysClockDiv (void)`
- Set system tick clock divider.*

  - `__STATIC_INLINE void Chip_Clock_SetADCClockDiv (uint32_t div)`
- Returns ADC clock divider.*

  - `__STATIC_INLINE uint32_t Chip_Clock_GetADCClockDiv (void)`
- Set the FLEXCOMM clock source.*

  - `__STATIC_INLINE void Chip_Clock_SetFLEXCOMMClockSource (uint32_t idx, CHIP_SYSCON_FLEXCOMMCLKSELSRC_T src)`
- Returns the FLEXCOMM clock source.*

  - `__STATIC_INLINE CHIP_SYSCON_FLEXCOMMCLKSELSRC_T Chip_Clock_GetFLEXCOMMClockSource (uint32_t idx)`
- Return FlexCOMM clock rate.*

  - `uint32_t Chip_Clock_GetFLEXCOMMClockRate (uint32_t id)`
- Set the ADC clock source.*

  - `__STATIC_INLINE void Chip_Clock_SetADCClockSource (CHIP_SYSCON_ADCCLKSELSRC_T src)`

- `__STATIC_INLINE CHIP_SYSCON_ADCCLKSELSRC_T Chip_Clock_GetADCClockSource (void)`  
*Returns the ADC clock source.*
- `uint32_t Chip_Clock_GetADCClockRate (void)`  
*Return ADC clock rate.*
- `__STATIC_INLINE void Chip_Clock_EnableRTCOsc (void)`  
*Enable the RTC 32KHz output.*
- `__STATIC_INLINE void Chip_Clock_DisableRTCOsc (void)`  
*Disable the RTC 32KHz output.*
- `__STATIC_INLINE bool Chip_Clock_GetRTCOsc (void)`
- `__STATIC_INLINE void Chip_Clock_SetAsyncSysconClockSource (CHIP_ASYNC_SYSCON_SRC_T src)`  
*Set asynchronous APB clock source.*
- `__STATIC_INLINE CHIP_ASYNC_SYSCON_SRC_T Chip_Clock_GetAsyncSysconClockSource (void)`  
*Get asynchronous APB clock source.*
- `uint32_t Chip_Clock_GetAsyncSyscon_ClockRate (void)`  
*Return asynchronous APB clock rate.*
- `__STATIC_INLINE void Chip_Clock_SetMainClockSource (CHIP_SYSCON_MAINCLKSRC_T src)`  
*Set main system clock source.*
- `CHIP_SYSCON_MAINCLKSRC_T Chip_Clock_GetMainClockSource (void)`  
*Get main system clock source.*
- `uint32_t Chip_Clock_GetMainClockRate (void)`  
*Return main clock rate.*
- `uint32_t Chip_Clock_GetSystemClockRate (void)`  
*Return system clock rate.*
- `uint32_t Chip_Clock_GetFRGInClockRate (void)`  
*Get the input clock frequency of FRG.*
- `__STATIC_INLINE void Chip_Clock_SetFRGClockSource (CHIP_SYSCON_FRGCLKSRC_T src)`  
*Set clock source used by FRG.*
- `__STATIC_INLINE CHIP_SYSCON_FRGCLKSRC_T Chip_Clock_GetFRGClockSource (void)`  
*Get clock source used by FRG.*
- `uint32_t Chip_Clock_GetFRGClockRate (void)`  
*Get Fraction Rate Generator (FRG) clock rate.*
- `uint32_t Chip_Clock_SetFRGClockRate (uint32_t rate)`  
*Set FRG rate to given rate.*

## 6.12.2 Macro Definition Documentation

### 6.12.2.1 `#define Chip_Clock_GetIntOscRate( ) SYSCON_FRO12MHZ_FREQ`

Definition at line 54 of file clock\_5411x.h.

### 6.12.2.2 `#define SYSCON_FRO12MHZ_FREQ (12000000)`

Definition at line 47 of file clock\_5411x.h.

### 6.12.2.3 `#define SYSCON_FRO48MHZ_FREQ (48000000)`

Definition at line 48 of file clock\_5411x.h.

### 6.12.2.4 `#define SYSCON_FRO96MHZ_FREQ (96000000)`

Definition at line 49 of file clock\_5411x.h.

#### 6.12.2.5 `#define SYSCON_RTC_FREQ (32768)`

Definition at line 51 of file clock\_5411x.h.

#### 6.12.2.6 `#define SYSCON_WDTOSC_FREQ (500000)`

Definition at line 50 of file clock\_5411x.h.

### 6.12.3 Enumeration Type Documentation

#### 6.12.3.1 `enum CHIP_ASYNC_SYSCON_SRC_T`

Clock source selections for the asynchronous APB clock

Enumerator

**`SYSCON_ASYNC_MAINCLK`** Main System clock

**`SYSCON_ASYNC_FRO12MHZ`** 12MHz FRO

Definition at line 621 of file clock\_5411x.h.

#### 6.12.3.2 `enum CHIP_SYSCON_ADCCLKSELSRC_T`

Clock sources for ADC clock source select

Enumerator

**`SYSCON_ADCCLKSELSRC_MAINCLK`** Main clock

**`SYSCON_ADCCLKSELSRC_SYSPLLOUT`** PLL output

**`SYSCON_ADCCLKSELSRC_FROHF`** High frequency FRO 48MHz or 96MHz

Definition at line 562 of file clock\_5411x.h.

#### 6.12.3.3 `enum CHIP_SYSCON_CLKOUTSRC_T`

Clock sources for CLKOUT

Enumerator

**`SYSCON_CLKOUTSRC_MAINCLK`** Main system clock for CLKOUT

**`SYSCON_CLKOUTSRC_CLKIN`** CLKIN for CLKOUT

**`SYSCON_CLKOUTSRC_WDTOSC`** Watchdog oscillator for CLKOUT

**`SYSCON_CLKOUTSRC_FROHF`** 48MHz or 96MHz FRO

**`SYSCON_CLKOUTSRC_PLL`** Output of the PLL

**`SYSCON_CLKOUTSRC_FRO12MHZ`** 12MHz FRO

**`SYSCON_CLKOUTSRC_RTC`** RTC oscillator 32KHz for CLKOUT

**`SYSCON_CLKOUTSRC_DISABLED`** Disable clock source for CLKOUT

Definition at line 337 of file clock\_5411x.h.

## 6.12.3.4 enum CHIP\_SYSCON\_CLOCK\_T

System and peripheral clocks enum

Enumerator

***SYSCON\_CLOCK\_ROM*** ROM clock  
***SYSCON\_CLOCK\_SRAM1*** SRAM1 clock  
***SYSCON\_CLOCK\_SRAM2*** SRAM2 clock  
***SYSCON\_CLOCK\_SRAMX*** SRAMX Clock  
***SYSCON\_CLOCK\_FLASH*** FLASH controller clock  
***SYSCON\_CLOCK\_FMC*** FMC clock  
***SYSCON\_CLOCK\_SPIFI*** SPIFI Clock  
***SYSCON\_CLOCK\_INPUTMUX*** Input mux clock  
***SYSCON\_CLOCK\_IOCON*** IOCON clock  
***SYSCON\_CLOCK\_GPIO0*** GPIO0 clock  
***SYSCON\_CLOCK\_GPIO1*** GPIO1 clock  
***SYSCON\_CLOCK\_PINT*** PININT clock  
***SYSCON\_CLOCK\_GINT*** grouped pin interrupt block clock  
***SYSCON\_CLOCK\_DMA*** DMA clock  
***SYSCON\_CLOCK\_CRC*** CRC clock  
***SYSCON\_CLOCK\_WWDT*** WDT clock  
***SYSCON\_CLOCK\_RTC*** RTC clock  
***SYSCON\_CLOCK\_MAILBOX*** Mailbox clock  
***SYSCON\_CLOCK\_ADC0*** ADC0 clock  
***SYSCON\_CLOCK\_MRT*** multi-rate timer clock  
***SYSCON\_CLOCK\_SCT0*** SCT0 clock  
***SYSCON\_CLOCK\_UTICK*** UTICK clock  
***SYSCON\_CLOCK\_FLEXCOMM0*** FLEXCOMM0 Clock  
***SYSCON\_CLOCK\_FLEXCOMM1*** FLEXCOMM1 Clock  
***SYSCON\_CLOCK\_FLEXCOMM2*** FLEXCOMM2 Clock  
***SYSCON\_CLOCK\_FLEXCOMM3*** FLEXCOMM3 Clock  
***SYSCON\_CLOCK\_FLEXCOMM4*** FLEXCOMM4 Clock  
***SYSCON\_CLOCK\_FLEXCOMM5*** FLEXCOMM5 Clock  
***SYSCON\_CLOCK\_FLEXCOMM6*** FLEXCOMM6 Clock  
***SYSCON\_CLOCK\_FLEXCOMM7*** FLEXCOMM7 Clock  
***SYSCON\_CLOCK\_DMIC*** D-MIC Clock  
***SYSCON\_CLOCK\_TIMER2*** TIMER2 clock  
***SYSCON\_CLOCK\_USB*** USB clock  
***SYSCON\_CLOCK\_TIMER0*** TIMER0 clock  
***SYSCON\_CLOCK\_TIMER1*** TIMER1 Clock  
***SYSCON\_CLOCK\_TIMER3*** TIMER3 clock  
***SYSCON\_CLOCK\_TIMER4*** TIMER4 clock

Definition at line 383 of file clock\_5411x.h.

#### 6.12.3.5 enum CHIP\_SYSCON\_FLEXCOMMCLKSELSRC\_T

Clock sources for FLEXCOMM clock source select

Enumerator

***SYSCON\_FLEXCOMMCLKSELSRC\_FRO12MHZ*** FRO 12-MHz  
***SYSCON\_FLEXCOMMCLKSELSRC\_FROHF*** HF-FRO 48-MHz or 96-MHz  
***SYSCON\_FLEXCOMMCLKSELSRC\_PLL*** PLL output  
***SYSCON\_FLEXCOMMCLKSELSRC\_MCLK*** MCLK output  
***SYSCON\_FLEXCOMMCLKSELSRC\_FRG*** FRG output  
***SYSCON\_FLEXCOMMCLKSELSRC\_NONE*** NONE output

Definition at line 522 of file clock\_5411x.h.

#### 6.12.3.6 enum CHIP\_SYSCON\_FRGCLKSRC\_T

Fractional Divider clock sources.

Enumerator

***SYSCON\_FRGCLKSRC\_MAINCLK*** Main Clock  
***SYSCON\_FRGCLKSRC\_PLL*** Output clock from PLL  
***SYSCON\_FRGCLKSRC\_FRO12MHZ*** FRO 12-MHz  
***SYSCON\_FRGCLKSRC\_FROHF*** FRO High Frequency (48 or 96) MHz  
***SYSCON\_FRGCLKSRC\_NONE*** No clock input

Definition at line 708 of file clock\_5411x.h.

#### 6.12.3.7 enum CHIP\_SYSCON\_MAIN\_A\_CLKSRC\_T

Clock source selections for only the main A system clock. The main A system clock is used as an input into the main B system clock selector. Main clock A only needs to be setup if the main clock A input is used in the main clock system selector.

Enumerator

***SYSCON\_MAIN\_A\_CLKSRC\_FRO12MHZ*** 12MHz FRO  
***SYSCON\_MAIN\_A\_CLKSRC\_CLKIN*** Crystal (main) oscillator in  
***SYSCON\_MAIN\_A\_CLKSRC\_WDTOSC*** Watchdog oscillator rate  
***SYSCON\_MAIN\_A\_CLKSRC\_FROHF*** 48MHz or 96MHz HF-FRO

Definition at line 136 of file clock\_5411x.h.

#### 6.12.3.8 enum CHIP\_SYSCON\_MAIN\_B\_CLKSRC\_T

Clock sources for only main B system clock

Enumerator

***SYSCON\_MAIN\_B\_CLKSRC\_MAINCLKSELA*** main clock A  
***SYSCON\_MAIN\_B\_CLKSRC\_PLL*** System PLL output  
***SYSCON\_MAIN\_B\_CLKSRC\_RTC*** RTC oscillator 32KHz output

Definition at line 303 of file clock\_5411x.h.



## 6.12.3.9 enum CHIP\_SYSCON\_MAINCLKSRC\_T

Clock sources for main system clock. This is a mix of both main clock A and B selections.

## Enumerator

***SYSCON\_MAINCLKSRC\_FRO12MHZ*** 12-MHz FRO  
***SYSCON\_MAINCLKSRC\_CLKIN*** Crystal (main) oscillator in  
***SYSCON\_MAINCLKSRC\_WDTOSC*** Watchdog oscillator rate  
***SYSCON\_MAINCLKSRC\_FROHF*** 48MHz or 96-MHz HF-FRO  
***SYSCON\_MAINCLKSRC\_PLLOUT*** System PLL output  
***SYSCON\_MAINCLKSRC\_RTC*** RTC oscillator 32KHz output

Definition at line 656 of file clock\_5411x.h.

## 6.12.3.10 enum CHIP\_SYSCON\_MCLKSRC\_T

MCLK Clock sources.

## Enumerator

***SYSCON\_MCLKSRC\_FROHF*** HF-FRO 48MHz or 96MHz  
***SYSCON\_MCLKSRC\_PLL*** Main pll  
***SYSCON\_MCLKSRC\_MCLKIN*** MCLK INPUT Clock pin set by IOCON  
***SYSCON\_MCLKSRC\_DISABLED*** Disable clock source to MCLK

Definition at line 211 of file clock\_5411x.h.

## 6.12.3.11 enum CHIP\_SYSCON\_USBCLKSRC\_T

USB Clock source.

## Enumerator

***SYSCON\_USBCLKSRC\_FROHF*** High frequency FRO 48MHz or 96MHz  
***SYSCON\_USBCLKSRC\_PLL*** USB PLL  
***SYSCON\_USBCLKSRC\_DISABLED*** USB Clock disabled

Definition at line 171 of file clock\_5411x.h.

## 6.12.3.12 enum WDT\_OSC\_FREQ\_T

WDT Osc frequency value table.

## Enumerator

***WDT\_FREQ\_RESERVED*** Reserved value  
***WDT\_FREQ\_400000*** WDT Freq 400 KHz  
***WDT\_FREQ\_600000*** WDT Freq 600 KHz  
***WDT\_FREQ\_750000*** WDT Freq 750 KHz  
***WDT\_FREQ\_900000*** WDT Freq 900 KHz  
***WDT\_FREQ\_1000000*** WDT Freq 1.0 MHz  
***WDT\_FREQ\_1200000*** WDT Freq 1.2 MHz

**WDT\_FREQ\_1300000** WDT Freq 1.3 MHz  
**WDT\_FREQ\_1400000** WDT Freq 1.4 MHz  
**WDT\_FREQ\_1500000** WDT Freq 1.5 MHz  
**WDT\_FREQ\_1600000** WDT Freq 1.6 MHz  
**WDT\_FREQ\_1700000** WDT Freq 1.7 MHz  
**WDT\_FREQ\_1800000** WDT Freq 1.8 MHz  
**WDT\_FREQ\_1900000** WDT Freq 1.9 MHz  
**WDT\_FREQ\_2000000** WDT Freq 2.0 MHz  
**WDT\_FREQ\_2050000** WDT Freq 2.05 MHz  
**WDT\_FREQ\_2100000** WDT Freq 2.1 MHz  
**WDT\_FREQ\_2200000** WDT Freq 2.2 MHz  
**WDT\_FREQ\_2250000** WDT Freq 2.25 MHz  
**WDT\_FREQ\_2300000** WDT Freq 2.3 MHz  
**WDT\_FREQ\_2400000** WDT Freq 2.4 MHz  
**WDT\_FREQ\_2450000** WDT Freq 2.45 MHz  
**WDT\_FREQ\_2500000** WDT Freq 2.5 MHz  
**WDT\_FREQ\_2600000** WDT Freq 2.6 MHz  
**WDT\_FREQ\_2650000** WDT Freq 2.65 MHz  
**WDT\_FREQ\_2700000** WDT Freq 2.7 MHz  
**WDT\_FREQ\_2800000** WDT Freq 2.8 MHz  
**WDT\_FREQ\_2850000** WDT Freq 2.85 MHz  
**WDT\_FREQ\_2900000** WDT Freq 2.9 MHz  
**WDT\_FREQ\_2950000** WDT Freq 2.95 MHz  
**WDT\_FREQ\_3000000** WDT Freq 3.0 MHz  
**WDT\_FREQ\_3050000** WDT Freq 3.05 MHz

Definition at line 75 of file clock\_5411x.h.

## 6.12.4 Function Documentation

### 6.12.4.1 void Chip\_Clock\_DisablePeriphClock ( **CHIP\_SYSCON\_CLOCK\_T** *clk* )

Disable a system or peripheral clock.

Parameters

<i>clk</i>	: Clock to disable
------------	--------------------

Returns

Nothing

Definition at line 151 of file clock\_5411x.c.

### 6.12.4.2 \_\_STATIC\_INLINE void Chip\_Clock\_DisableRTCOsc ( void )

Disable the RTC 32KHz output.

Returns

Nothing

Definition at line 608 of file clock\_5411x.h.

6.12.4.3 void Chip\_Clock\_EnablePeriphClock ( CHIP\_SYSCON\_CLOCK\_T *clk* )

Enable a system or peripheral clock.

**Parameters**

<i>clk</i>	: Clock to enable
------------	-------------------

**Returns**

Nothing

Definition at line 135 of file clock\_5411x.c.

**6.12.4.4 `__STATIC_INLINE void Chip_Clock_EnableRTCOsc ( void )`**

Enable the RTC 32KHz output.

**Returns**

Nothing

**Note**

This clock can be used for the main clock directly, but do not use this clock with the system PLL.

Definition at line 599 of file clock\_5411x.h.

**6.12.4.5 `__STATIC_INLINE uint32_t Chip_Clock_GetADCClockDiv ( void )`**

Returns ADC clock divider.

**Returns**

ADC clock divider, 0 = disabled

Definition at line 514 of file clock\_5411x.h.

**6.12.4.6 `uint32_t Chip_Clock_GetADCClockRate ( void )`**

Return ADC clock rate.

**Returns**

ADC clock rate

Definition at line 185 of file clock\_5411x.c.

**6.12.4.7 `__STATIC_INLINE CHIP_SYSCON_ADCCLKSELSRC_T Chip_Clock_GetADCClockSource ( void )`**

Returns the ADC clock source.

**Returns**

Returns which clock is used for the ADC clock source

Definition at line 582 of file clock\_5411x.h.

#### 6.12.4.8 `uint32_t Chip_Clock_GetAsyncSyscon_ClockRate ( void )`

Return asynchronous APB clock rate.

##### Returns

Asynchronous APB clock rate

##### Note

Includes adjustments by Async clock divider (ASYNCCLKDIV).

Definition at line 214 of file `clock_5411x.c`.

#### 6.12.4.9 `__STATIC_INLINE CHIP_ASYNC_SYSCON_SRC_T Chip_Clock_GetAsyncSysconClockSource ( void )`

Get asynchronous APB clock source.

##### Returns

Clock source for asynchronous APB clock

Definition at line 640 of file `clock_5411x.h`.

#### 6.12.4.10 `__STATIC_INLINE uint32_t Chip_Clock_GetCLKOUTDiv ( void )`

Get CLKOUT clock divider.

##### Returns

Clock output source

Definition at line 375 of file `clock_5411x.h`.

#### 6.12.4.11 `__STATIC_INLINE CHIP_SYSCON_CLKOUTSRC_T Chip_Clock_GetCLKOUTSource ( void )`

Get CLKOUT clock source.

##### Returns

Clock output source

Definition at line 366 of file `clock_5411x.h`.

#### 6.12.4.12 `__STATIC_INLINE uint32_t Chip_Clock_GetExtClockInRate ( void )`

Returns the external clock input rate.

##### Returns

External clock input rate

Definition at line 60 of file `clock_5411x.h`.

#### 6.12.4.13 `uint32_t Chip_Clock_GetFLEXCOMMClockRate ( uint32_t id )`

Return FlexCOMM clock rate.

**Parameters**

<i>id</i>	: FlexCOMM ID (Valid range: 0 to 7)
-----------	-------------------------------------

**Returns**

FlexCOMM clock rate

Definition at line 376 of file clock\_5411x.c.

6.12.4.14 `__STATIC_INLINE CHIP_SYSCON_FLEXCOMMCLKSELSRC_T Chip_Clock_GetFLEXCOMMClockSource (uint32_t idx )`

Returns the FLEXCOMM clock source.

**Parameters**

<i>idx</i>	: Index of the flexcomm (0 to 7)
------------	----------------------------------

**Returns**

Returns which clock is used for the FLEXCOMM clock source

Definition at line 547 of file clock\_5411x.h.

6.12.4.15 `uint32_t Chip_Clock_GetFRGClockRate ( void )`

Get Fraction Rate Generator (FRG) clock rate.

**Returns**

UART base clock rate

Definition at line 268 of file clock\_5411x.c.

6.12.4.16 `__STATIC_INLINE CHIP_SYSCON_FRGCLKSRC_T Chip_Clock_GetFRGClockSource ( void )`

Get clock source used by FRG.

**Returns**

Clock source used by FRG

**Note**

Definition at line 737 of file clock\_5411x.h.

6.12.4.17 `uint32_t Chip_Clock_GetFRGInClockRate ( void )`

Get the input clock frequency of FRG.

**Returns**

Frequency in Hz on success (0 on failure)

Definition at line 292 of file clock\_5411x.c.

**6.12.4.18** `__STATIC_INLINE uint32_t Chip_Clock_GetFROHFRate ( void )`

Gets the HF-FRO Frequency rate.

**Returns**

Nothing

Definition at line 147 of file clock\_5411x.h.

**6.12.4.19** `uint32_t Chip_Clock_GetMain_A_ClockRate ( void )`

Return main A clock rate.

**Returns**

main A clock rate in Hz

Definition at line 83 of file clock\_5411x.c.

**6.12.4.20** `__STATIC_INLINE CHIP_SYSCON_MAIN_A_CLKSRC_T Chip_Clock_GetMain_A_ClockSource ( void )`

Returns the main A clock source.

**Returns**

Returns which clock is used for the main A

Definition at line 289 of file clock\_5411x.h.

**6.12.4.21** `uint32_t Chip_Clock_GetMain_B_ClockRate ( void )`

Return main B clock rate.

**Returns**

main B clock rate

Definition at line 113 of file clock\_5411x.c.

**6.12.4.22** `__STATIC_INLINE CHIP_SYSCON_MAIN_B_CLKSRC_T Chip_Clock_GetMain_B_ClockSource ( void )`

Returns the main B clock source.

**Returns**

Returns which clock is used for the main B

Definition at line 323 of file clock\_5411x.h.

**6.12.4.23** `uint32_t Chip_Clock_GetMainClockRate ( void )`

Return main clock rate.

**Returns**

main clock rate

Definition at line 244 of file clock\_5411x.c.

#### 6.12.4.24 **CHIP\_SYSCON\_MAINCLKSRC\_T** Chip\_Clock\_GetMainClockSource ( void )

Get main system clock source.

##### Returns

Clock source for main system

##### Note

Definition at line 224 of file clock\_5411x.c.

#### 6.12.4.25 **\_\_STATIC\_INLINE int** Chip\_Clock\_GetMCLKDir ( void )

Definition at line 280 of file clock\_5411x.h.

#### 6.12.4.26 **\_\_STATIC\_INLINE uint32\_t** Chip\_Clock\_GetMCLKDiv ( void )

Get MCLK clock div.

##### Returns

MCLK divider

Definition at line 234 of file clock\_5411x.h.

#### 6.12.4.27 **\_\_STATIC\_INLINE CHIP\_SYSCON\_MCLKSRC\_T** Chip\_Clock\_GetMCLKSource ( void )

Get MCLK clock source.

##### Returns

MCLK clock source

Definition at line 243 of file clock\_5411x.h.

#### 6.12.4.28 **\_\_STATIC\_INLINE bool** Chip\_Clock\_GetRTCOsc ( void )

Definition at line 613 of file clock\_5411x.h.

#### 6.12.4.29 **\_\_STATIC\_INLINE uint32\_t** Chip\_Clock\_GetRTCOscRate ( void )

Returns the RTC clock rate.

##### Returns

RTC oscillator clock rate in Hz

Definition at line 69 of file clock\_5411x.h.

#### 6.12.4.30 **\_\_STATIC\_INLINE uint32\_t** Chip\_Clock\_GetSysClockDiv ( void )

Get system clock divider.



**Parameters**

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

**Returns**

System clock divider

Definition at line 490 of file clock\_5411x.h.

**6.12.4.31 uint32\_t Chip\_Clock\_GetSystemClockRate ( void )**

Return system clock rate.

**Returns**

system clock rate

**Note**

This is the main clock rate divided by AHBCLKDIV.

Definition at line 261 of file clock\_5411x.c.

**6.12.4.32 \_\_STATIC\_INLINE uint32\_t Chip\_Clock\_GetSysTickClockDiv ( void )**

Returns system tick clock divider.

**Returns**

system tick clock divider

Definition at line 462 of file clock\_5411x.h.

**6.12.4.33 uint32\_t Chip\_Clock\_GetSysTickClockRate ( void )**

Returns the system tick rate as used with the system tick divider.

**Returns**

the system tick rate

Definition at line 167 of file clock\_5411x.c.

**6.12.4.34 \_\_STATIC\_INLINE uint32\_t Chip\_Clock\_GetUSBClockDiv ( void )**

Gets the clock divider used by USB.

**Returns**

Returns clock divider used by USB

Definition at line 203 of file clock\_5411x.h.

#### 6.12.4.35 `__STATIC_INLINE CHIP_SYSCON_USBCLKSRC_T Chip_Clock_GetUSBClockSource ( void )`

Gets the clock source used by USB.

##### Returns

Returns which clock is used for USB

Definition at line 194 of file clock\_5411x.h.

#### 6.12.4.36 `uint32_t Chip_Clock_GetWDTOSCRate ( void )`

Return estimated watchdog oscillator rate.

##### Returns

Estimated watchdog oscillator rate

##### Note

This rate is accurate to plus or minus 40%.

Definition at line 411 of file clock\_5411x.c.

#### 6.12.4.37 `__STATIC_INLINE void Chip_Clock_SetADCClockDiv ( uint32_t div )`

Set system tick clock divider.

##### Parameters

<i>div</i>	: divider for system clock
------------	----------------------------

##### Returns

Nothing

##### Note

Use 0 to disable, or a divider value of 1 to 255. The system tick rate is the main system clock divided by this value. Use caution when using the CMSIS [SysTick\\_Config\(\)](#) functions as they typically use SystemCoreClock for setup.

Definition at line 504 of file clock\_5411x.h.

#### 6.12.4.38 `__STATIC_INLINE void Chip_Clock_SetADCClockSource ( CHIP_SYSCON_ADCCLKSELSRC_T src )`

Set the ADC clock source.

##### Parameters

<i>src</i>	: ADC clock source
------------	--------------------

##### Returns

Nothing

Definition at line 573 of file clock\_5411x.h.

#### 6.12.4.39 `__STATIC_INLINE void Chip_Clock_SetAsyncSysconClockSource ( CHIP_ASYNC_SYSCON_SRC_T src )`

Set asynchronous APB clock source.

## Parameters

<i>src</i>	: Clock source for asynchronous APB clock
------------	---

## Returns

Nothing

Definition at line 631 of file clock\_5411x.h.

6.12.4.40 `__STATIC_INLINE void Chip_Clock_SetCLKOUTSource ( CHIP_SYSCON_CLKOUTSRC_T src, uint32_t div )`

Set CLKOUT clock source and divider.

## Parameters

<i>src</i>	: Clock source for CLKOUT (see <a href="#">CHIP_SYSCON_CLKOUTSRC_T</a> )
<i>div</i>	: divider for CLKOUT clock [Valid range 1 to 256]

## Returns

Nothing

## Note

The CLKOUT clock rate is the clock source divided by the divider.

Definition at line 355 of file clock\_5411x.h.

6.12.4.41 `__STATIC_INLINE void Chip_Clock_SetFLEXCOMMClockSource ( uint32_t idx, CHIP_SYSCON_FLEXCOMMCLKSELSRC_T src )`

Set the FLEXCOMM clock source.

## Parameters

<i>idx</i>	: Index of the flexcomm (0 to 7)
<i>src</i>	: FLEXCOMM clock source (See <a href="#">CHIP_SYSCON_FLEXCOMMCLKSELSRC_T</a> )

## Returns

Nothing

Definition at line 537 of file clock\_5411x.h.

6.12.4.42 `uint32_t Chip_Clock_SetFRGClockRate ( uint32_t rate )`

Set FRG rate to given rate.

## Returns

Actual FRG clock rate

## Note

If FRG is used for UART base clock, *rate* is recommended to be 16 times desired baud rate; **This API must only be called after setting the source using [Chip\\_Clock\\_SetFRGClockSource\(\)](#)**

Definition at line 322 of file clock\_5411x.c.

6.12.4.43 `__STATIC_INLINE void Chip_Clock_SetFRGClockSource ( CHIP_SYSCON_FRGCLKSRC_T src )`

Set clock source used by FRG.

#### Returns

Clock source used by FRG

#### Note

Definition at line 727 of file clock\_5411x.h.

6.12.4.44 `__STATIC_INLINE void Chip_Clock_SetMain_A_ClockSource ( CHIP_SYSCON_MAIN_A_CLKSRC_T src )`

Set main A system clock source.

#### Parameters

<i>src</i>	: Clock source for main A
------------	---------------------------

#### Returns

Nothing

#### Note

This function only needs to be setup if main clock A will be selected in the [Chip\\_Clock\\_GetMain\\_B\\_ClockRate\(\)](#) function.

Definition at line 163 of file clock\_5411x.h.

6.12.4.45 `__STATIC_INLINE void Chip_Clock_SetMain_B_ClockSource ( CHIP_SYSCON_MAIN_B_CLKSRC_T src )`

Set main B system clock source.

#### Parameters

<i>src</i>	: Clock source for main B
------------	---------------------------

#### Returns

Nothing

Definition at line 314 of file clock\_5411x.h.

6.12.4.46 `__STATIC_INLINE void Chip_Clock_SetMainClockSource ( CHIP_SYSCON_MAINCLKSRC_T src )`

Set main system clock source.

#### Parameters

---

<i>src</i>	: Clock source for main system (See <a href="#">CHIP_SYSCON_MAINCLKSRC_T</a> )
------------	--

**Returns**

Nothing

Definition at line 670 of file clock\_5411x.h.

6.12.4.47 `__STATIC_INLINE void Chip_Clock_SetMCLKClockSource ( CHIP_SYSCON_MCLKSRC_T src, uint32_t div )`

Set the MCLK clock source.

**Parameters**

<i>src</i>	Clock Source for MCLK (see <a href="#">CHIP_SYSCON_MCLKSRC_T</a> )
<i>div</i>	Value by which the source clock rate be divided (must be greater than 0)

**Returns**

Nothing

Definition at line 224 of file clock\_5411x.h.

6.12.4.48 `__STATIC_INLINE void Chip_Clock_SetMCLKDir ( int dir )`

Set MCLK pin direction to INPUT or OUTPUT.

**Parameters**

<i>dir</i>	: 0 => INPUT, anything else => OUTPUT
------------	---------------------------------------

**Returns**

Nothing

Definition at line 271 of file clock\_5411x.h.

6.12.4.49 `__STATIC_INLINE void Chip_Clock_SetMCLKDirInput ( void )`

Set MCLK pin direction to INPUT.

**Returns**

Nothing

Definition at line 252 of file clock\_5411x.h.

6.12.4.50 `__STATIC_INLINE void Chip_Clock_SetMCLKDirOutput ( void )`

Set MCLK pin direction to OUTPUT.

**Returns**

Nothing

Definition at line 261 of file clock\_5411x.h.

6.12.4.51 `__STATIC_INLINE void Chip_Clock_SetSysClockDiv ( uint32_t div )`

Set system clock divider.

## Parameters

<i>div</i>	: divider for system clock
------------	----------------------------

## Returns

Nothing

## Note

Use a divider value of 1 to 255. The system clock rate is the main system clock divided by this value.

Definition at line 480 of file clock\_5411x.h.

#### 6.12.4.52 `__STATIC_INLINE void Chip_Clock_SetSysTickClockDiv ( uint32_t div )`

Set system tick clock divider (external CLKIN as SYSTICK reference only)

## Parameters

<i>div</i>	: divider for system clock
------------	----------------------------

## Returns

Nothing

## Note

Use 0 to disable, or a divider value of 1 to 255. The system tick rate is the external CLKIN rate divided by this value. The extern CLKIN pin signal, divided by the SYSTICKCLKDIV divider, is selected by clearing C<sub>↔</sub>LKSOURCE bit 2 in the System Tick CSR register. The core clock must be at least 2.5 times faster than the reference system tick clock otherwise the count values are unpredictable.

Definition at line 453 of file clock\_5411x.h.

#### 6.12.4.53 `__STATIC_INLINE void Chip_Clock_SetUSBClockSource ( CHIP_SYSCON_USBCLKSRC_T src, uint32_t div )`

Set USB clock source.

## Parameters

<i>src</i>	: Clock source for USB (See <a href="#">CHIP_SYSCON_USBCLKSRC_T</a> )
<i>div</i>	: Value by which the clock must be divided (valid range: 1 - 256)

## Returns

Nothing

Definition at line 183 of file clock\_5411x.h.

#### 6.12.4.54 `__STATIC_INLINE void Chip_Clock_SetWDTOSCRate ( WDT_OSC_FREQ_T freq, uint32_t div )`

Set the WDT Oscillator frequency and divider.

**Parameters**

<i>freq</i>	: WDT OSC Frequency to set [See <a href="#">WDT_OSC_FREQ_T</a> ]
<i>div</i>	: Divider value [Valid values are 2, 4, 6, 8 ... 64]

**Returns**

Nothing

**Note**

The actual frequency of the WDT Oscillator can be +/- 40% of frequency set in *freq*.

Definition at line 118 of file clock\_5411x.h.



## 6.13 CHIP: LPC5411X Cyclic Redundancy Check Engine driver

### 6.13.1 Detailed Description

#### Data Structures

- struct [LPC\\_CRC\\_T](#)  
*CRC register block structure.*

#### Macros

- #define [CRC\\_MODE\\_POLY\\_BITMASK](#) ((0x03)) /\*\* CRC polynomial Bit mask \*/
- #define [CRC\\_MODE\\_POLY\\_CCITT](#) (0x00) /\*\* Select CRC-CCITT polynomial \*/
- #define [CRC\\_MODE\\_POLY\\_CRC16](#) (0x01) /\*\* Select CRC-16 polynomial \*/
- #define [CRC\\_MODE\\_POLY\\_CRC32](#) (0x02) /\*\* Select CRC-32 polynomial \*/
- #define [CRC\\_MODE\\_WRDATA\\_BITMASK](#) (0x03 << 2) /\*\* CRC WR\_Data Config Bit mask \*/
- #define [CRC\\_MODE\\_WRDATA\\_BIT\\_RVS](#) (1 << 2) /\*\* Select Bit order reverse for WR\_DATA (per byte) \*/
- #define [CRC\\_MODE\\_WRDATA\\_CMPL](#) (1 << 3) /\*\* Select One's complement for WR\_DATA \*/
- #define [CRC\\_MODE\\_SUM\\_BITMASK](#) (0x03 << 4) /\*\* CRC Sum Config Bit mask \*/
- #define [CRC\\_MODE\\_SUM\\_BIT\\_RVS](#) (1 << 4) /\*\* Select Bit order reverse for CRC\_SUM \*/
- #define [CRC\\_MODE\\_SUM\\_CMPL](#) (1 << 5) /\*\* Select One's complement for CRC\_SUM \*/
- #define [MODE\\_CFG\\_CCITT](#) (0x00) /\*\* Pre-defined mode word for default CCITT setup \*/
- #define [MODE\\_CFG\\_CRC16](#) (0x15) /\*\* Pre-defined mode word for default CRC16 setup \*/
- #define [MODE\\_CFG\\_CRC32](#) (0x36) /\*\* Pre-defined mode word for default CRC32 setup \*/
- #define [CRC\\_SEED\\_CCITT](#) (0x0000FFFF) /\*\* Initial seed value for CCITT mode \*/
- #define [CRC\\_SEED\\_CRC16](#) (0x00000000) /\*\* Initial seed value for CRC16 mode \*/
- #define [CRC\\_SEED\\_CRC32](#) (0xFFFFFFFF) /\*\* Initial seed value for CRC32 mode \*/

#### Enumerations

- enum [CRC\\_POLY\\_T](#) { [CRC\\_POLY\\_CCITT](#) = CRC\_MODE\_POLY\_CCITT, [CRC\\_POLY\\_CRC16](#) = CRC\_MODE\_POLY\_CRC16, [CRC\\_POLY\\_CRC32](#) = CRC\_MODE\_POLY\_CRC32, [CRC\\_POLY\\_LAST](#) }
- CRC polynomial.*

#### Functions

- `__STATIC_INLINE void Chip\_CRC\_Init (LPC\_CRC\_T *pCRC)`  
*Initializes the CRC Engine.*
- `__STATIC_INLINE void Chip\_CRC\_Deinit (LPC\_CRC\_T *pCRC)`  
*Deinitializes the CRC Engine.*
- `__STATIC_INLINE void Chip\_CRC\_SetPoly (LPC\_CRC\_T *pCRC, CRC\_POLY\_T poly, uint32_t flags)`  
*Set the polynomial used for the CRC calculation.*
- `__STATIC_INLINE void Chip\_CRC\_UseCRC16 (LPC\_CRC\_T *pCRC)`  
*Sets up the CRC engine for CRC16 mode.*
- `__STATIC_INLINE void Chip\_CRC\_UseCRC32 (LPC\_CRC\_T *pCRC)`  
*Sets up the CRC engine for CRC32 mode.*
- `__STATIC_INLINE void Chip\_CRC\_UseCCITT (LPC\_CRC\_T *pCRC)`  
*Sets up the CRC engine for CCITT mode.*
- `__STATIC_INLINE void Chip\_CRC\_UseDefaultConfig (LPC\_CRC\_T *pCRC, CRC\_POLY\_T poly)`  
*Engage the CRC engine with defaults based on the polynomial to be used.*
- `__STATIC_INLINE void Chip\_CRC\_SetMode (LPC\_CRC\_T *pCRC, uint32_t mode)`

- Set the CRC Mode bits.*

  - `__STATIC_INLINE uint32_t Chip_CRC_GetMode (LPC_CRC_T *pCRC)`

*Get the CRC Mode bits.*

  - `__STATIC_INLINE void Chip_CRC_SetSeed (LPC_CRC_T *pCRC, uint32_t seed)`

*Set the seed bits used by the CRC\_SUM register.*

  - `__STATIC_INLINE uint32_t Chip_CRC_GetSeed (LPC_CRC_T *pCRC)`

*Get the CRC seed value.*

  - `__STATIC_INLINE void Chip_CRC_Write8 (LPC_CRC_T *pCRC, uint8_t data)`

*Convenience function for writing 8-bit data to the CRC engine.*

  - `__STATIC_INLINE void Chip_CRC_Write16 (LPC_CRC_T *pCRC, uint16_t data)`

*Convenience function for writing 16-bit data to the CRC engine.*

  - `__STATIC_INLINE void Chip_CRC_Write32 (LPC_CRC_T *pCRC, uint32_t data)`

*Convenience function for writing 32-bit data to the CRC engine.*

  - `__STATIC_INLINE uint32_t Chip_CRC_Sum (LPC_CRC_T *pCRC)`

*Gets the CRC Sum based on the Mode and Seed as previously configured.*

  - `__STATIC_INLINE uint32_t Chip_CRC_CRC8 (LPC_CRC_T *pCRC, const uint8_t *data, uint32_t bytes)`

*Convenience function for computing a standard CCITT checksum from an 8-bit data block.*

  - `__STATIC_INLINE uint32_t Chip_CRC_CRC16 (LPC_CRC_T *pCRC, const uint16_t *data, uint32_t hwords)`

*Convenience function for computing a standard CRC16 checksum from 16-bit data block.*

  - `__STATIC_INLINE uint32_t Chip_CRC_CRC32 (LPC_CRC_T *pCRC, const uint32_t *data, uint32_t words)`

*Convenience function for computing a standard CRC32 checksum from 32-bit data block.*

## 6.13.2 Macro Definition Documentation

### 6.13.2.1 `#define CRC_MODE_POLY_BITMASK ((0x03))` */\*\* CRC polynomial Bit mask \*/*

Definition at line 62 of file `crc_5411x.h`.

### 6.13.2.2 `#define CRC_MODE_POLY_CCITT (0x00)` */\*\* Select CRC-CCITT polynomial \*/*

Definition at line 63 of file `crc_5411x.h`.

### 6.13.2.3 `#define CRC_MODE_POLY_CRC16 (0x01)` */\*\* Select CRC-16 polynomial \*/*

Definition at line 64 of file `crc_5411x.h`.

### 6.13.2.4 `#define CRC_MODE_POLY_CRC32 (0x02)` */\*\* Select CRC-32 polynomial \*/*

Definition at line 65 of file `crc_5411x.h`.

### 6.13.2.5 `#define CRC_MODE_SUM_BIT_RVS (1 << 4)` */\*\* Select Bit order reverse for CRC\_SUM \*/*

Definition at line 70 of file `crc_5411x.h`.

### 6.13.2.6 `#define CRC_MODE_SUM_BITMASK (0x03 << 4)` */\*\* CRC Sum Config Bit mask \*/*

Definition at line 69 of file `crc_5411x.h`.

6.13.2.7 `#define CRC_MODE_SUM_CMPL (1 << 5) /** Select One's complement for CRC_SUM */`

Definition at line 71 of file `crc_5411x.h`.

6.13.2.8 `#define CRC_MODE_WRDATA_BIT_RVS (1 << 2) /** Select Bit order reverse for WR_DATA (per byte) */`

Definition at line 67 of file `crc_5411x.h`.

6.13.2.9 `#define CRC_MODE_WRDATA_BITMASK (0x03 << 2) /** CRC WR_Data Config Bit mask */`

Definition at line 66 of file `crc_5411x.h`.

6.13.2.10 `#define CRC_MODE_WRDATA_CMPL (1 << 3) /** Select One's complement for WR_DATA */`

Definition at line 68 of file `crc_5411x.h`.

6.13.2.11 `#define CRC_SEED_CCITT (0x0000FFFF) /** Initial seed value for CCITT mode */`

Definition at line 77 of file `crc_5411x.h`.

6.13.2.12 `#define CRC_SEED_CRC16 (0x00000000) /** Initial seed value for CRC16 mode */`

Definition at line 78 of file `crc_5411x.h`.

6.13.2.13 `#define CRC_SEED_CRC32 (0xFFFFFFFF) /** Initial seed value for CRC32 mode */`

Definition at line 79 of file `crc_5411x.h`.

6.13.2.14 `#define MODE_CFG_CCITT (0x00) /** Pre-defined mode word for default CCITT setup */`

Definition at line 73 of file `crc_5411x.h`.

6.13.2.15 `#define MODE_CFG_CRC16 (0x15) /** Pre-defined mode word for default CRC16 setup */`

Definition at line 74 of file `crc_5411x.h`.

6.13.2.16 `#define MODE_CFG_CRC32 (0x36) /** Pre-defined mode word for default CRC32 setup */`

Definition at line 75 of file `crc_5411x.h`.

### 6.13.3 Enumeration Type Documentation

6.13.3.1 `enum CRC_POLY_T`

CRC polynomial.

Enumerator

**`CRC_POLY_CCITT`** CRC-CCIT polynomial

**`CRC_POLY_CRC16`** CRC-16 polynomial

***CRC\_POLY\_CRC32*** CRC-32 polynomial  
***CRC\_POLY\_LAST***

Definition at line 84 of file `crc_5411x.h`.

#### 6.13.4 Function Documentation

6.13.4.1 `__STATIC_INLINE uint32_t Chip_CRC_CRC16 ( LPC_CRC_T * pCRC, const uint16_t * data, uint32_t hwords )`

Convenience function for computing a standard CRC16 checksum from 16-bit data block.

Parameters

<i>pCRC</i>	: Pointer to the crc register block
<i>data</i>	: Pointer to the block of 16-bit data
<i>hwords</i>	: The number of 16 byte entries pointed to by data

Returns

Check sum value

Definition at line 308 of file `crc_5411x.h`.

6.13.4.2 `__STATIC_INLINE uint32_t Chip_CRC_CRC32 ( LPC_CRC_T * pCRC, const uint32_t * data, uint32_t words )`

Convenience function for computing a standard CRC32 checksum from 32-bit data block.

Parameters

<i>pCRC</i>	: Pointer to the crc register block
<i>data</i>	: Pointer to the block of 32-bit data
<i>words</i>	: The number of 32-bit entries pointed to by data

Returns

Check sum value

Definition at line 327 of file `crc_5411x.h`.

6.13.4.3 `__STATIC_INLINE uint32_t Chip_CRC_CRC8 ( LPC_CRC_T * pCRC, const uint8_t * data, uint32_t bytes )`

Convenience function for computing a standard CCITT checksum from an 8-bit data block.

Parameters

<i>pCRC</i>	: Pointer to the crc register block
<i>data</i>	: Pointer to the block of 8-bit data
<i>bytes</i>	: The number of bytes pointed to by data

Returns

Check sum value

Definition at line 289 of file `crc_5411x.h`.

6.13.4.4 `__STATIC_INLINE void Chip_CRC_Deinit ( LPC_CRC_T * pCRC )`

Deinitializes the CRC Engine.

## Parameters

<i>pCRC</i>	: Pointer to the crc register block
-------------	-------------------------------------

## Returns

Nothing

Definition at line 114 of file crc\_5411x.h.

#### 6.13.4.5 `__STATIC_INLINE uint32_t Chip_CRC_GetMode ( LPC_CRC_T * pCRC )`

Get the CRC Mode bits.

## Parameters

<i>pCRC</i>	: Pointer to the crc register block
-------------	-------------------------------------

## Returns

The current value of the CRC Mode bits

Definition at line 213 of file crc\_5411x.h.

#### 6.13.4.6 `__STATIC_INLINE uint32_t Chip_CRC_GetSeed ( LPC_CRC_T * pCRC )`

Get the CRC seed value.

## Parameters

<i>pCRC</i>	: Pointer to the crc register block
-------------	-------------------------------------

## Returns

Seed value

Definition at line 234 of file crc\_5411x.h.

#### 6.13.4.7 `__STATIC_INLINE void Chip_CRC_Init ( LPC_CRC_T * pCRC )`

Initializes the CRC Engine.

## Parameters

<i>pCRC</i>	: Pointer to the crc register block
-------------	-------------------------------------

## Returns

Nothing

Definition at line 96 of file crc\_5411x.h.

#### 6.13.4.8 `__STATIC_INLINE void Chip_CRC_SetMode ( LPC_CRC_T * pCRC, uint32_t mode )`

Set the CRC Mode bits.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
<i>mode</i>	: Mode value

**Returns**

Nothing

Definition at line 203 of file crc\_5411x.h.

6.13.4.9 `__STATIC_INLINE void Chip_CRC_SetPoly ( LPC_CRC_T * pCRC, CRC_POLY_T poly, uint32_t flags )`

Set the polynomial used for the CRC calculation.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
<i>poly</i>	: The enumerated polynomial to be used
<i>flags</i>	: An Or'ed value of flags that setup the mode

**Returns**

Nothing

**Note**

Flags for setting up the mode word include CRC\_MODE\_WRDATA\_BIT\_RVS, CRC\_MODE\_WRDATA\_CMPL, CRC\_MODE\_SUM\_BIT\_RVS, and CRC\_MODE\_SUM\_CMPL.

Definition at line 135 of file crc\_5411x.h.

6.13.4.10 `__STATIC_INLINE void Chip_CRC_SetSeed ( LPC_CRC_T * pCRC, uint32_t seed )`

Set the seed bits used by the CRC\_SUM register.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
<i>seed</i>	: Seed value

**Returns**

Nothing

Definition at line 224 of file crc\_5411x.h.

6.13.4.11 `__STATIC_INLINE uint32_t Chip_CRC_Sum ( LPC_CRC_T * pCRC )`

Gets the CRC Sum based on the Mode and Seed as previously configured.

**Parameters**


---

<i>pCRC</i>	: Pointer to the crc register block
-------------	-------------------------------------

**Returns**

CRC Checksum value

Definition at line 277 of file crc\_5411x.h.

**6.13.4.12 `__STATIC_INLINE void Chip_CRC_UseCCITT ( LPC_CRC_T * pCRC )`**

Sets up the CRC engine for CCITT mode.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
-------------	-------------------------------------

**Returns**

Nothing

Definition at line 167 of file crc\_5411x.h.

**6.13.4.13 `__STATIC_INLINE void Chip_CRC_UseCRC16 ( LPC_CRC_T * pCRC )`**

Sets up the CRC engine for CRC16 mode.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
-------------	-------------------------------------

**Returns**

Nothing

Definition at line 145 of file crc\_5411x.h.

**6.13.4.14 `__STATIC_INLINE void Chip_CRC_UseCRC32 ( LPC_CRC_T * pCRC )`**

Sets up the CRC engine for CRC32 mode.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
-------------	-------------------------------------

**Returns**

Nothing

Definition at line 156 of file crc\_5411x.h.

**6.13.4.15 `__STATIC_INLINE void Chip_CRC_UseDefaultConfig ( LPC_CRC_T * pCRC, CRC_POLY_T poly )`**

Engage the CRC engine with defaults based on the polynomial to be used.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
<i>poly</i>	: The enumerated polynomial to be used

**Returns**

Nothing

Definition at line 179 of file crc\_5411x.h.

**6.13.4.16 \_\_STATIC\_INLINE void Chip\_CRC\_Write16 ( LPC\_CRC\_T \* pCRC, uint16\_t data )**

Convenience function for writing 16-bit data to the CRC engine.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
<i>data</i>	: 16-bit data to write

**Returns**

Nothing

Definition at line 256 of file crc\_5411x.h.

**6.13.4.17 \_\_STATIC\_INLINE void Chip\_CRC\_Write32 ( LPC\_CRC\_T \* pCRC, uint32\_t data )**

Convenience function for writing 32-bit data to the CRC engine.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
<i>data</i>	: 32-bit data to write

**Returns**

Nothing

Definition at line 267 of file crc\_5411x.h.

**6.13.4.18 \_\_STATIC\_INLINE void Chip\_CRC\_Write8 ( LPC\_CRC\_T \* pCRC, uint8\_t data )**

Convenience function for writing 8-bit data to the CRC engine.

**Parameters**

<i>pCRC</i>	: Pointer to the crc register block
<i>data</i>	: 8-bit data to write

**Returns**

Nothing

Definition at line 245 of file crc\_5411x.h.



## 6.14 CHIP: LPC5411X DMA Controller driver channel specific functions (legacy)

### 6.14.1 Detailed Description

#### Data Structures

- struct [DMA\\_CHDESC\\_T](#)

#### Functions

- `__STATIC_INLINE void Chip\_DMA\_SetupChannelConfig (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch, uint32\_t cfg)`  
*Setup a DMA channel configuration.*
- `__STATIC_INLINE uint32\_t Chip\_DMA\_GetChannelStatus (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch)`  
*Returns channel specific status flags.*
- `__STATIC_INLINE void Chip\_DMA\_SetupChannelTransfer (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch, uint32\_t cfg)`  
*Setup a DMA channel transfer configuration.*
- `__STATIC_INLINE void Chip\_DMA\_SetTranBits (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch, uint32\_t mask)`  
*Set DMA transfer register interrupt bits (safe)*
- `__STATIC_INLINE void Chip\_DMA\_ClearTranBits (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch, uint32\_t mask)`  
*Clear DMA transfer register interrupt bits (safe)*
- `__STATIC_INLINE void Chip\_DMA\_SetupChannelTransferSize (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch, uint32\_t trans)`  
*Update the transfer size in an existing DMA channel transfer configuration.*
- `__STATIC_INLINE void Chip\_DMA\_SetChannelValid (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch)`  
*Sets a DMA channel configuration as valid.*
- `__STATIC_INLINE void Chip\_DMA\_SetChannelInvalid (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch)`  
*Sets a DMA channel configuration as invalid.*
- `__STATIC_INLINE void Chip\_DMA\_SWTriggerChannel (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch)`  
*Performs a software trigger of the DMA channel.*
- `__STATIC_INLINE bool Chip\_DMA\_SetupTranChannel (LPC\_DMA\_T *pDMA, DMA\_CHID\_T ch, DMA\_CHDESC\_T *desc)`  
*Sets up a DMA channel with the passed DMA transfer descriptor.*

#### Variables

- [DMA\\_CHDESC\\_T](#) [Chip\\_DMA\\_Table](#) [[MAX\\_DMA\\_CHANNEL](#)]

### 6.14.2 Function Documentation

#### 6.14.2.1 `__STATIC_INLINE void Chip\_DMA\_ClearTranBits ( LPC\_DMA\_T * pDMA, DMA\_CHID\_T ch, uint32\_t mask )`

Clear DMA transfer register interrupt bits (safe)

##### Parameters

<a href="#">pDMA</a>	: The base of DMA controller on the chip
----------------------	--

<i>ch</i>	: DMA channel ID
<i>mask</i>	: Bits to clear

**Returns**

Nothing

**Note**

This function safely clears bits in the DMA channel specific XFRCFG register.

Definition at line 624 of file dma\_5411x.h.

**6.14.2.2** `__STATIC_INLINE uint32_t Chip_DMA_GetChannelStatus ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Returns channel specific status flags.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

AN Or'ed value of DMA\_CTLSTAT\_VALIDPENDING and DMA\_CTLSTAT\_TRIG

Definition at line 574 of file dma\_5411x.h.

**6.14.2.3** `__STATIC_INLINE void Chip_DMA_SetChannelInvalid ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Sets a DMA channel configuration as invalid.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 659 of file dma\_5411x.h.

**6.14.2.4** `__STATIC_INLINE void Chip_DMA_SetChannelValid ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Sets a DMA channel configuration as valid.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 648 of file dma\_5411x.h.

6.14.2.5 `__STATIC_INLINE void Chip_DMA_SetTranBits ( LPC_DMA_T * pDMA, DMA_CHID_T ch, uint32_t mask )`

Set DMA transfer register interrupt bits (safe)

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID
<i>mask</i>	: Bits to set

**Returns**

Nothing

**Note**

This function safely sets bits in the DMA channel specific XFERCFG register.

Definition at line 610 of file dma\_5411x.h.

6.14.2.6 `__STATIC_INLINE void Chip_DMA_SetupChannelConfig ( LPC_DMA_T * pDMA, DMA_CHID_T ch, uint32_t cfg )`

Setup a DMA channel configuration.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID
<i>cfg</i>	: An Or'ed value of DMA_CFG_* values that define the channel's configuration

**Returns**

Nothing

**Note**

This function sets up all configurable options for the DMA channel. These options are usually set once for a channel and then unchanged.

The following example show how to configure the channel for peripheral DMA requests, burst transfer size of 1 (in 'transfers', not bytes), continuous reading of the same source address, incrementing destination address, and highest channel priority.

Example: `Chip_DMA_SetupChannelConfig(pDMA, SSP0_RX_DMA, (DMA_CFG_PERIPHREQEN | DMA_CFG_↵_TRIGBURST_BURST | DMA_CFG_BURSTPOWER_1 | DMA_CFG_SRCBURSTWRAP | DMA_CFG_CHPRI↵ORITY(0)));`

The following example show how to configure the channel for an external trigger from the input mux with low edge polarity, a burst transfer size of 8, incrementing source and destination addresses, and lowest channel priority.

Example: `Chip_DMA_SetupChannelConfig(pDMA, DMA_CH14, (DMA_CFG_HWTRIGEN | DMA_CFG_TRIGP↵OL_LOW | DMA_CFG_TRIGTYPE_EDGE | DMA_CFG_TRIGBURST_BURST | DMA_CFG_BURSTPOWER_8 | DMA_CFG_CHPRIORITY(3)));`

For non-peripheral DMA triggering (DMA\_CFG\_HWTRIGEN definition), use the DMA input mux functions to configure the DMA trigger source for a DMA channel.

Definition at line 563 of file dma\_5411x.h.

6.14.2.7 `__STATIC_INLINE void Chip_DMA_SetupChannelTransfer ( LPC_DMA_T * pDMA, DMA_CHID_T ch, uint32_t cfg )`

Setup a DMA channel transfer configuration.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID
<i>cfg</i>	: An Or'ed value of DMA_XFERCFG_* values that define the channel's transfer configuration

## Returns

Nothing

## Note

This function sets up the transfer configuration for the DMA channel.

The following example show how to configure the channel's transfer for multiple transfer descriptors (ie, ping-pong), interrupt 'A' trigger on transfer descriptor completion, 128 byte size transfers, and source and destination address increment.

Example: `Chip_DMA_SetupChannelTransfer(pDMA, SSP0_RX_DMA, (DMA_XFERCFG_CFGVALID | DMA_XFERCFG_RELOAD | DMA_XFERCFG_SETINTA | DMA_XFERCFG_WIDTH_8 | DMA_XFERCFG_SRCINC_1 | DMA_XFERCFG_DSTINC_1 | DMA_XFERCFG_XFERCOUNT(128)));`

Definition at line 596 of file dma\_5411x.h.

6.14.2.8 `__STATIC_INLINE void Chip_DMA_SetupChannelTransferSize ( LPC_DMA_T * pDMA, DMA_CHID_T ch, uint32_t trans )`

Update the transfer size in an existing DMA channel transfer configuration.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID
<i>trans</i>	: Number of transfers to update the transfer configuration to (1 - 1023)

## Returns

Nothing

Definition at line 636 of file dma\_5411x.h.

6.14.2.9 `__STATIC_INLINE bool Chip_DMA_SetupTranChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch, DMA_CHDESC_T * desc )`

Sets up a DMA channel with the passed DMA transfer descriptor.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID
<i>desc</i>	: Pointer to DMA transfer descriptor

## Returns

false if the DMA channel was active, otherwise true

**Note**

This function will set the DMA descriptor in the SRAM table to the the passed descriptor. This function is only meant to be used when the DMA channel is not active and can be used to setup the initial transfer for a linked list or ping-pong buffer or just a single transfer without a next descriptor.

If using this function to write the initial transfer descriptor in a linked list or ping-pong buffer configuration, it should contain a non-NULL 'next' field pointer.

Definition at line 691 of file dma\_5411x.h.

**6.14.2.10   \_\_STATIC\_INLINE void Chip\_DMA\_SWTriggerChannel ( LPC\_DMA\_T \* *pDMA*, DMA\_CHID\_T *ch* )**

Performs a software trigger of the DMA channel.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 670 of file dma\_5411x.h.

**6.14.3   Variable Documentation****6.14.3.1   DMA\_CHDESC\_T Chip\_DMA\_Table[MAX\_DMA\_CHANNEL]**

## 6.15 CHIP: LPC5411X DMA Controller driver common channel functions (legacy)

### 6.15.1 Detailed Description

#### Functions

- `__STATIC_INLINE void Chip_DMA_EnableChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Enables a single DMA channel.*
- `__STATIC_INLINE void Chip_DMA_DisableChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Disables a single DMA channel.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetEnabledChannels (LPC_DMA_T *pDMA)`  
*Returns all enabled DMA channels.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetActiveChannels (LPC_DMA_T *pDMA)`  
*Returns all active DMA channels.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetBusyChannels (LPC_DMA_T *pDMA)`  
*Returns all busy DMA channels.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetErrorIntChannels (LPC_DMA_T *pDMA)`  
*Returns pending error interrupt status for all DMA channels.*
- `__STATIC_INLINE void Chip_DMA_ClearErrorIntChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Clears a pending error interrupt status for a single DMA channel.*
- `__STATIC_INLINE void Chip_DMA_EnableIntChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Enables a single DMA channel's interrupt used in common DMA interrupt.*
- `__STATIC_INLINE void Chip_DMA_DisableIntChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Disables a single DMA channel's interrupt used in common DMA interrupt.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetEnableIntChannels (LPC_DMA_T *pDMA)`  
*Returns all enabled interrupt channels.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetActiveIntAChannels (LPC_DMA_T *pDMA)`  
*Returns active A interrupt status for all channels.*
- `__STATIC_INLINE void Chip_DMA_ClearActiveIntAChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Clears active A interrupt status for a single channel.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetActiveIntBChannels (LPC_DMA_T *pDMA)`  
*Returns active B interrupt status for all channels.*
- `__STATIC_INLINE void Chip_DMA_ClearActiveIntBChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Clears active B interrupt status for a single channel.*
- `__STATIC_INLINE void Chip_DMA_SetValidChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Sets the VALIDPENDING control bit for a single channel.*
- `__STATIC_INLINE void Chip_DMA_SetTrigChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Sets the TRIG bit for a single channel.*
- `__STATIC_INLINE void Chip_DMA_AbortChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Aborts a DMA operation for a single channel.*

### 6.15.2 Function Documentation

#### 6.15.2.1 `__STATIC_INLINE void Chip_DMA_AbortChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Aborts a DMA operation for a single channel.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

**Note**

To abort a channel, the channel should first be disabled. Then wait until the channel is no longer busy by checking the corresponding bit in BUSY. Finally, abort the channel operation. This prevents the channel from restarting an incomplete operation when it is enabled again.

Definition at line 509 of file dma\_5411x.h.

#### 6.15.2.2 `__STATIC_INLINE void Chip_DMA_ClearActiveIntAChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Clears active A interrupt status for a single channel.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 443 of file dma\_5411x.h.

#### 6.15.2.3 `__STATIC_INLINE void Chip_DMA_ClearActiveIntBChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Clears active B interrupt status for a single channel.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 468 of file dma\_5411x.h.

#### 6.15.2.4 `__STATIC_INLINE void Chip_DMA_ClearErrorIntChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Clears a pending error interrupt status for a single DMA channel.

**Parameters**


---



<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 381 of file dma\_5411x.h.

**6.15.2.5** `__STATIC_INLINE void Chip_DMA_DisableChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Disables a single DMA channel.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 312 of file dma\_5411x.h.

**6.15.2.6** `__STATIC_INLINE void Chip_DMA_DisableIntChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Disables a single DMA channel's interrupt used in common DMA interrupt.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 403 of file dma\_5411x.h.

**6.15.2.7** `__STATIC_INLINE void Chip_DMA_EnableChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Enables a single DMA channel.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 301 of file dma\_5411x.h.

**6.15.2.8** `__STATIC_INLINE void Chip_DMA_EnableIntChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Enables a single DMA channel's interrupt used in common DMA interrupt.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

Definition at line 392 of file dma\_5411x.h.

#### 6.15.2.9 `__STATIC_INLINE uint32_t Chip_DMA_GetActiveChannels ( LPC_DMA_T * pDMA )`

Returns all active DMA channels.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

**Returns**

An Or'ed value of all active DMA channels (0 - 15)

**Note**

A high values in bits 0 .. 15 in the return values indicates that the channel for that bit (bit 0 = channel 0, bit 1 - channel 1, etc.) is active. A low state is inactive. A active channel indicates that a DMA operation has been started but not yet fully completed.

Definition at line 340 of file dma\_5411x.h.

#### 6.15.2.10 `__STATIC_INLINE uint32_t Chip_DMA_GetActiveIntAChannels ( LPC_DMA_T * pDMA )`

Returns active A interrupt status for all channels.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

**Returns**

Nothing

**Note**

A high values in bits 0 .. 15 in the return values indicates that the channel for that bit (bit 0 = channel 0, bit 1 - channel 1, etc.) has an active A interrupt for the channel. A low state indicates that the A interrupt is not active.

Definition at line 432 of file dma\_5411x.h.

#### 6.15.2.11 `__STATIC_INLINE uint32_t Chip_DMA_GetActiveIntBChannels ( LPC_DMA_T * pDMA )`

Returns active B interrupt status for all channels.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

## Returns

Nothing

## Note

A high values in bits 0 .. 15 in the return values indicates that the channel for that bit (bit 0 = channel 0, bit 1 - channel 1, etc.) has an active B interrupt for the channel. A low state indicates that the B interrupt is not active.

Definition at line 457 of file dma\_5411x.h.

6.15.2.12 `__STATIC_INLINE uint32_t Chip_DMA_GetBusyChannels ( LPC_DMA_T * pDMA )`

Returns all busy DMA channels.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

## Returns

An Or'ed value of all busy DMA channels (0 - 15)

## Note

A high values in bits 0 .. 15 in the return values indicates that the channel for that bit (bit 0 = channel 0, bit 1 - channel 1, etc.) is busy. A low state is not busy. A DMA channel is considered busy when there is any operation related to that channel in the DMA controllers internal pipeline.

Definition at line 356 of file dma\_5411x.h.

6.15.2.13 `__STATIC_INLINE uint32_t Chip_DMA_GetEnabledChannels ( LPC_DMA_T * pDMA )`

Returns all enabled DMA channels.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

## Returns

An Or'ed value of all enabled DMA channels (0 - 15)

## Note

A high values in bits 0 .. 15 in the return values indicates that the channel for that bit (bit 0 = channel 0, bit 1 - channel 1, etc.) is enabled. A low state is disabled.

Definition at line 325 of file dma\_5411x.h.

6.15.2.14 `__STATIC_INLINE uint32_t Chip_DMA_GetEnableIntChannels ( LPC_DMA_T * pDMA )`

Returns all enabled interrupt channels.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

**Returns**

Nothing

**Note**

A high values in bits 0 .. 15 in the return values indicates that the channel for that bit (bit 0 = channel 0, bit 1 - channel 1, etc.) has an enabled interrupt for the channel. A low state indicates that the DMA channel will not contribute to the common DMA interrupt status.

Definition at line 418 of file dma\_5411x.h.

#### 6.15.2.15 `__STATIC_INLINE uint32_t Chip_DMA_GetErrorIntChannels ( LPC_DMA_T * pDMA )`

Returns pending error interrupt status for all DMA channels.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

**Returns**

An Or'ed value of all channels (0 - 15) error interrupt status

**Note**

A high values in bits 0 .. 15 in the return values indicates that the channel for that bit (bit 0 = channel 0, bit 1 - channel 1, etc.) has a pending error interrupt. A low state indicates no error interrupt.

Definition at line 370 of file dma\_5411x.h.

#### 6.15.2.16 `__STATIC_INLINE void Chip_DMA_SetTrigChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Sets the TRIG bit for a single channel.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

**Returns**

Nothing

**Note**

See the User Manual for more information for what this bit does.

Definition at line 493 of file dma\_5411x.h.

#### 6.15.2.17 `__STATIC_INLINE void Chip_DMA_SetValidChannel ( LPC_DMA_T * pDMA, DMA_CHID_T ch )`

Sets the VALIDPENDING control bit for a single channel.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
<i>ch</i>	: DMA channel ID

## Returns

Nothing

## Note

See the User Manual for more information for what this bit does.

Definition at line 481 of file dma\_5411x.h.

## 6.16 CHIP: LPC5411X DMA Controller driver common functions (legacy)

### 6.16.1 Detailed Description

#### Functions

- `__STATIC_INLINE void Chip_DMA_Init (LPC_DMA_T *pDMA)`  
*Initialize DMA controller.*
- `__STATIC_INLINE void Chip_DMA_DeInit (LPC_DMA_T *pDMA)`  
*De-Initialize DMA controller.*
- `__STATIC_INLINE void Chip_DMA_Enable (LPC_DMA_T *pDMA)`  
*Enable DMA controller.*
- `__STATIC_INLINE void Chip_DMA_Disable (LPC_DMA_T *pDMA)`  
*Disable DMA controller.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetIntStatus (LPC_DMA_T *pDMA)`  
*Get pending interrupt or error interrupts.*
- `__STATIC_INLINE void Chip_DMA_SetSRAMBase (LPC_DMA_T *pDMA, uint32_t base)`  
*Set DMA controller SRAM base address.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetSRAMBase (LPC_DMA_T *pDMA)`  
*Returns DMA controller SRAM base address.*

### 6.16.2 Function Documentation

#### 6.16.2.1 `__STATIC_INLINE void Chip_DMA_DeInit ( LPC_DMA_T * pDMA )`

De-Initialize DMA controller.

##### Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

##### Returns

Nothing

Definition at line 221 of file dma\_5411x.h.

#### 6.16.2.2 `__STATIC_INLINE void Chip_DMA_Disable ( LPC_DMA_T * pDMA )`

Disable DMA controller.

##### Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

##### Returns

Nothing

Definition at line 242 of file dma\_5411x.h.

#### 6.16.2.3 `__STATIC_INLINE void Chip_DMA_Enable ( LPC_DMA_T * pDMA )`

Enable DMA controller.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

## Returns

Nothing

Definition at line 232 of file dma\_5411x.h.

6.16.2.4 `__STATIC_INLINE uint32_t Chip_DMA_GetIntStatus ( LPC_DMA_T * pDMA )`

Get pending interrupt or error interrupts.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

## Returns

An Or'ed value of DMA\_INTSTAT\_\* types

## Note

If any DMA channels have an active interrupt or error interrupt pending, this functional will a common status that applies to all channels.

Definition at line 255 of file dma\_5411x.h.

6.16.2.5 `__STATIC_INLINE uint32_t Chip_DMA_GetSRAMBase ( LPC_DMA_T * pDMA )`

Returns DMA controller SRAM base address.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

## Returns

The base address where the DMA descriptors are stored

Definition at line 282 of file dma\_5411x.h.

6.16.2.6 `__STATIC_INLINE void Chip_DMA_Init ( LPC_DMA_T * pDMA )`

Initialize DMA controller.

## Parameters

<i>pDMA</i>	: The base of DMA controller on the chip
-------------	--

## Returns

Nothing

Definition at line 210 of file dma\_5411x.h.

6.16.2.7 `__STATIC_INLINE void Chip_DMA_SetSRAMBase ( LPC_DMA_T * pDMA, uint32_t base )`

Set DMA controller SRAM base address.

**Parameters**

<i>pDMA</i>	: The base of DMA controller on the chip
<i>base</i>	: The base address where the DMA descriptors will be stored

**Returns**

Nothing

**Note**

A 256 byte block of memory aligned on a 256 byte boundary must be provided for this function. It sets the base address used for DMA descriptor table (16 descriptors total that use 16 bytes each).

A pre-defined table with correct alignment can be used for this function by calling `Chip_DMA_SetSRAMBase(LP_C_DMA, DMA_ADDR(Chip_DMA_Table));`

Definition at line 272 of file `dma_5411x.h`.



## 6.17 CHIP: LPC5411X DMA Engine driver (legacy)

### 6.17.1 Detailed Description

#### Data Structures

- struct [LPC\\_DMA\\_COMMON\\_T](#)  
*DMA Controller shared registers structure.*
- struct [LPC\\_DMA\\_CHANNEL\\_T](#)  
*DMA Controller shared registers structure.*
- struct [LPC\\_DMA\\_T](#)  
*DMA Controller register block structure.*

#### Modules

- [CHIP: LPC5411X DMA Controller driver channel specific functions \(legacy\)](#)
- [CHIP: LPC5411X DMA Controller driver common channel functions \(legacy\)](#)
- [CHIP: LPC5411X DMA Controller driver common functions \(legacy\)](#)

#### Macros

- `#define MAX_DMA_CHANNEL (20)`
- `#define DMA_INTSTAT_ACTIVEINT 0x2`
- `#define DMA_INTSTAT_ACTIVEERRINT 0x4`
- `#define DMA_ADDR(addr) ((uint32_t) (addr))`
- `#define DMA_CFG_PERIPHREQEN (1 << 0)`
- `#define DMA_CFG_HWTRIGEN (1 << 1)`
- `#define DMA_CFG_TRIGPOL_LOW (0 << 4)`
- `#define DMA_CFG_TRIGPOL_HIGH (1 << 4)`
- `#define DMA_CFG_TRIGTYPE_EDGE (0 << 5)`
- `#define DMA_CFG_TRIGTYPE_LEVEL (1 << 5)`
- `#define DMA_CFG_TRIGBURST_SNGL (0 << 6)`
- `#define DMA_CFG_TRIGBURST_BURST (1 << 6)`
- `#define DMA_CFG_BURSTPOWER_1 (0 << 8)`
- `#define DMA_CFG_BURSTPOWER_2 (1 << 8)`
- `#define DMA_CFG_BURSTPOWER_4 (2 << 8)`
- `#define DMA_CFG_BURSTPOWER_8 (3 << 8)`
- `#define DMA_CFG_BURSTPOWER_16 (4 << 8)`
- `#define DMA_CFG_BURSTPOWER_32 (5 << 8)`
- `#define DMA_CFG_BURSTPOWER_64 (6 << 8)`
- `#define DMA_CFG_BURSTPOWER_128 (7 << 8)`
- `#define DMA_CFG_BURSTPOWER_256 (8 << 8)`
- `#define DMA_CFG_BURSTPOWER_512 (9 << 8)`
- `#define DMA_CFG_BURSTPOWER_1024 (10 << 8)`
- `#define DMA_CFG_BURSTPOWER(n) ((n) << 8)`
- `#define DMA_CFG_SRCBURSTWRAP (1 << 14)`
- `#define DMA_CFG_DSTBURSTWRAP (1 << 15)`
- `#define DMA_CFG_CHPRIORITY(p) ((p) << 16)`
- `#define DMA_CTLSTAT_VALIDPENDING (1 << 0)`
- `#define DMA_CTLSTAT_TRIG (1 << 2)`
- `#define DMA_XFERCFG_CFGVALID (1 << 0)`
- `#define DMA_XFERCFG_RELOAD (1 << 1)`
- `#define DMA_XFERCFG_SWTRIG (1 << 2)`

- `#define DMA_XFERCFG_CLRTRIG (1 << 3)`
- `#define DMA_XFERCFG_SETINTA (1 << 4)`
- `#define DMA_XFERCFG_SETINTB (1 << 5)`
- `#define DMA_XFERCFG_WIDTH_8 (0 << 8)`
- `#define DMA_XFERCFG_WIDTH_16 (1 << 8)`
- `#define DMA_XFERCFG_WIDTH_32 (2 << 8)`
- `#define DMA_XFERCFG_SRCINC_0 (0 << 12)`
- `#define DMA_XFERCFG_SRCINC_1 (1 << 12)`
- `#define DMA_XFERCFG_SRCINC_2 (2 << 12)`
- `#define DMA_XFERCFG_SRCINC_4 (3 << 12)`
- `#define DMA_XFERCFG_DSTINC_0 (0 << 14)`
- `#define DMA_XFERCFG_DSTINC_1 (1 << 14)`
- `#define DMA_XFERCFG_DSTINC_2 (2 << 14)`
- `#define DMA_XFERCFG_DSTINC_4 (3 << 14)`
- `#define DMA_XFERCFG_XFERCOUNT(n) ((n - 1) << 16)`

## Enumerations

- `enum DMA_CHID_T {`  
`DMA_CH0, DMA_CH1, DMA_CH2, DMA_CH3,`  
`DMA_CH4, DMA_CH5, DMA_CH6, DMA_CH7,`  
`DMA_CH8, DMA_CH9, DMA_CH10, DMA_CH11,`  
`DMA_CH12, DMA_CH13, DMA_CH14, DMA_CH15,`  
`DMA_CH16, DMA_CH17, DMA_CH18, DMA_CH19,`  
`DMAREQ_FLEXCOMM0_RX = DMA_CH0, DMAREQ_FLEXCOMM0_TX, DMAREQ_FLEXCOMM1_RX,`  
`DMAREQ_FLEXCOMM1_TX,`  
`DMAREQ_FLEXCOMM2_RX, DMAREQ_FLEXCOMM2_TX, DMAREQ_FLEXCOMM3_RX, DMAREQ_FLEXCOMM3_TX,`  
`DMAREQ_FLEXCOMM4_RX, DMAREQ_FLEXCOMM4_TX, DMAREQ_FLEXCOMM5_RX, DMAREQ_FLEXCOMM5_TX,`  
`DMAREQ_FLEXCOMM6_RX, DMAREQ_FLEXCOMM6_TX, DMAREQ_FLEXCOMM7_RX, DMAREQ_FLEXCOMM7_TX,`  
`DMAREQ_DMIC0, DMAREQ_DMIC1, DMAREQ_SPIFI }`

## 6.17.2 Macro Definition Documentation

### 6.17.2.1 `#define DMA_ADDR( addr ) ((uint32_t) (addr))`

Definition at line 149 of file `dma_5411x.h`.

### 6.17.2.2 `#define DMA_CFG_BURSTPOWER( n ) ((n) << 8)`

Set DMA burst size to  $2^n$  transfers, max  $n=10$

Definition at line 172 of file `dma_5411x.h`.

### 6.17.2.3 `#define DMA_CFG_BURSTPOWER_1 (0 << 8)`

Set DMA burst size to 1 transfer

Definition at line 161 of file `dma_5411x.h`.

**6.17.2.4 #define DMA\_CFG\_BURSTPOWER\_1024 (10 << 8)**

Set DMA burst size to 1024 transfers

Definition at line 171 of file dma\_5411x.h.

**6.17.2.5 #define DMA\_CFG\_BURSTPOWER\_128 (7 << 8)**

Set DMA burst size to 128 transfers

Definition at line 168 of file dma\_5411x.h.

**6.17.2.6 #define DMA\_CFG\_BURSTPOWER\_16 (4 << 8)**

Set DMA burst size to 16 transfers

Definition at line 165 of file dma\_5411x.h.

**6.17.2.7 #define DMA\_CFG\_BURSTPOWER\_2 (1 << 8)**

Set DMA burst size to 2 transfers

Definition at line 162 of file dma\_5411x.h.

**6.17.2.8 #define DMA\_CFG\_BURSTPOWER\_256 (8 << 8)**

Set DMA burst size to 256 transfers

Definition at line 169 of file dma\_5411x.h.

**6.17.2.9 #define DMA\_CFG\_BURSTPOWER\_32 (5 << 8)**

Set DMA burst size to 32 transfers

Definition at line 166 of file dma\_5411x.h.

**6.17.2.10 #define DMA\_CFG\_BURSTPOWER\_4 (2 << 8)**

Set DMA burst size to 4 transfers

Definition at line 163 of file dma\_5411x.h.

**6.17.2.11 #define DMA\_CFG\_BURSTPOWER\_512 (9 << 8)**

Set DMA burst size to 512 transfers

Definition at line 170 of file dma\_5411x.h.

**6.17.2.12 #define DMA\_CFG\_BURSTPOWER\_64 (6 << 8)**

Set DMA burst size to 64 transfers

Definition at line 167 of file dma\_5411x.h.

**6.17.2.13 #define DMA\_CFG\_BURSTPOWER\_8 (3 << 8)**

Set DMA burst size to 8 transfers

Definition at line 164 of file dma\_5411x.h.

**6.17.2.14 #define DMA\_CFG\_CHPRIORITY( p ) ((p) << 16)**

Sets DMA channel priority, min 0 (highest), max 3 (lowest)

Definition at line 175 of file dma\_5411x.h.

**6.17.2.15 #define DMA\_CFG\_DSTBURSTWRAP (1 << 15)**

Destination burst wrapping is enabled for this DMA channel

Definition at line 174 of file dma\_5411x.h.

**6.17.2.16 #define DMA\_CFG\_HWTRIGEN (1 << 1)**

Use hardware triggering via input mux

Definition at line 154 of file dma\_5411x.h.

**6.17.2.17 #define DMA\_CFG\_PERIPHREQEN (1 << 0)**

Enables Peripheral DMA requests

Definition at line 153 of file dma\_5411x.h.

**6.17.2.18 #define DMA\_CFG\_SRCBURSTWRAP (1 << 14)**

Source burst wrapping is enabled for this DMA channel

Definition at line 173 of file dma\_5411x.h.

**6.17.2.19 #define DMA\_CFG\_TRIGBURST\_BURST (1 << 6)**

Burst transfer (see UM)

Definition at line 160 of file dma\_5411x.h.

**6.17.2.20 #define DMA\_CFG\_TRIGBURST\_SINGL (0 << 6)**

Single transfer. Hardware trigger causes a single transfer

Definition at line 159 of file dma\_5411x.h.

**6.17.2.21 #define DMA\_CFG\_TRIGPOL\_HIGH (1 << 4)**

Hardware trigger is active high or rising edge

Definition at line 156 of file dma\_5411x.h.

**6.17.2.22 #define DMA\_CFG\_TRIGPOL\_LOW (0 << 4)**

Hardware trigger is active low or falling edge

Definition at line 155 of file dma\_5411x.h.

**6.17.2.23 #define DMA\_CFG\_TRIGTYPE\_EDGE (0 << 5)**

Hardware trigger is edge triggered

Definition at line 157 of file dma\_5411x.h.

**6.17.2.24 #define DMA\_CFG\_TRIGTYPE\_LEVEL (1 << 5)**

Hardware trigger is level triggered

Definition at line 158 of file dma\_5411x.h.

**6.17.2.25 #define DMA\_CTLSTAT\_TRIG (1 << 2)**

Trigger flag. Indicates that the trigger for this channel is currently set

Definition at line 179 of file dma\_5411x.h.

**6.17.2.26 #define DMA\_CTLSTAT\_VALIDPENDING (1 << 0)**

Valid pending flag for this channel

Definition at line 178 of file dma\_5411x.h.

**6.17.2.27 #define DMA\_INTSTAT\_ACTIVEERRINT 0x4**

Summarizes whether any error interrupts are pending

Definition at line 146 of file dma\_5411x.h.

**6.17.2.28 #define DMA\_INTSTAT\_ACTIVEINT 0x2**

Summarizes whether any enabled interrupts are pending

Definition at line 145 of file dma\_5411x.h.

**6.17.2.29 #define DMA\_XFERCFG\_CFGVALID (1 << 0)**

Configuration Valid flag

Definition at line 182 of file dma\_5411x.h.

**6.17.2.30 #define DMA\_XFERCFG\_CLRTRIG (1 << 3)**

Clear Trigger

Definition at line 185 of file dma\_5411x.h.

**6.17.2.31 #define DMA\_XFERCFG\_DSTINC\_0 (0 << 14)**

DMA destination address is not incremented after a transfer

Definition at line 195 of file dma\_5411x.h.

**6.17.2.32 #define DMA\_XFERCFG\_DSTINC\_1 (1 << 14)**

DMA destination address is incremented by 1 (width) after a transfer

Definition at line 196 of file dma\_5411x.h.

**6.17.2.33 #define DMA\_XFERCFG\_DSTINC\_2 (2 << 14)**

DMA destination address is incremented by 2 (width) after a transfer

Definition at line 197 of file dma\_5411x.h.

**6.17.2.34 #define DMA\_XFERCFG\_DSTINC\_4 (3 << 14)**

DMA destination address is incremented by 4 (width) after a transfer

Definition at line 198 of file dma\_5411x.h.

**6.17.2.35 #define DMA\_XFERCFG\_RELOAD (1 << 1)**

Indicates whether the channels control structure will be reloaded when the current descriptor is exhausted

Definition at line 183 of file dma\_5411x.h.

**6.17.2.36 #define DMA\_XFERCFG\_SETINTA (1 << 4)**

Set Interrupt flag A for this channel to fire when descriptor is complete

Definition at line 186 of file dma\_5411x.h.

**6.17.2.37 #define DMA\_XFERCFG\_SETINTB (1 << 5)**

Set Interrupt flag B for this channel to fire when descriptor is complete

Definition at line 187 of file dma\_5411x.h.

**6.17.2.38 #define DMA\_XFERCFG\_SRCINC\_0 (0 << 12)**

DMA source address is not incremented after a transfer

Definition at line 191 of file dma\_5411x.h.

**6.17.2.39 #define DMA\_XFERCFG\_SRCINC\_1 (1 << 12)**

DMA source address is incremented by 1 (width) after a transfer

Definition at line 192 of file dma\_5411x.h.

**6.17.2.40 #define DMA\_XFERCFG\_SRCINC\_2 (2 << 12)**

DMA source address is incremented by 2 (width) after a transfer

Definition at line 193 of file dma\_5411x.h.

**6.17.2.41 #define DMA\_XFERCFG\_SRCINC\_4 (3 << 12)**

DMA source address is incremented by 4 (width) after a transfer

Definition at line 194 of file dma\_5411x.h.

**6.17.2.42 #define DMA\_XFERCFG\_SWTRIG (1 << 2)**

Software Trigger

Definition at line 184 of file dma\_5411x.h.

**6.17.2.43 #define DMA\_XFERCFG\_WIDTH\_16 (1 << 8)**

16-bit transfers are performed

Definition at line 189 of file dma\_5411x.h.

**6.17.2.44 #define DMA\_XFERCFG\_WIDTH\_32 (2 << 8)**

32-bit transfers are performed

Definition at line 190 of file dma\_5411x.h.

**6.17.2.45 #define DMA\_XFERCFG\_WIDTH\_8 (0 << 8)**

8-bit transfers are performed

Definition at line 188 of file dma\_5411x.h.

**6.17.2.46 #define DMA\_XFERCFG\_XFERCOUNT( n ) ((n - 1) << 16)**

DMA transfer count in 'transfers', between (0)1 and (1023)1024

Definition at line 199 of file dma\_5411x.h.

**6.17.2.47 #define MAX\_DMA\_CHANNEL (20)**

Definition at line 83 of file dma\_5411x.h.

**6.17.3 Enumeration Type Documentation****6.17.3.1 enum DMA\_CHID\_T**

Enumerator

***DMA\_CH0***

***DMA\_CH1***

***DMA\_CH2***

*DMA\_CH3*  
*DMA\_CH4*  
*DMA\_CH5*  
*DMA\_CH6*  
*DMA\_CH7*  
*DMA\_CH8*  
*DMA\_CH9*  
*DMA\_CH10*  
*DMA\_CH11*  
*DMA\_CH12*  
*DMA\_CH13*  
*DMA\_CH14*  
*DMA\_CH15*  
*DMA\_CH16*  
*DMA\_CH17*  
*DMA\_CH18*  
*DMA\_CH19*  
*DMAREQ\_FLEXCOMM0\_RX*  
*DMAREQ\_FLEXCOMM0\_TX*  
*DMAREQ\_FLEXCOMM1\_RX*  
*DMAREQ\_FLEXCOMM1\_TX*  
*DMAREQ\_FLEXCOMM2\_RX*  
*DMAREQ\_FLEXCOMM2\_TX*  
*DMAREQ\_FLEXCOMM3\_RX*  
*DMAREQ\_FLEXCOMM3\_TX*  
*DMAREQ\_FLEXCOMM4\_RX*  
*DMAREQ\_FLEXCOMM4\_TX*  
*DMAREQ\_FLEXCOMM5\_RX*  
*DMAREQ\_FLEXCOMM5\_TX*  
*DMAREQ\_FLEXCOMM6\_RX*  
*DMAREQ\_FLEXCOMM6\_TX*  
*DMAREQ\_FLEXCOMM7\_RX*  
*DMAREQ\_FLEXCOMM7\_TX*  
*DMAREQ\_DMIC0*  
*DMAREQ\_DMIC1*  
*DMAREQ\_SPIFI*

Definition at line 100 of file dma\_5411x.h.



## 6.18 CHIP: LPC5411X DMA Service driver

### 6.18.1 Detailed Description

#### Data Structures

- struct [DMA\\_PERIPHERAL\\_CONTEXT\\_T](#)
- struct [DMA\\_DUAL\\_DESCRIPTOR\\_T](#)

#### Typedefs

- typedef void(\* [DMA\\_CALLBACK\\_T](#)) (int32\_t)

#### Functions

- void [Chip\\_DMASERVICE\\_Init](#) ([DMA\\_CHDESC\\_T](#) \*base)  
*Initialize DMA service.*
- void [Chip\\_DMASERVICE\\_Isr](#) (void)  
*DMA service interrupt handler.*
- void [Chip\\_DMASERVICE\\_RegisterCb](#) (const [DMA\\_PERIPHERAL\\_CONTEXT\\_T](#) \*pContext, [DMA\\_CALLBACK\\_T](#) pCallback)  
*Register callback function.*
- void [Chip\\_DMASERVICE\\_SingleBuffer](#) (const [DMA\\_PERIPHERAL\\_CONTEXT\\_T](#) \*pContext, uint32\_t pMem, uint32\_t length)  
*Use Single buffer mechanism.*
- void [Chip\\_DMASERVICE\\_DoubleBuffer](#) (const [DMA\\_PERIPHERAL\\_CONTEXT\\_T](#) \*pContext, uint32\_t pMem, uint32\_t length, [DMA\\_DUAL\\_DESCRIPTOR\\_T](#) \*pD)  
*Use double buffer mechanism.*

#### Variables

- uint32\_t [channel](#)
- volatile uint32\_t \* [register\\_location](#)
- uint32\_t [width](#)
- uint32\_t [src\\_increment](#)
- uint32\_t [dst\\_increment](#)
- bool [write](#)
- [DMA\\_CHDESC\\_T](#) [descr](#) [2]

### 6.18.2 Typedef Documentation

#### 6.18.2.1 typedef void(\* [DMA\\_CALLBACK\\_T](#)) (int32\_t)

Definition at line 44 of file `dma_service_5411x.h`.

### 6.18.3 Function Documentation

#### 6.18.3.1 void [Chip\\_DMASERVICE\\_DoubleBuffer](#) ( const [DMA\\_PERIPHERAL\\_CONTEXT\\_T](#) \* *pContext*, uint32\_t *pMem*, uint32\_t *length*, [DMA\\_DUAL\\_DESCRIPTOR\\_T](#) \* *pD* )

Use double buffer mechanism.

**Parameters**

<i>pContext</i>	: Pointer to peripheral channel context
<i>pMem</i>	: Pointer to data memory
<i>length</i>	: Transfer length in bytes
<i>pD</i>	: Pointer to dma dual descriptor

**Returns**

Nothing

Definition at line 135 of file dma\_service\_5411x.c.

**6.18.3.2 void Chip\_DMASERVICE\_Init ( DMA\_CHDESC\_T \* base )**

Initialize DMA service.

**Parameters**

<i>base</i>	: The base address where the DMA descriptors will be stored
-------------	---

**Returns**

Nothing

Definition at line 37 of file dma\_service\_5411x.c.

**6.18.3.3 void Chip\_DMASERVICE\_Isr ( void )**

DMA service interrupt handler.

**Parameters**

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

**Returns**

Nothing Must be called from DMA\_IRQHandler

Definition at line 62 of file dma\_service\_5411x.c.

**6.18.3.4 void Chip\_DMASERVICE\_RegisterCb ( const DMA\_PERIPHERAL\_CONTEXT\_T \* pContext, DMA\_CALLBACK\_T pCallback )**

Register callback function.

**Parameters**

<i>pContext</i>	: Pointer to peripheral channel context
<i>pCallback</i>	: Pointer to callback function

**Returns**

Nothing

Definition at line 89 of file dma\_service\_5411x.c.

6.18.3.5 void Chip\_DMASERVICE\_SingleBuffer ( const DMA\_PERIPHERAL\_CONTEXT\_T \* *pContext*, uint32\_t *pMem*,  
uint32\_t *length* )

Use Single buffer mechanism.

**Parameters**

<i>pContext</i>	: Pointer to peripheral channel context
<i>pMem</i>	: Pointer to data memory
<i>length</i>	: Transfer length in bytes

**Returns**

Nothing

Definition at line 99 of file dma\_service\_5411x.c.

## 6.18.4 Variable Documentation

### 6.18.4.1 uint32\_t channel

Definition at line 47 of file dma\_service\_5411x.h.

### 6.18.4.2 DMA\_CHDESC\_T descr[2]

Definition at line 56 of file dma\_service\_5411x.h.

### 6.18.4.3 uint32\_t dst\_increment

Definition at line 51 of file dma\_service\_5411x.h.

### 6.18.4.4 volatile uint32\_t\* register\_location

Definition at line 48 of file dma\_service\_5411x.h.

### 6.18.4.5 uint32\_t src\_increment

Definition at line 50 of file dma\_service\_5411x.h.

### 6.18.4.6 uint32\_t width

Definition at line 49 of file dma\_service\_5411x.h.

### 6.18.4.7 bool write

Definition at line 52 of file dma\_service\_5411x.h.

## 6.19 CHIP: LPC5411X DMIC driver

### 6.19.1 Detailed Description

#### Data Structures

- struct [LPC\\_DMIC\\_Channel\\_Type](#)
- struct [LPC\\_DMIC\\_T](#)
- struct [DMIC\\_STATISTICS\\_T](#)  
*DMIC statistics structure.*
- struct [DMIC\\_CHANNEL\\_CONFIG\\_T](#)

#### Macros

- `#define DMIC_FIFO_ENABLE_P 0`
- `#define DMIC_FIFO_RESETN_P 1`
- `#define DMIC_FIFO_INTREN_P 2`
- `#define DMIC_FIFO_DMAEN_P 3`
- `#define DMIC_FIFO_TLVL_P 16`
- `#define DMIC_FIFO_ENABLE (1<<DMIC_FIFO_ENABLE_P)`
- `#define DMIC_FIFO_RESETN (1<<DMIC_FIFO_RESETN_P)`
- `#define DMIC_FIFO_INTREN (1<<DMIC_FIFO_INTREN_P)`
- `#define DMIC_FIFO_DMAEN (1<<DMIC_FIFO_DMAEN_P)`
- `#define DMIC_FIFO_INT_P 0`
- `#define DMIC_FIFO_OVERRUN_P 1`
- `#define DMIC_FIFO_UNDERRUN_P 2`
- `#define DMIC_FIFO_INT (1<<DMIC_FIFO_INT_P)`
- `#define DMIC_FIFO_OVERRUN (1<<DMIC_FIFO_OVERRUN_P)`
- `#define DMIC_FIFO_UNDERRUN (1<<DMIC_FIFO_UNDERRUN_P)`
- `#define DMIC_PHY_FALL_P 0`
- `#define DMIC_PHY_HALF_P 1`
- `#define DMIC_PHY_FALL (1<<DMIC_PHY_FALL_P)`
- `#define DMIC_PHY_HALF (1<<DMIC_PHY_HALF_P)`
- `#define DMIC_DCPOLE_P 0`
- `#define DMIC_DCGAIN_REDUCE_P 4`
- `#define DMIC_SATURATE_AT16BIT_P 8`

#### Enumerations

- enum [OP\\_MODE\\_T](#) { [DMIC\\_OP\\_POLL](#), [DMIC\\_OP\\_INTR](#), [DMIC\\_OP\\_DMA](#) }
- enum [STEREO\\_SIDE\\_T](#) { [DMIC\\_LEFT](#) = 0, [DMIC\\_RIGHT](#) = 1 }
- enum [PDM\\_DIV\\_T](#) {  
[DMIC\\_PDM\\_DIV1](#) = 0, [DMIC\\_PDM\\_DIV2](#) = 1, [DMIC\\_PDM\\_DIV3](#) = 2, [DMIC\\_PDM\\_DIV4](#) = 3,  
[DMIC\\_PDM\\_DIV6](#) = 4, [DMIC\\_PDM\\_DIV8](#) = 5, [DMIC\\_PDM\\_DIV12](#) = 6, [DMIC\\_PDM\\_DIV16](#) = 7,  
[DMIC\\_PDM\\_DIV24](#) = 8, [DMIC\\_PDM\\_DIV32](#) = 9, [DMIC\\_PDM\\_DIV48](#) = 10, [DMIC\\_PDM\\_DIV64](#) = 11,  
[DMIC\\_PDM\\_DIV96](#) = 12, [DMIC\\_PDM\\_DIV128](#) = 13 }
- enum [COMPENSATION\\_T](#) { [DMIC\\_COMP0\\_0](#) = 0, [DMIC\\_COMP0\\_16](#) = 1, [DMIC\\_COMP0\\_15](#) = 2, [DMIC\\_COMP0\\_13](#) = 3 }
- enum [DC\\_REMOVAL\\_T](#) { [DMIC\\_DC\\_NOREMOVE](#) = 0, [DMIC\\_DC\\_CUT155](#) = 1, [DMIC\\_DC\\_CUT78](#) = 2, [DMIC\\_DC\\_CUT39](#) = 3 }
- enum [DMIC\\_IO\\_T](#) {  
[pdm\\_dual](#) = 0, [pdm\\_stereo](#) = 4, [pdm\\_bypass](#) = 3, [pdm\\_bypass\\_clk0](#) = 1,  
[pdm\\_bypass\\_clk1](#) = 2 }

## Functions

- void [Chip\\_DMIC\\_Init](#) (const [CHIP\\_SYSCON\\_CLOCK\\_T](#) clock, const [CHIP\\_SYSCON\\_PERIPH\\_RESET\\_T](#) reset)  
*Initialize DMIC interface.*
- void [Chip\\_DMIC\\_CfgIO](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, [DMIC\\_IO\\_T](#) cfg)  
*Configure DMIC io.*
- void [Chip\\_DMIC\\_SetOpMode](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, [OP\\_MODE\\_T](#) mode)  
*Set DMIC operating mode.*
- void [Chip\\_DMIC\\_CfgChannel](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, uint32\_t channel, [DMIC\\_CHANNEL\\_CONFIG\\_T](#) \*channel\_cfg)  
*Configure DMIC channel.*
- void [Chip\\_DMIC\\_CfgChannelDc](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, uint32\_t channel, [DC\\_REMOVAL\\_T](#) dc\_cut\_level, uint32\_t post\_dc\_gain\_reduce, bool saturate16bit)  
*Configure DMIC channel DC removal setting.*
- void [Chip\\_DMIC\\_Use2fs](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, bool use2fs)  
*Configure Clock scaling.*
- void [Chip\\_DMIC\\_EnableChannel](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, uint32\_t channelmask)  
*Configure Clock scaling.*
- void [Chip\\_DMIC\\_FifoChannel](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, uint32\_t channel, uint32\_t trig\_level, uint32\_t enable, uint32\_t resetn)  
*Configure fifo settings for DMIC channel.*
- [\\_\\_STATIC\\_INLINE](#) uint32\_t [Chip\\_DMIC\\_FifoGetStatus](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, uint32\_t channel)  
*Get FIFO status.*
- [\\_\\_STATIC\\_INLINE](#) void [Chip\\_DMIC\\_FifoClearStatus](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, uint32\_t channel, uint32\_t mask)  
*Clear FIFO status.*
- [\\_\\_STATIC\\_INLINE](#) uint32\_t [Chip\\_DMIC\\_FifoGetData](#) ([LPC\\_DMIC\\_T](#) \*pDMIC, uint32\_t channel)  
*Get FIFO data.*

## Variables

- [\\_\\_IO](#) uint32\_t [OSR](#)
- [\\_\\_IO](#) uint32\_t [DIVHFCLK](#)
- [\\_\\_IO](#) uint32\_t [PREAC2FSCOEF](#)
- [\\_\\_IO](#) uint32\_t [PREAC4FSCOEF](#)
- [\\_\\_IO](#) uint32\_t [GAINSHFT](#)
- [\\_\\_IO](#) uint32\_t [TDM96EN](#)
- [\\_\\_IO](#) uint32\_t [TDM19EN](#)
- [\\_\\_IO](#) uint32\_t [reserved](#) [25]
- [\\_\\_IO](#) uint32\_t [FIFO\\_CTRL](#)
- [\\_\\_IO](#) uint32\_t [FIFO\\_STATUS](#)
- [\\_\\_IO](#) uint32\_t [FIFO\\_DATA](#)
- [\\_\\_IO](#) uint32\_t [PHY\\_CTRL](#)
- [\\_\\_IO](#) uint32\_t [DC\\_CTRL](#)
- [\\_\\_IO](#) uint32\_t [reserved1](#) [27]
- [\\_\\_IO](#) [LPC\\_DMIC\\_Channel\\_Type](#) [CHANNEL](#) [15]
- [\\_\\_IO](#) uint32\_t [CHANEN](#)
- [\\_\\_IO](#) uint32\_t [reserved0](#) [2]
- [\\_\\_IO](#) uint32\_t [IOCFG](#)
- [\\_\\_IO](#) uint32\_t [USE2FS](#)
- [\\_\\_IO](#) uint32\_t [reserved](#) [27]
- [\\_\\_IO](#) uint32\_t [HWVADGAIN](#)

- [\\_\\_IO uint32\\_t HWVADHPFS](#)
- [\\_\\_IO uint32\\_t HWVADST10](#)
- [\\_\\_IO uint32\\_t HWVADRSTT](#)
- [\\_\\_IO uint32\\_t HWVADTHGN](#)
- [\\_\\_IO uint32\\_t HWVADTHGS](#)
- [\\_\\_IO uint32\\_t HWVADLOWZ](#)
- [\\_\\_IO uint32\\_t reserved1 \[24\]](#)
- [\\_\\_O uint32\\_t ID](#)
- [uint32\\_t fifo\\_ints](#)
- [uint32\\_t fifo\\_overrun](#)
- [uint32\\_t fifo\\_underrun](#)
- [STEREO\\_SIDE\\_T side](#)
- [PDM\\_DIV\\_T divhclk](#)
- [uint32\\_t osr](#)
- [int32\\_t gainshft](#)
- [COMPENSATION\\_T preac2coef](#)
- [COMPENSATION\\_T preac4coef](#)
- [DMA\\_PERIPHERAL\\_CONTEXT\\_T dmic\\_ch0\\_dma\\_context](#)
- [DMA\\_PERIPHERAL\\_CONTEXT\\_T dmic\\_ch1\\_dma\\_context](#)
- [DMA\\_PERIPHERAL\\_CONTEXT\\_T dmic\\_ch0\\_dma\\_interleaved\\_context](#)
- [DMA\\_PERIPHERAL\\_CONTEXT\\_T dmic\\_ch1\\_dma\\_interleaved\\_context](#)

## 6.19.2 Macro Definition Documentation

### 6.19.2.1 #define DMIC\_DCGAIN\_REDUCE\_P 4

Definition at line 137 of file `dmic_5411x.h`.

### 6.19.2.2 #define DMIC\_DCPOLE\_P 0

Definition at line 136 of file `dmic_5411x.h`.

### 6.19.2.3 #define DMIC\_FIFO\_DMAEN (1<<DMIC\_FIFO\_DMAEN\_P)

Definition at line 116 of file `dmic_5411x.h`.

### 6.19.2.4 #define DMIC\_FIFO\_DMAEN\_P 3

Definition at line 109 of file `dmic_5411x.h`.

### 6.19.2.5 #define DMIC\_FIFO\_ENABLE (1<<DMIC\_FIFO\_ENABLE\_P)

Definition at line 113 of file `dmic_5411x.h`.

### 6.19.2.6 #define DMIC\_FIFO\_ENABLE\_P 0

Definition at line 106 of file `dmic_5411x.h`.

### 6.19.2.7 #define DMIC\_FIFO\_INT (1<<DMIC\_FIFO\_INT\_P)

Definition at line 123 of file `dmic_5411x.h`.

**6.19.2.8 #define DMIC\_FIFO\_INT\_P 0**

Definition at line 119 of file dmic\_5411x.h.

**6.19.2.9 #define DMIC\_FIFO\_INTREN (1<<DMIC\_FIFO\_INTREN\_P)**

Definition at line 115 of file dmic\_5411x.h.

**6.19.2.10 #define DMIC\_FIFO\_INTREN\_P 2**

Definition at line 108 of file dmic\_5411x.h.

**6.19.2.11 #define DMIC\_FIFO\_OVERRUN (1<<DMIC\_FIFO\_OVERRUN\_P)**

Definition at line 124 of file dmic\_5411x.h.

**6.19.2.12 #define DMIC\_FIFO\_OVERRUN\_P 1**

Definition at line 120 of file dmic\_5411x.h.

**6.19.2.13 #define DMIC\_FIFO\_RESETN (1<<DMIC\_FIFO\_RESETN\_P)**

Definition at line 114 of file dmic\_5411x.h.

**6.19.2.14 #define DMIC\_FIFO\_RESETN\_P 1**

Definition at line 107 of file dmic\_5411x.h.

**6.19.2.15 #define DMIC\_FIFO\_TLVL\_P 16**

Definition at line 111 of file dmic\_5411x.h.

**6.19.2.16 #define DMIC\_FIFO\_UNDERRUN (1<<DMIC\_FIFO\_UNDERRUN\_P)**

Definition at line 125 of file dmic\_5411x.h.

**6.19.2.17 #define DMIC\_FIFO\_UNDERRUN\_P 2**

Definition at line 121 of file dmic\_5411x.h.

**6.19.2.18 #define DMIC\_PHY\_FALL (1<<DMIC\_PHY\_FALL\_P)**

Definition at line 131 of file dmic\_5411x.h.

**6.19.2.19 #define DMIC\_PHY\_FALL\_P 0**

Definition at line 128 of file dmic\_5411x.h.



6.19.2.20 `#define DMIC_PHY_HALF (1<<DMIC_PHY_HALF_P)`

Definition at line 132 of file `dmic_5411x.h`.

6.19.2.21 `#define DMIC_PHY_HALF_P 1`

Definition at line 129 of file `dmic_5411x.h`.

6.19.2.22 `#define DMIC_SATURATE_AT16BIT_P 8`

Definition at line 138 of file `dmic_5411x.h`.

### 6.19.3 Enumeration Type Documentation

6.19.3.1 `enum COMPENSATION_T`

Enumerator

***DMIC\_COMP0\_0***

***DMIC\_COMP0\_16***

***DMIC\_COMP0\_15***

***DMIC\_COMP0\_13***

Definition at line 170 of file `dmic_5411x.h`.

6.19.3.2 `enum DC_REMOVAL_T`

Enumerator

***DMIC\_DC\_NOREMOVE***

***DMIC\_DC\_CUT155***

***DMIC\_DC\_CUT78***

***DMIC\_DC\_CUT39***

Definition at line 177 of file `dmic_5411x.h`.

6.19.3.3 `enum DMIC_IO_T`

Enumerator

***pdm\_dual*** Two separate pairs of PDM wires

***pdm\_stereo*** Stereo Mic

***pdm\_bypass*** Clk Bypas clocks both channels

***pdm\_bypass\_clk0*** Clk Bypas clocks only channel0

***pdm\_bypass\_clk1*** Clk Bypas clocks only channel1

Definition at line 184 of file `dmic_5411x.h`.

#### 6.19.3.4 enum OP\_MODE\_T

Enumerator

***DMIC\_OP\_POLL***

***DMIC\_OP\_INTR***

***DMIC\_OP\_DMA***

Definition at line 142 of file dmic\_5411x.h.

#### 6.19.3.5 enum PDM\_DIV\_T

Enumerator

***DMIC\_PDM\_DIV1***

***DMIC\_PDM\_DIV2***

***DMIC\_PDM\_DIV3***

***DMIC\_PDM\_DIV4***

***DMIC\_PDM\_DIV6***

***DMIC\_PDM\_DIV8***

***DMIC\_PDM\_DIV12***

***DMIC\_PDM\_DIV16***

***DMIC\_PDM\_DIV24***

***DMIC\_PDM\_DIV32***

***DMIC\_PDM\_DIV48***

***DMIC\_PDM\_DIV64***

***DMIC\_PDM\_DIV96***

***DMIC\_PDM\_DIV128***

Definition at line 153 of file dmic\_5411x.h.

#### 6.19.3.6 enum STEREO\_SIDE\_T

Enumerator

***DMIC\_LEFT***

***DMIC\_RIGHT***

Definition at line 148 of file dmic\_5411x.h.

### 6.19.4 Function Documentation

#### 6.19.4.1 void Chip\_DMIC\_CfgChannel ( LPC\_DMIC\_T \* pDMIC, uint32\_t channel, DMIC\_CHANNEL\_CONFIG\_T \* channel\_cfg )

Configure DMIC channel.

Parameters

---

<i>pDMIC</i>	: The base address of DMIC interface
<i>channel</i>	: DMIC channel
<i>channel_cfg</i>	: Channel configuration

**Returns**

Nothing

Definition at line 109 of file dmic\_5411x.c.

6.19.4.2 void Chip\_DMIC\_CfgChannelDc ( LPC\_DMIC\_T \* *pDMIC*, uint32\_t *channel*, DC\_REMOVAL\_T *dc\_cut\_level*, uint32\_t *post\_dc\_gain\_reduce*, bool *saturate16bit* )

Configure DMIC channel DC removal setting.

**Parameters**

<i>pDMIC</i>	: The base address of DMIC interface
<i>channel</i>	: DMIC channel
<i>dc_cut_level</i>	: DC cut level
<i>post_dc_gain_reduce</i>	: Post DC cut gain adjustment
<i>saturate16bit</i>	: Saturation setting

**Returns**

Nothing

Definition at line 120 of file dmic\_5411x.c.

6.19.4.3 void Chip\_DMIC\_CfgIO ( LPC\_DMIC\_T \* *pDMIC*, DMIC\_IO\_T *cfg* )

Configure DMIC io.

**Parameters**

<i>pDMIC</i>	: The base address of DMIC interface
<i>cfg</i>	: DMIC io configuration

**Returns**

Nothing

Definition at line 86 of file dmic\_5411x.c.

6.19.4.4 void Chip\_DMIC\_EnableChannel ( LPC\_DMIC\_T \* *pDMIC*, uint32\_t *channelmask* )

Configure Clock scaling.

**Parameters**

<i>pDMIC</i>	: The base address of DMIC interface
<i>channelmask</i>	: Channel selection

**Returns**

Nothing

Definition at line 132 of file dmic\_5411x.c.

6.19.4.5 void Chip\_DMIC\_FifoChannel ( LPC\_DMIC\_T \* *pDMIC*, uint32\_t *channel*, uint32\_t *trig\_level*, uint32\_t *enable*,  
uint32\_t *resetn* )

Configure fifo settings for DMIC channel.

## Parameters

<i>pDMIC</i>	: The base address of DMIC interface
<i>channel</i>	: DMIC channel
<i>trig_level</i>	: FIFO trigger level
<i>enable</i>	: FIFO level
<i>resetrn</i>	: FIFO reset

## Returns

Nothing

Definition at line 138 of file `dmic_5411x.c`.

6.19.4.6 `__STATIC_INLINE void Chip_DMIC_FifoClearStatus ( LPC_DMIC_T * pDMIC, uint32_t channel, uint32_t mask )`

Clear FIFO status.

## Parameters

<i>pDMIC</i>	: The base address of DMIC interface
<i>channel</i>	: DMIC channel
<i>mask</i>	: Bits to be cleared

## Returns

FIFO status

Definition at line 290 of file `dmic_5411x.h`.

6.19.4.7 `__STATIC_INLINE uint32_t Chip_DMIC_FifoGetData ( LPC_DMIC_T * pDMIC, uint32_t channel )`

Get FIFO data.

## Parameters

<i>pDMIC</i>	: The base address of DMIC interface
<i>channel</i>	: DMIC channel

## Returns

FIFO data

Definition at line 301 of file `dmic_5411x.h`.

6.19.4.8 `__STATIC_INLINE uint32_t Chip_DMIC_FifoGetStatus ( LPC_DMIC_T * pDMIC, uint32_t channel )`

Get FIFO status.

## Parameters

<i>pDMIC</i>	: The base address of DMIC interface
<i>channel</i>	: DMIC channel

## Returns

FIFO status

Definition at line 278 of file `dmic_5411x.h`.

6.19.4.9 void Chip\_DMIC\_Init ( const CHIP\_SYSCON\_CLOCK\_T *clock*, const CHIP\_SYSCON\_PERIPH\_RESET\_T *reset* )

Initialize DMIC interface.

## Parameters

<i>clock</i>	: DMIC clock assignment
<i>reset</i>	: DMIC reset assignment

## Returns

Nothing

Definition at line 76 of file `dmic_5411x.c`.

#### 6.19.4.10 void Chip\_DMIC\_SetOpMode ( LPC\_DMIC\_T \* *pDMIC*, OP\_MODE\_T *mode* )

Set DMIC operating mode.

## Parameters

<i>pDMIC</i>	: The base address of DMIC interface
<i>mode</i>	: DMIC mode

## Returns

Nothing

Definition at line 92 of file `dmic_5411x.c`.

#### 6.19.4.11 void Chip\_DMIC\_Use2fs ( LPC\_DMIC\_T \* *pDMIC*, bool *use2fs* )

Configure Clock scaling.

## Parameters

<i>pDMIC</i>	: The base address of DMIC interface
<i>use2fs</i>	: clock scaling

## Returns

Nothing

Definition at line 126 of file `dmic_5411x.c`.

### 6.19.5 Variable Documentation

#### 6.19.5.1 \_\_IO uint32\_t CHANEN

Definition at line 65 of file `dmic_5411x.h`.

#### 6.19.5.2 \_\_IO LPC\_DMIC\_Channel\_Type CHANNEL[15]

Definition at line 63 of file `dmic_5411x.h`.

#### 6.19.5.3 \_\_IO uint32\_t DC\_CTRL

Definition at line 57 of file `dmic_5411x.h`.

#### 6.19.5.4 `__IO uint32_t DIVHFCLK`

Definition at line 46 of file `dmic_5411x.h`.

#### 6.19.5.5 `PDM_DIV_T divhfclk`

Definition at line 194 of file `dmic_5411x.h`.

#### 6.19.5.6 `DMA_PERIPHERAL_CONTEXT_T dmic_ch0_dma_context`

Definition at line 35 of file `dmic_5411x.c`.

#### 6.19.5.7 `DMA_PERIPHERAL_CONTEXT_T dmic_ch0_dma_interleaved_context`

Definition at line 56 of file `dmic_5411x.c`.

#### 6.19.5.8 `DMA_PERIPHERAL_CONTEXT_T dmic_ch1_dma_context`

Definition at line 45 of file `dmic_5411x.c`.

#### 6.19.5.9 `DMA_PERIPHERAL_CONTEXT_T dmic_ch1_dma_interleaved_context`

Definition at line 66 of file `dmic_5411x.c`.

#### 6.19.5.10 `__IO uint32_t FIFO_CTRL`

Definition at line 53 of file `dmic_5411x.h`.

#### 6.19.5.11 `__IO uint32_t FIFO_DATA`

Definition at line 55 of file `dmic_5411x.h`.

#### 6.19.5.12 `uint32_t fifo_ints`

count: FIFO interrupts

Definition at line 88 of file `dmic_5411x.h`.

#### 6.19.5.13 `uint32_t fifo_overrun`

count: FIFO over-run errors

Definition at line 89 of file `dmic_5411x.h`.

#### 6.19.5.14 `__IO uint32_t FIFO_STATUS`

Definition at line 54 of file `dmic_5411x.h`.



**6.19.5.15 uint32\_t fifo\_underrun**

count: FIFO under-run errors

Definition at line 90 of file dmics\_5411x.h.

**6.19.5.16 \_\_IO uint32\_t GAINSHFT**

Definition at line 49 of file dmics\_5411x.h.

**6.19.5.17 int32\_t gainshft**

Definition at line 196 of file dmics\_5411x.h.

**6.19.5.18 \_\_IO uint32\_t HWVADGAIN**

Definition at line 71 of file dmics\_5411x.h.

**6.19.5.19 \_\_IO uint32\_t HWVADHPFS**

Definition at line 72 of file dmics\_5411x.h.

**6.19.5.20 \_\_IO uint32\_t HWVADLOWZ**

Definition at line 77 of file dmics\_5411x.h.

**6.19.5.21 \_\_IO uint32\_t HWVADRSTT**

Definition at line 74 of file dmics\_5411x.h.

**6.19.5.22 \_\_IO uint32\_t HWVADST10**

Definition at line 73 of file dmics\_5411x.h.

**6.19.5.23 \_\_IO uint32\_t HWVADTHGN**

Definition at line 75 of file dmics\_5411x.h.

**6.19.5.24 \_\_IO uint32\_t HWVADTHGS**

Definition at line 76 of file dmics\_5411x.h.

**6.19.5.25 \_\_O uint32\_t ID**

Definition at line 79 of file dmics\_5411x.h.

**6.19.5.26 \_\_IO uint32\_t IOCFG**

Definition at line 67 of file dmics\_5411x.h.

**6.19.5.27   \_\_IO uint32\_t OSR**

Definition at line 45 of file dmic\_5411x.h.

**6.19.5.28   uint32\_t osr**

Definition at line 195 of file dmic\_5411x.h.

**6.19.5.29   \_\_IO uint32\_t PHY\_CTRL**

Definition at line 56 of file dmic\_5411x.h.

**6.19.5.30   COMPENSATION\_T preac2coef**

Definition at line 197 of file dmic\_5411x.h.

**6.19.5.31   \_\_IO uint32\_t PREAC2FSCOE**

Definition at line 47 of file dmic\_5411x.h.

**6.19.5.32   COMPENSATION\_T preac4coef**

Definition at line 198 of file dmic\_5411x.h.

**6.19.5.33   \_\_IO uint32\_t PREAC4FSCOE**

Definition at line 48 of file dmic\_5411x.h.

**6.19.5.34   \_\_IO uint32\_t reserved[25]**

Definition at line 52 of file dmic\_5411x.h.

**6.19.5.35   \_\_IO uint32\_t reserved[27]**

Definition at line 69 of file dmic\_5411x.h.

**6.19.5.36   \_\_IO uint32\_t reserved0[2]**

Definition at line 66 of file dmic\_5411x.h.

**6.19.5.37   \_\_IO uint32\_t reserved1[27]**

Definition at line 58 of file dmic\_5411x.h.

**6.19.5.38   \_\_IO uint32\_t reserved1[24]**

Definition at line 78 of file dmic\_5411x.h.

**6.19.5.39 STEREO\_SIDE\_T side**

Definition at line 193 of file dmic\_5411x.h.

**6.19.5.40 \_\_IO uint32\_t TDM19EN**

Definition at line 51 of file dmic\_5411x.h.

**6.19.5.41 \_\_IO uint32\_t TDM96EN**

Definition at line 50 of file dmic\_5411x.h.

**6.19.5.42 \_\_IO uint32\_t USE2FS**

Definition at line 68 of file dmic\_5411x.h.

## 6.20 CHIP: LPC5411X Enhanced boot block support

### 6.20.1 Detailed Description

#### Data Structures

- struct [PINTABLE\\_T](#)

*LPC5411X Pin table structure used for enhanced boot block support.*

#### Macros

- #define [IMAGE\\_ENH\\_MARKER\\_OFF](#) 0x24
- #define [IMAGE\\_SINGLE\\_ENH\\_SIG](#) 0xEDDC9494
- #define [IMAGE\\_DUAL\\_ENH\\_SIG](#) 0x0FFEB6B6
- #define [IMAGE\\_BOOT\\_BLOCK\\_OFF](#) 0x28
- #define [IMAGE\\_ENH\\_BLOCK\\_MARKER](#) 0xFEEDA5A5
- #define [SETPORTPIN](#)(port, pin) (((port) & 0x7) << 5) | ((pin) & 0x1F)

#### Enumerations

- enum [IMAGE\\_T](#) {  
[IMG\\_NORMAL](#) = 0, [IMG\\_ISP\\_WAIT](#), [IMG\\_NO\\_WAIT](#), [IMG\\_NO\\_CRC](#),  
[IMG\\_JUST\\_BOOT](#) = 0xFF }
- enum [IFSEL\\_T](#) {  
[SL\\_AUTO](#) = 0, [SL\\_I2C0](#), [SL\\_I2C1](#), [SL\\_I2C2](#),  
[SL\\_SPI0](#), [SL\\_SPI1](#) }

### 6.20.2 Macro Definition Documentation

#### 6.20.2.1 #define IMAGE\_BOOT\_BLOCK\_OFF 0x28

Definition at line 54 of file pintable\_5411x.h.

#### 6.20.2.2 #define IMAGE\_DUAL\_ENH\_SIG 0x0FFEB6B6

Definition at line 51 of file pintable\_5411x.h.

#### 6.20.2.3 #define IMAGE\_ENH\_BLOCK\_MARKER 0xFEEDA5A5

Definition at line 57 of file pintable\_5411x.h.

#### 6.20.2.4 #define IMAGE\_ENH\_MARKER\_OFF 0x24

Definition at line 45 of file pintable\_5411x.h.

#### 6.20.2.5 #define IMAGE\_SINGLE\_ENH\_SIG 0xEDDC9494

Definition at line 48 of file pintable\_5411x.h.

6.20.2.6 `#define SETPORTPIN( port, pin ) (((port) & 0x7) << 5) | ((pin) & 0x1F)`

Definition at line 60 of file pintable\_5411x.h.

### 6.20.3 Enumeration Type Documentation

#### 6.20.3.1 enum IFSEL\_T

Host interface sources (ifSel) for the pin table structure

Enumerator

**SL\_AUTO** Auto-detect used for host interface

**SL\_I2C0** I2C0 used for host interface

**SL\_I2C1** I2C1 used for host interface

**SL\_I2C2** I2C2 used for host interface

**SL\_SPI0** SPI0 used for host interface

**SL\_SPI1** SPI1 used for host interface

Definition at line 72 of file pintable\_5411x.h.

#### 6.20.3.2 enum IMAGE\_T

Image type (img\_type) for the pin table structure

Enumerator

**IMG\_NORMAL** Normal image check, assert dynamic ISP to halt boot

**IMG\_ISP\_WAIT** Wait for host system to send SH\_CMD\_BOOT command

**IMG\_NO\_WAIT** Boot image without checking dynamic ISP, CRC is still done

**IMG\_NO\_CRC** No CRC check made. Used during development. Dynamic ISP still works

**IMG\_JUST\_BOOT** Disables XOR and CRC checks, will always boot

Definition at line 63 of file pintable\_5411x.h.

## 6.21 CHIP: LPC5411X GPIO driver

### 6.21.1 Detailed Description

#### Data Structures

- struct [LPC\\_GPIO\\_T](#)  
*GPIO port register block structure.*

#### Functions

- `__STATIC_INLINE void Chip\_GPIO\_Init (LPC\_GPIO\_T *pGPIO)`  
*Initialize GPIO block.*
- `__STATIC_INLINE void Chip\_GPIO\_DeInit (LPC\_GPIO\_T *pGPIO)`  
*De-Initialize GPIO block.*
- `__STATIC_INLINE void Chip\_GPIO\_WritePortBit (LPC\_GPIO\_T *pGPIO, uint32_t port, uint8_t pin, bool setting)`  
*Set a GPIO port/pin state.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPinState (LPC\_GPIO\_T *pGPIO, uint8_t port, uint8_t pin, bool setting)`  
*Set a GPIO pin state via the GPIO byte register.*
- `__STATIC_INLINE bool Chip\_GPIO\_ReadPortBit (LPC\_GPIO\_T *pGPIO, uint32_t port, uint8_t pin)`  
*Read a GPIO pin state via the GPIO byte register.*
- `__STATIC_INLINE bool Chip\_GPIO\_GetPinState (LPC\_GPIO\_T *pGPIO, uint8_t port, uint8_t pin)`  
*Get a GPIO pin state via the GPIO byte register.*
- `__STATIC_INLINE void Chip\_GPIO\_WriteDirBit (LPC\_GPIO\_T *pGPIO, uint32_t port, uint8_t pin, bool setting)`  
*Set GPIO direction for a single GPIO pin.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPinDIROutput (LPC\_GPIO\_T *pGPIO, uint8_t port, uint8_t pin)`  
*Set GPIO direction for a single GPIO pin to an output.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPinDIRInput (LPC\_GPIO\_T *pGPIO, uint8_t port, uint8_t pin)`  
*Set GPIO direction for a single GPIO pin to an input.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPinDIR (LPC\_GPIO\_T *pGPIO, uint8_t port, uint8_t pin, bool output)`  
*Set GPIO direction for a single GPIO pin.*
- `__STATIC_INLINE bool Chip\_GPIO\_ReadDirBit (LPC\_GPIO\_T *pGPIO, uint32_t port, uint8_t bit)`  
*Read a GPIO direction (out or in)*
- `__STATIC_INLINE bool Chip\_GPIO\_GetPinDIR (LPC\_GPIO\_T *pGPIO, uint8_t port, uint8_t pin)`  
*Get GPIO direction for a single GPIO pin.*
- `__STATIC_INLINE void Chip\_GPIO\_SetDir (LPC\_GPIO\_T *pGPIO, uint8_t portNum, uint32_t bitValue, uint8_t out)`  
*Set Direction for a GPIO port.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPortDIROutput (LPC\_GPIO\_T *pGPIO, uint8_t port, uint32_t pinMask)`  
*Set GPIO direction for a all selected GPIO pins to an output.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPortDIRInput (LPC\_GPIO\_T *pGPIO, uint8_t port, uint32_t pinMask)`  
*Set GPIO direction for a all selected GPIO pins to an input.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPortDIR (LPC\_GPIO\_T *pGPIO, uint8_t port, uint32_t pinMask, bool outSet)`  
*Set GPIO direction for a all selected GPIO pins to an input or output.*
- `__STATIC_INLINE uint32_t Chip\_GPIO\_GetPortDIR (LPC\_GPIO\_T *pGPIO, uint8_t port)`

- Get GPIO direction for a all GPIO pins.*
- `__STATIC_INLINE void Chip_GPIO_SetPortMask (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t mask)`  
*Set GPIO port mask value for GPIO masked read and write.*
  - `__STATIC_INLINE uint32_t Chip_GPIO_GetPortMask (LPC_GPIO_T *pGPIO, uint8_t port)`  
*Get GPIO port mask value used for GPIO masked read and write.*
  - `__STATIC_INLINE void Chip_GPIO_SetPortValue (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t value)`  
*Set all GPIO raw pin states (regardless of masking)*
  - `__STATIC_INLINE uint32_t Chip_GPIO_GetPortValue (LPC_GPIO_T *pGPIO, uint8_t port)`  
*Get all GPIO raw pin states (regardless of masking)*
  - `__STATIC_INLINE void Chip_GPIO_SetMaskedPortValue (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t value)`  
*Set all GPIO pin states, but mask via the MASKP0 register.*
  - `__STATIC_INLINE uint32_t Chip_GPIO_GetMaskedPortValue (LPC_GPIO_T *pGPIO, uint8_t port)`  
*Get all GPIO pin statesm but mask via the MASKP0 register.*
  - `__STATIC_INLINE void Chip_GPIO_SetValue (LPC_GPIO_T *pGPIO, uint8_t portNum, uint32_t bitValue)`  
*Set a GPIO port/bit to the high state.*
  - `__STATIC_INLINE void Chip_GPIO_SetPortOutHigh (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pins)`  
*Set selected GPIO output pins to the high state.*
  - `__STATIC_INLINE void Chip_GPIO_SetPinOutHigh (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`  
*Set an individual GPIO output pin to the high state.*
  - `__STATIC_INLINE void Chip_GPIO_ClearValue (LPC_GPIO_T *pGPIO, uint8_t portNum, uint32_t bitValue)`  
*Set a GPIO port/bit to the low state.*
  - `__STATIC_INLINE void Chip_GPIO_SetPortOutLow (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pins)`  
*Set selected GPIO output pins to the low state.*
  - `__STATIC_INLINE void Chip_GPIO_SetPinOutLow (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`  
*Set an individual GPIO output pin to the low state.*
  - `__STATIC_INLINE void Chip_GPIO_SetPortToggle (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pins)`  
*Toggle selected GPIO output pins to the opposite state.*
  - `__STATIC_INLINE void Chip_GPIO_SetPinToggle (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`  
*Toggle an individual GPIO output pin to the opposite state.*
  - `__STATIC_INLINE uint32_t Chip_GPIO_ReadValue (LPC_GPIO_T *pGPIO, uint8_t portNum)`  
*Read current bit states for the selected port.*

## 6.21.2 Function Documentation

### 6.21.2.1 `__STATIC_INLINE void Chip_GPIO_ClearValue ( LPC_GPIO_T * pGPIO, uint8_t portNum, uint32_t bitValue )`

Set a GPIO port/bit to the low state.

#### Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>portNum</i>	: port number (support port 0 only)
<i>bitValue</i>	: bit(s) in the port to set low

#### Returns

None

#### Note

Any bit set as a '0' will not have it's state changed. This only applies to ports configured as an output. It is recommended to use the `Chip_GPIO_SetPortOutLow()` function instead.

Definition at line 432 of file `gpio_5411x.h`.

6.21.2.2 `__STATIC_INLINE void Chip_GPIO_DeInit ( LPC_GPIO_T * pGPIO )`

De-Initialize GPIO block.



## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip (not used)
--------------	--

## Returns

Nothing

Definition at line 79 of file gpio\_5411x.h.

### 6.21.2.3 `__STATIC_INLINE uint32_t Chip_GPIO_GetMaskedPortValue ( LPC_GPIO_T * pGPIO, uint8_t port )`

Get all GPIO pin statesm but mask via the MASKP0 register.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)

## Returns

Current (masked) state of all GPIO pins

Definition at line 374 of file gpio\_5411x.h.

### 6.21.2.4 `__STATIC_INLINE bool Chip_GPIO_GetPinDIR ( LPC_GPIO_T * pGPIO, uint8_t port, uint8_t pin )`

Get GPIO direction for a single GPIO pin.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to read (supports port 0 only)
<i>pin</i>	: GPIO pin to get direction for

## Returns

true if the GPIO is an output, false if input

Definition at line 222 of file gpio\_5411x.h.

### 6.21.2.5 `__STATIC_INLINE bool Chip_GPIO_GetPinState ( LPC_GPIO_T * pGPIO, uint8_t port, uint8_t pin )`

Get a GPIO pin state via the GPIO byte register.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to read
<i>pin</i>	: GPIO pin to get state for

## Returns

true if the GPIO is high, false if low

## Note

This function replaces [Chip\\_GPIO\\_ReadPortBit\(\)](#)

Definition at line 134 of file gpio\_5411x.h.

6.21.2.6 `__STATIC_INLINE uint32_t Chip_GPIO_GetPortDIR ( LPC_GPIO_T * pGPIO, uint8_t port )`

Get GPIO direction for a all GPIO pins.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number

## Returns

a bitfield containing the input and output states for each pin

## Note

For pins 0..n, a high state in a bit corresponds to an output state for the same pin, while a low state corresponds to an input state.

Definition at line 303 of file gpio\_5411x.h.

#### 6.21.2.7 `__STATIC_INLINE uint32_t Chip_GPIO_GetPortMask ( LPC_GPIO_T * pGPIO, uint8_t port )`

Get GPIO port mask value used for GPIO masked read and write.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)

## Returns

Returns value set with the [Chip\\_GPIO\\_SetPortMask\(\)](#) function.

Definition at line 328 of file gpio\_5411x.h.

#### 6.21.2.8 `__STATIC_INLINE uint32_t Chip_GPIO_GetPortValue ( LPC_GPIO_T * pGPIO, uint8_t port )`

Get all GPIO raw pin states (regardless of masking)

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)

## Returns

Current (raw) state of all GPIO pins

Definition at line 351 of file gpio\_5411x.h.

#### 6.21.2.9 `__STATIC_INLINE void Chip_GPIO_Init ( LPC_GPIO_T * pGPIO )`

Initialize GPIO block.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip (not used)
--------------	--

## Returns

Nothing

**Note**

*pGPIO* parameter is ignored and all available GPIOs will be turned on by default.

Definition at line 66 of file gpio\_5411x.h.

6.21.2.10 `__STATIC_INLINE bool Chip_GPIO_ReadDirBit ( LPC_GPIO_T * pGPIO, uint32_t port, uint8_t bit )`

Read a GPIO direction (out or in)

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to read
<i>bit</i>	: GPIO bit direction to read

**Returns**

true if the GPIO is an output, false if input

**Note**

It is recommended to use the [Chip\\_GPIO\\_GetPinDIR\(\)](#) function instead.

Definition at line 210 of file gpio\_5411x.h.

6.21.2.11 `__STATIC_INLINE bool Chip_GPIO_ReadPortBit ( LPC_GPIO_T * pGPIO, uint32_t port, uint8_t pin )`

Read a GPIO pin state via the GPIO byte register.

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to read
<i>pin</i>	: GPIO pin to read

**Returns**

true if the GPIO pin is high, false if low

**Note**

It is recommended to use the [Chip\\_GPIO\\_GetPinState\(\)](#) function instead.

Definition at line 121 of file gpio\_5411x.h.

6.21.2.12 `__STATIC_INLINE uint32_t Chip_GPIO_ReadValue ( LPC_GPIO_T * pGPIO, uint8_t portNum )`

Read current bit states for the selected port.

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
--------------	---

<i>portNum</i>	: port number to read (supports port 0 only)
----------------	--

**Returns**

Current value of GPIO port

**Note**

The current states of the bits for the port are read, regardless of whether the GPIO port bits are input or output. It is recommended to use the [Chip\\_GPIO\\_GetPortValue\(\)](#) function instead.

Definition at line 502 of file gpio\_5411x.h.

6.21.2.13 `__STATIC_INLINE void Chip_GPIO_SetDir ( LPC_GPIO_T * pGPIO, uint8_t portNum, uint32_t bitValue, uint8_t out )`

Set Direction for a GPIO port.

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>portNum</i>	: port Number
<i>bitValue</i>	: GPIO bit to set
<i>out</i>	: Direction value, 0 = input, !0 = output

**Returns**

None

**Note**

Bits set to '0' are not altered. It is recommended to use the [Chip\\_GPIO\\_SetPortDIR\(\)](#) function instead.

Definition at line 237 of file gpio\_5411x.h.

6.21.2.14 `__STATIC_INLINE void Chip_GPIO_SetMaskedPortValue ( LPC_GPIO_T * pGPIO, uint8_t port, uint32_t value )`

Set all GPIO pin states, but mask via the MASKP0 register.

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)
<i>value</i>	: Value to set all GPIO pin states (0..n) to

**Returns**

Nothing

Definition at line 363 of file gpio\_5411x.h.

6.21.2.15 `__STATIC_INLINE void Chip_GPIO_SetPinDIR ( LPC_GPIO_T * pGPIO, uint8_t port, uint8_t pin, bool output )`

Set GPIO direction for a single GPIO pin.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to set
<i>pin</i>	: GPIO pin to set direction for
<i>output</i>	: true for output, false for input

## Returns

Nothing

Definition at line 192 of file gpio\_5411x.h.

6.21.2.16 `__STATIC_INLINE void Chip_GPIO_SetPinDIRInput ( LPC_GPIO_T * pGPIO, uint8_t port, uint8_t pin )`

Set GPIO direction for a single GPIO pin to an input.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to set
<i>pin</i>	: GPIO pin to set direction on as input

## Returns

Nothing

Definition at line 179 of file gpio\_5411x.h.

6.21.2.17 `__STATIC_INLINE void Chip_GPIO_SetPinDIROutput ( LPC_GPIO_T * pGPIO, uint8_t port, uint8_t pin )`

Set GPIO direction for a single GPIO pin to an output.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to set
<i>pin</i>	: GPIO pin to set direction on as output

## Returns

Nothing

Definition at line 167 of file gpio\_5411x.h.

6.21.2.18 `__STATIC_INLINE void Chip_GPIO_SetPinOutHigh ( LPC_GPIO_T * pGPIO, uint8_t port, uint8_t pin )`

Set an individual GPIO output pin to the high state.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)
<i>pin</i>	: pin number (0..n) to set high

## Returns

None

**Note**

Any bit set as a '0' will not have it's state changed. This only applies to ports configured as an output.

Definition at line 417 of file gpio\_5411x.h.

6.21.2.19 `__STATIC_INLINE void Chip_GPIO_SetPinOutLow ( LPC_GPIO_T * pGPIO, uint8_t port, uint8_t pin )`

Set an individual GPIO output pin to the low state.

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)
<i>pin</i>	: pin number (0..n) to set low

**Returns**

None

**Note**

Any bit set as a '0' will not have it's state changed. This only applies to ports configured as an output.

Definition at line 460 of file gpio\_5411x.h.

6.21.2.20 `__STATIC_INLINE void Chip_GPIO_SetPinState ( LPC_GPIO_T * pGPIO, uint8_t port, uint8_t pin, bool setting )`

Set a GPIO pin state via the GPIO byte register.

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to set
<i>pin</i>	: GPIO pin to set
<i>setting</i>	: true for high, false for low

**Returns**

Nothing

**Note**

This function replaces [Chip\\_GPIO\\_WritePortBit\(\)](#)

Definition at line 108 of file gpio\_5411x.h.

6.21.2.21 `__STATIC_INLINE void Chip_GPIO_SetPinToggle ( LPC_GPIO_T * pGPIO, uint8_t port, uint8_t pin )`

Toggle an individual GPIO output pin to the opposite state.

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)
<i>pin</i>	: pin number (0..n) to toggle

**Returns**

None

**Note**

Any bit set as a '0' will not have it's state changed. This only applies to ports configured as an output.

Definition at line 488 of file gpio\_5411x.h.

6.21.2.22 `__STATIC_INLINE void Chip_GPIO_SetPortDIR ( LPC_GPIO_T * pGPIO, uint8_t port, uint32_t pinMask, bool outSet )`

Set GPIO direction for a all selected GPIO pins to an input or output.

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number
<i>pinMask</i>	: GPIO pin mask to set direction on (bits 0..b for pins 0..n)
<i>outSet</i>	: Direction value, false = set as inputs, true = set as outputs

**Returns**

Nothing

**Note**

Sets multiple GPIO pins to the input direction, each bit's position that is high sets the corresponding pin number for that bit to an input.

Definition at line 285 of file gpio\_5411x.h.

6.21.2.23 `__STATIC_INLINE void Chip_GPIO_SetPortDIRInput ( LPC_GPIO_T * pGPIO, uint8_t port, uint32_t pinMask )`

Set GPIO direction for a all selected GPIO pins to an input.

**Parameters**

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number
<i>pinMask</i>	: GPIO pin mask to set direction on as input (bits 0..b for pins 0..n)

**Returns**

Nothing

**Note**

Sets multiple GPIO pins to the input direction, each bit's position that is high sets the corresponding pin number for that bit to an input.

Definition at line 270 of file gpio\_5411x.h.

6.21.2.24 `__STATIC_INLINE void Chip_GPIO_SetPortDIROutput ( LPC_GPIO_T * pGPIO, uint8_t port, uint32_t pinMask )`

Set GPIO direction for a all selected GPIO pins to an output.



## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number
<i>pinMask</i>	: GPIO pin mask to set direction on as output (bits 0..b for pins 0..n)

## Returns

Nothing

## Note

Sets multiple GPIO pins to the output direction, each bit's position that is high sets the corresponding pin number for that bit to an output.

Definition at line 256 of file gpio\_5411x.h.

6.21.2.25 `__STATIC_INLINE void Chip_GPIO_SetPortMask ( LPC_GPIO_T * pGPIO, uint8_t port, uint32_t mask )`

Set GPIO port mask value for GPIO masked read and write.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)
<i>mask</i>	: Mask value for read and write

## Returns

Nothing

## Note

Controls which bits corresponding to PIO0\_n are active in the P0MPORT register (bit 0 = PIO0\_0, bit 1 = PIO0\_1, ..., bit 17 = PIO0\_17).

Definition at line 317 of file gpio\_5411x.h.

6.21.2.26 `__STATIC_INLINE void Chip_GPIO_SetPortOutHigh ( LPC_GPIO_T * pGPIO, uint8_t port, uint32_t pins )`

Set selected GPIO output pins to the high state.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)
<i>pins</i>	: pins (0..n) to set high

## Returns

None

## Note

Any bit set as a '0' will not have it's state changed. This only applies to ports configured as an output.

Definition at line 403 of file gpio\_5411x.h.

6.21.2.27 `__STATIC_INLINE void Chip_GPIO_SetPortOutLow ( LPC_GPIO_T * pGPIO, uint8_t port, uint32_t pins )`

Set selected GPIO output pins to the low state.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)
<i>pins</i>	: pins (0..n) to set low

## Returns

None

## Note

Any bit set as a '0' will not have it's state changed. This only applies to ports configured as an output.

Definition at line 446 of file gpio\_5411x.h.

6.21.2.28 `__STATIC_INLINE void Chip_GPIO_SetPortToggle ( LPC_GPIO_T * pGPIO, uint8_t port, uint32_t pins )`

Toggle selected GPIO output pins to the opposite state.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)
<i>pins</i>	: pins (0..n) to toggle

## Returns

None

## Note

Any bit set as a '0' will not have it's state changed. This only applies to ports configured as an output.

Definition at line 474 of file gpio\_5411x.h.

6.21.2.29 `__STATIC_INLINE void Chip_GPIO_SetPortValue ( LPC_GPIO_T * pGPIO, uint8_t port, uint32_t value )`

Set all GPIO raw pin states (regardless of masking)

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: port Number (supports port 0 only)
<i>value</i>	: Value to set all GPIO pin states (0..n) to

## Returns

Nothing

Definition at line 340 of file gpio\_5411x.h.

6.21.2.30 `__STATIC_INLINE void Chip_GPIO_SetValue ( LPC_GPIO_T * pGPIO, uint8_t portNum, uint32_t bitValue )`

Set a GPIO port/bit to the high state.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>portNum</i>	: port number (supports port 0 only)
<i>bitValue</i>	: bit(s) in the port to set high

## Returns

None

## Note

Any bit set as a '0' will not have it's state changed. This only applies to ports configured as an output. It is recommended to use the [Chip\\_GPIO\\_SetPortOutHigh\(\)](#) function instead.

Definition at line 389 of file gpio\_5411x.h.

6.21.2.31 `__STATIC_INLINE void Chip_GPIO_WriteDirBit ( LPC_GPIO_T * pGPIO, uint32_t port, uint8_t pin, bool setting )`

Set GPIO direction for a single GPIO pin.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to set
<i>pin</i>	: GPIO pin to set
<i>setting</i>	: true for output, false for input

## Returns

Nothing

## Note

It is recommended to use the [Chip\\_GPIO\\_SetPinDIROutput\(\)](#), [Chip\\_GPIO\\_SetPinDIRInput\(\)](#) or [Chip\\_GPIO\\_SetPinDIR\(\)](#) functions instead of this function.

Definition at line 150 of file gpio\_5411x.h.

6.21.2.32 `__STATIC_INLINE void Chip_GPIO_WritePortBit ( LPC_GPIO_T * pGPIO, uint32_t port, uint8_t pin, bool setting )`

Set a GPIO port/pin state.

## Parameters

<i>pGPIO</i>	: The base of GPIO peripheral on the chip
<i>port</i>	: GPIO port to set
<i>pin</i>	: GPIO pin to set
<i>setting</i>	: true for high, false for low

## Returns

Nothing

## Note

It is recommended to use the [Chip\\_GPIO\\_SetPinState\(\)](#) function instead.

Definition at line 94 of file gpio\_5411x.h.

## 6.22 CHIP: LPC5411X GPIO group driver

### 6.22.1 Detailed Description

#### Data Structures

- struct [LPC\\_GPIOGROUPINT\\_T](#)  
*GPIO grouped interrupt register block structure.*

#### Macros

- #define [GPIOGR\\_INT](#) (1 << 0)
- #define [GPIOGR\\_COMB](#) (1 << 1)
- #define [GPIOGR\\_TRIG](#) (1 << 2)

#### Functions

- `__STATIC_INLINE void Chip\_GPIOGP\_Init (LPC\_GPIOGROUPINT\_T *pGPIOGPINT)`  
*Initialize GPIO group interrupt block.*
- `__STATIC_INLINE void Chip\_GPIOGP\_DeInit (LPC\_GPIOGROUPINT\_T *pGPIOGPINT)`  
*De-Initialize GPIO group interrupt block.*
- `__STATIC_INLINE void Chip\_GPIOGP\_ClearIntStatus (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group)`  
*Clear interrupt pending status for the selected group.*
- `__STATIC_INLINE bool Chip\_GPIOGP\_GetIntStatus (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group)`  
*Returns current GPIO group inetrrupt pending status.*
- `__STATIC_INLINE void Chip\_GPIOGP\_SelectOrMode (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group)`  
*Selected GPIO group functionality for trigger on any pin in group (OR mode)*
- `__STATIC_INLINE void Chip\_GPIOGP\_SelectAndMode (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group)`  
*Selected GPIO group functionality for trigger on all matching pins in group (AND mode)*
- `__STATIC_INLINE void Chip\_GPIOGP\_SelectEdgeMode (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group)`  
*Selected GPIO group functionality edge trigger mode.*
- `__STATIC_INLINE void Chip\_GPIOGP\_SelectLevelMode (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group)`  
*Selected GPIO group functionality level trigger mode.*
- `__STATIC_INLINE void Chip\_GPIOGP\_SelectLowLevel (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)`  
*Set selected pins for the group and port to low level trigger.*
- `__STATIC_INLINE void Chip\_GPIOGP\_SelectHighLevel (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)`  
*Set selected pins for the group and port to high level trigger.*
- `__STATIC_INLINE void Chip\_GPIOGP\_DisableGroupPins (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)`  
*Disabled selected pins for the group interrupt.*
- `__STATIC_INLINE void Chip\_GPIOGP\_EnableGroupPins (LPC\_GPIOGROUPINT\_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)`  
*Enable selected pins for the group interrupt.*

## 6.22.2 Macro Definition Documentation

### 6.22.2.1 #define GPIOGR\_COMB (1 << 1)

GPIO interrupt OR(0)/AND(1) mode bit

Definition at line 59 of file gpiogroup\_5411x.h.

### 6.22.2.2 #define GPIOGR\_INT (1 << 0)

LPC5411x GPIO group bit definitionsGPIO interrupt pending/clear bit

Definition at line 58 of file gpiogroup\_5411x.h.

### 6.22.2.3 #define GPIOGR\_TRIG (1 << 2)

GPIO interrupt edge(0)/level(1) mode bit

Definition at line 60 of file gpiogroup\_5411x.h.

## 6.22.3 Function Documentation

### 6.22.3.1 \_\_STATIC\_INLINE void Chip\_GPIOGP\_ClearIntStatus ( LPC\_GPIOGROUPINT\_T \* pGPIOGPINT, uint8\_t group )

Clear interrupt pending status for the selected group.

Parameters

<i>pGPIOGPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number

Returns

None

Definition at line 89 of file gpiogroup\_5411x.h.

### 6.22.3.2 \_\_STATIC\_INLINE void Chip\_GPIOGP\_DeInit ( LPC\_GPIOGROUPINT\_T \* pGPIOGPINT )

De-Initialize GPIO group interrupt block.

Parameters

<i>pGPIOGPINT</i>	: The base of GPIO group peripheral on the chip
-------------------	---

Returns

Nothing

Definition at line 78 of file gpiogroup\_5411x.h.

### 6.22.3.3 \_\_STATIC\_INLINE void Chip\_GPIOGP\_DisableGroupPins ( LPC\_GPIOGROUPINT\_T \* pGPIOGPINT, uint8\_t group, uint8\_t port, uint32\_t pinMask )

Disabled selected pins for the group interrupt.

## Parameters

<i>pGPIOGPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number
<i>port</i>	: GPIO port number
<i>pinMask</i>	: Or'ed value of pins to disable interrupt for (bit 0 = pin 0, 1 = pin1, etc.)

## Returns

None

## Note

Disabled pins do not contribute to the group interrupt.

Definition at line 190 of file gpiogroup\_5411x.h.

6.22.3.4 `__STATIC_INLINE void Chip_GPIOGP_EnableGroupPins ( LPC_GPIOGROUPINT_T * pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask )`

Enable selected pins for the group interrupt.

## Parameters

<i>pGPIOGPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number
<i>port</i>	: GPIO port number
<i>pinMask</i>	: Or'ed value of pins to enable interrupt for (bit 0 = pin 0, 1 = pin1, etc.)

## Returns

None

## Note

Enabled pins contribute to the group interrupt.

Definition at line 207 of file gpiogroup\_5411x.h.

6.22.3.5 `__STATIC_INLINE bool Chip_GPIOGP_GetIntStatus ( LPC_GPIOGROUPINT_T * pGPIOGPINT, uint8_t group )`

Returns current GPIO group inetrrupt pending status.

## Parameters

<i>pGPIOGPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number

## Returns

true if the group interrupt is pending, otherwise false.

Definition at line 100 of file gpiogroup\_5411x.h.

6.22.3.6 `__STATIC_INLINE void Chip_GPIOGP_Init ( LPC_GPIOGROUPINT_T * pGPIOGPINT )`

Initialize GPIO group interrupt block.

## Parameters

<i>pGPIOPINT</i>	: The base of GPIO group peripheral on the chip
------------------	---

## Returns

Nothing

Definition at line 67 of file gpiogroup\_5411x.h.

#### 6.22.3.7 `__STATIC_INLINE void Chip_GPIOGP_SelectAndMode ( LPC_GPIOGROUPINT_T * pGPIOPINT, uint8_t group )`

Selected GPIO group functionality for trigger on all matching pins in group (AND mode)

## Parameters

<i>pGPIOPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number

## Returns

None

Definition at line 122 of file gpiogroup\_5411x.h.

#### 6.22.3.8 `__STATIC_INLINE void Chip_GPIOGP_SelectEdgeMode ( LPC_GPIOGROUPINT_T * pGPIOPINT, uint8_t group )`

Selected GPIO group functionality edge trigger mode.

## Parameters

<i>pGPIOPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number

## Returns

None

Definition at line 133 of file gpiogroup\_5411x.h.

#### 6.22.3.9 `__STATIC_INLINE void Chip_GPIOGP_SelectHighLevel ( LPC_GPIOGROUPINT_T * pGPIOPINT, uint8_t group, uint8_t port, uint32_t pinMask )`

Set selected pins for the group and port to high level trigger.

## Parameters

<i>pGPIOPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number
<i>port</i>	: GPIO port number
<i>pinMask</i>	: Or'ed value of pins to select for high level (bit 0 = pin 0, 1 = pin1, etc.)

## Returns

None

Definition at line 173 of file gpiogroup\_5411x.h.

6.22.3.10 `__STATIC_INLINE void Chip_GPIOGP_SelectLevelMode ( LPC_GPIOGROUPINT_T * pGPIOGPINT, uint8_t group )`

Selected GPIO group functionality level trigger mode.



## Parameters

<i>pGPIOPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number

## Returns

None

Definition at line 144 of file gpiogroup\_5411x.h.

6.22.3.11 `__STATIC_INLINE void Chip_GPIOGP_SelectLowLevel ( LPC_GPIOGROUPINT_T * pGPIOPINT, uint8_t group, uint8_t port, uint32_t pinMask )`

Set selected pins for the group and port to low level trigger.

## Parameters

<i>pGPIOPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number
<i>port</i>	: GPIO port number
<i>pinMask</i>	: Or'ed value of pins to select for low level (bit 0 = pin 0, 1 = pin1, etc.)

## Returns

None

Definition at line 157 of file gpiogroup\_5411x.h.

6.22.3.12 `__STATIC_INLINE void Chip_GPIOGP_SelectOrMode ( LPC_GPIOGROUPINT_T * pGPIOPINT, uint8_t group )`

Selected GPIO group functionality for trigger on any pin in group (OR mode)

## Parameters

<i>pGPIOPINT</i>	: Pointer to GPIO group register block
<i>group</i>	: GPIO group number

## Returns

None

Definition at line 111 of file gpiogroup\_5411x.h.

## 6.23 CHIP: LPC5411X I2C master-only driver

### 6.23.1 Detailed Description

This driver only works in master mode. To describe the I2C transactions following symbols are used in driver documentation.

#### Key to symbols

S (1 bit) : Start bit P (1 bit) : Stop bit Rd/Wr (1 bit) : Read/Write bit. Rd equals 1, Wr equals 0. A, NA (1 bit) : Acknowledge and Not-Acknowledge bit. Addr (7 bits): I2C 7 bit address. Note that this can be expanded as usual to get a 10 bit I2C address. Data (8 bits): A plain data byte. Sometimes, I write DataLow, DataHigh for 16 bit data. [..]: Data sent by I2C device, as opposed to data sent by the host adapter.

#### Data Structures

- struct [I2CM\\_XFER\\_T](#)  
*Master transfer data structure definitions.*

#### Macros

- #define [I2CM\\_STATUS\\_OK](#) 0x00
- #define [I2CM\\_STATUS\\_ERROR](#) 0x01
- #define [I2CM\\_STATUS\\_NAK\\_ADR](#) 0x02
- #define [I2CM\\_STATUS\\_BUS\\_ERROR](#) 0x03
- #define [I2CM\\_STATUS\\_NAK\\_DAT](#) 0x04
- #define [I2CM\\_STATUS\\_ARBLOST](#) 0x05
- #define [I2CM\\_STATUS\\_BUSY](#) 0xFF

#### Functions

- static [INLINE](#) void [Chip\\_I2CM\\_SetDutyCycle](#) ([LPC\\_I2C\\_T](#) \*pI2C, uint16\_t sclH, uint16\_t sclL)  
*Sets HIGH and LOW duty cycle registers.*
- void [Chip\\_I2CM\\_SetBusSpeed](#) ([LPC\\_I2C\\_T](#) \*pI2C, uint32\_t busSpeed)  
*Set up bus speed for LPC\_I2C controller.*
- static [INLINE](#) void [Chip\\_I2CM\\_Enable](#) ([LPC\\_I2C\\_T](#) \*pI2C)  
*Enable I2C Master interface.*
- static [INLINE](#) void [Chip\\_I2CM\\_Disable](#) ([LPC\\_I2C\\_T](#) \*pI2C)  
*Disable I2C Master interface.*
- static [INLINE](#) uint32\_t [Chip\\_I2CM\\_GetStatus](#) ([LPC\\_I2C\\_T](#) \*pI2C)  
*Get I2C Status.*
- static [INLINE](#) void [Chip\\_I2CM\\_ClearStatus](#) ([LPC\\_I2C\\_T](#) \*pI2C, uint32\_t clrStatus)  
*Clear I2C status bits (master)*
- static [INLINE](#) bool [Chip\\_I2CM\\_IsMasterPending](#) ([LPC\\_I2C\\_T](#) \*pI2C)  
*Check if I2C Master is pending.*
- static [INLINE](#) uint32\_t [Chip\\_I2CM\\_GetMasterState](#) ([LPC\\_I2C\\_T](#) \*pI2C)  
*Get current state of the I2C Master.*
- static [INLINE](#) void [Chip\\_I2CM\\_SendStart](#) ([LPC\\_I2C\\_T](#) \*pI2C)  
*Transmit START or Repeat-START signal on I2C bus.*
- static [INLINE](#) void [Chip\\_I2CM\\_SendStop](#) ([LPC\\_I2C\\_T](#) \*pI2C)  
*Transmit STOP signal on I2C bus.*

- static **INLINE** void `Chip_I2CM_MasterContinue` (`LPC_I2C_T *pI2C`)  
*Master Continue transfer operation.*
- static **INLINE** void `Chip_I2CM_WriteByte` (`LPC_I2C_T *pI2C`, `uint8_t data`)  
*Transmit a single data byte through the I2C peripheral (master)*
- static **INLINE** `uint8_t` `Chip_I2CM_ReadByte` (`LPC_I2C_T *pI2C`)  
*Read a single byte data from the I2C peripheral (master)*
- `uint32_t` `Chip_I2CM_XferHandler` (`LPC_I2C_T *pI2C`, `I2CM_XFER_T *xfer`)  
*Transfer state change handler.*
- void `Chip_I2CM_Xfer` (`LPC_I2C_T *pI2C`, `I2CM_XFER_T *xfer`)  
*Transmit and Receive data in master mode.*
- `uint32_t` `Chip_I2CM_XferBlocking` (`LPC_I2C_T *pI2C`, `I2CM_XFER_T *xfer`)  
*Transmit and Receive data in master mode.*

## 6.23.2 Macro Definition Documentation

### 6.23.2.1 `#define I2CM_STATUS_ARBLOST 0x05`

Arbitration lost.

Definition at line 68 of file `i2cm_5411x.h`.

### 6.23.2.2 `#define I2CM_STATUS_BUS_ERROR 0x03`

I2C bus error

Definition at line 66 of file `i2cm_5411x.h`.

### 6.23.2.3 `#define I2CM_STATUS_BUSY 0xFF`

I2C transmistter is busy.

Definition at line 69 of file `i2cm_5411x.h`.

### 6.23.2.4 `#define I2CM_STATUS_ERROR 0x01`

Unknown error condition.

Definition at line 64 of file `i2cm_5411x.h`.

### 6.23.2.5 `#define I2CM_STATUS_NAK_ADR 0x02`

No acknowledgement received from slave during address phase.

Definition at line 65 of file `i2cm_5411x.h`.

### 6.23.2.6 `#define I2CM_STATUS_NAK_DAT 0x04`

No acknowledgement received from slave during data phase.

Definition at line 67 of file `i2cm_5411x.h`.

### 6.23.2.7 `#define I2CM_STATUS_OK 0x00`

`I2CM_5411X_STATUS_TYPES` I2C master transfer status typesRequested Request was executed successfully.

Definition at line 63 of file `i2cm_5411x.h`.

### 6.23.3 Function Documentation

6.23.3.1 **static INLINE void** Chip\_I2CM\_ClearStatus ( LPC\_I2C\_T \* *pI2C*, uint32\_t *clrStatus* ) [static]

Clear I2C status bits (master)

#### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
<i>clrStatus</i>	: Status bit to clear, ORed Value of I2C_STAT_MSTRARBLOSS and I2C_STAT_MSTST↵ STPERR

#### Returns

Nothing

#### Note

This function clears selected status flags.

Definition at line 158 of file i2cm\_5411x.h.

6.23.3.2 **static INLINE void** Chip\_I2CM\_Disable ( LPC\_I2C\_T \* *pI2C* ) [static]

Disable I2C Master interface.

#### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

#### Returns

Nothing

#### Note

Definition at line 135 of file i2cm\_5411x.h.

6.23.3.3 **static INLINE void** Chip\_I2CM\_Enable ( LPC\_I2C\_T \* *pI2C* ) [static]

Enable I2C Master interface.

#### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

#### Returns

Nothing

#### Note

Definition at line 124 of file i2cm\_5411x.h.

6.23.3.4 **static INLINE uint32\_t** Chip\_I2CM\_GetMasterState ( LPC\_I2C\_T \* *pI2C* ) [static]

Get current state of the I2C Master.

## Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

## Returns

Master State Code, a value in the range of 0 - 4

## Note

After the Master is pending this state code tells the reason for Master pending.

Definition at line 182 of file i2cm\_5411x.h.

**6.23.3.5** `static INLINE uint32_t Chip_I2CM_GetStatus ( LPC_I2C_T * pI2C ) [static]`

Get I2C Status.

## Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

## Returns

I2C Status register value

## Note

This function returns the value of the status register.

Definition at line 146 of file i2cm\_5411x.h.

**6.23.3.6** `static INLINE bool Chip_I2CM_IsMasterPending ( LPC_I2C_T * pI2C ) [static]`

Check if I2C Master is pending.

## Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

## Returns

Returns TRUE if the Master is pending else returns FALSE

## Note

Definition at line 170 of file i2cm\_5411x.h.

**6.23.3.7** `static INLINE void Chip_I2CM_MasterContinue ( LPC_I2C_T * pI2C ) [static]`

Master Continue transfer operation.

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

Nothing

**Note**

This function sets the master controller to continue transmission. This should be called only when master is pending. The function writes a complete value to Master Control register, ORing is not advised.

Definition at line 221 of file i2cm\_5411x.h.

**6.23.3.8** `static INLINE uint8_t Chip_I2CM_ReadByte ( LPC_I2C_T * pI2C ) [static]`

Read a single byte data from the I2C peripheral (master)

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

A single byte of data read

**Note**

This function reads a byte from the I2C receive hold register regardless of I2C state.

Definition at line 247 of file i2cm\_5411x.h.

**6.23.3.9** `static INLINE void Chip_I2CM_SendStart ( LPC_I2C_T * pI2C ) [static]`

Transmit START or Repeat-START signal on I2C bus.

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

Nothing

**Note**

This function sets the controller to transmit START condition when the bus becomes free. This should be called only when master is pending. The function writes a complete value to Master Control register, ORing is not advised.

Definition at line 195 of file i2cm\_5411x.h.

**6.23.3.10** `static INLINE void Chip_I2CM_SendStop ( LPC_I2C_T * pI2C ) [static]`

Transmit STOP signal on I2C bus.

## Parameters

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

## Returns

Nothing

## Note

This function sets the controller to transmit STOP condition. This should be called only when master is pending. The function writes a complete value to Master Control register, ORing is not advised.

Definition at line 208 of file i2cm\_5411x.h.

6.23.3.11 void Chip\_I2CM\_SetBusSpeed ( LPC\_I2C\_T \* *pl2C*, uint32\_t *busSpeed* )

Set up bus speed for LPC\_I2C controller.

## Parameters

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>busSpeed</i>	: I2C bus clock rate

## Returns

Nothing

## Note

Per I2C specification the busSpeed should be

- 100000 for Standard mode
- 400000 for Fast mode
- 1000000 for Fast mode plus IOCON registers corresponding to I2C pads should be updated according to the bus mode.

This API will also support non-standard bus speeds provided it is an integer multiple of (18, 16, 14, 12, 10, 8, 6, 4).

This API will automatically calculate and set divider value.

Definition at line 51 of file i2cm\_5411x.c.

6.23.3.12 static INLINE void Chip\_I2CM\_SetDutyCycle ( LPC\_I2C\_T \* *pl2C*, uint16\_t *sclH*, uint16\_t *sclL* ) [static]

Sets HIGH and LOW duty cycle registers.

## Parameters

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>sclH</i>	: Number of I2C_PCLK cycles for the SCL HIGH time value between (2 - 9).
<i>sclL</i>	: Number of I2C_PCLK cycles for the SCL LOW time value between (2 - 9).

## Returns

Nothing

**Note**

The I2C clock divider should be set to the appropriate value before calling this function. The I2C baud is determined by the following formula:

$$\text{I2C\_bitFrequency} = (\text{I2C\_PCLK}) / (\text{I2C\_CLKDIV} * (\text{sclH} + \text{sclL}))$$

where I2C\_PCLK is the frequency of the System clock and I2C\_CLKDIV is I2C clock divider

Definition at line 96 of file i2cm\_5411x.h.

**6.23.3.13** `static INLINE void Chip_I2CM_WriteByte ( LPC_I2C_T * pl2C, uint8_t data ) [static]`

Transmit a single data byte through the I2C peripheral (master)

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>data</i>	: Byte to transmit

**Returns**

Nothing

**Note**

This function attempts to place a byte into the I2C Master Data Register

Definition at line 235 of file i2cm\_5411x.h.

**6.23.3.14** `void Chip_I2CM_Xfer ( LPC_I2C_T * pl2C, I2CM_XFER_T * xfer )`

Transmit and Receive data in master mode.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>xfer</i>	: Pointer to a <a href="#">I2CM_XFER_T</a> structure see notes below

**Returns**

Nothing

**Note**

The parameter *xfer* should have its member *slaveAddr* initialized to the 7-Bit slave address to which the master will do the xfer, Bit0 to bit6 should have the address and Bit8 is ignored. During the transfer no code (like event handler) must change the content of the memory pointed to by *xfer*. The member of *xfer*, *txBuff* and *txSz* be initialized to the memory from which the I2C must pick the data to be transferred to slave and the number of bytes to send respectively, similarly *rxBuff* and *rxSz* must have pointer to memory where data received from slave be stored and the number of data to get from slave respectively. Following types of transfers are possible:

- Write-only transfer: When *rxSz* member of *xfer* is set to 0.

S Addr Wr [A] txBuff0 [A] txBuff1 [A] ... txBuffN [A] P

- If I2CM\_XFER\_OPTION\_IGNORE\_NACK is set in *options* member

S Addr Wr [A] txBuff0 [A or NA] ... txBuffN [A or NA] P

- Read-only transfer: When *txSz* member of *xfer* is set to 0.

S Addr Rd [A] [rxBuff0] A [rxBuff1] A ... [rxBuffN] NA P



- If I2CM\_XFER\_OPTION\_LAST\_RX\_ACK is set in *options* member  
S Addr Rd [A] [rxBuff0] A [rxBuff1] A ... [rxBuffN] A P
- Read-Write transfer: When *rxSz* and *@ txSz* members of *xfer* are non-zero.

S Addr Wr [A] txBuff0 [A] txBuff1 [A] ... txBuffN [A]  
S Addr Rd [A] [rxBuff0] A [rxBuff1] A ... [rxBuffN] NA P

Definition at line 163 of file i2cm\_5411x.c.

#### 6.23.3.15 uint32\_t Chip\_I2CM\_XferBlocking ( LPC\_I2C\_T \* *pI2C*, I2CM\_XFER\_T \* *xfer* )

Transmit and Receive data in master mode.

##### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
<i>xfer</i>	: Pointer to a <a href="#">I2CM_XFER_T</a> structure see notes below

##### Returns

Returns non-zero value on succesful completion of transfer.

##### Note

This function operates same as [Chip\\_I2CM\\_Xfer\(\)](#), but is a blocking call.

Definition at line 176 of file i2cm\_5411x.c.

#### 6.23.3.16 uint32\_t Chip\_I2CM\_XferHandler ( LPC\_I2C\_T \* *pI2C*, I2CM\_XFER\_T \* *xfer* )

Transfer state change handler.

##### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
<i>xfer</i>	: Pointer to a <a href="#">I2CM_XFER_T</a> structure see notes below

##### Returns

Returns non-zero value on completion of transfer. The *status* member of *xfer* structure contains the current status of the transfer at the end of the call.

##### Note

The parameter *xfer* should be same as the one passed to [Chip\\_I2CM\\_Xfer\(\)](#) routine. This function should be called from the I2C interrupt handler only when a master interrupt occurs.

Definition at line 69 of file i2cm\_5411x.c.

## 6.24 CHIP: LPC5411X I2C slave-only driver

### 6.24.1 Detailed Description

This driver only works in slave mode.

#### Data Structures

- struct [I2CS\\_XFER\\_T](#)

#### Typedefs

- typedef void(\* [I2CSlaveXferStart](#)) (uint8\_t addr)  
*I2C slave service start callback This callback is called from the I2C slave handler when an I2C slave address is received and needs servicing. It's used to indicate the start of a slave transfer that will happen on the slave bus.*
- typedef uint8\_t(\* [I2CSlaveXferSend](#)) (uint8\_t \*data)  
*I2C slave send data callback This callback is called from the I2C slave handler when an I2C slave address needs data to send.  
If you want to NAK the master, return I2C\_SLVCTL\_SLVNACK to the caller. Return I2C\_SLVCTL\_SLVCONTINUE or 0 to the caller for normal non-DMA data transfer. If you've setup a DMA descriptor for the transfer, return I2C\_SLVCTL\_SLVDMA to the caller.*
- typedef uint8\_t(\* [I2CSlaveXferRecv](#)) (uint8\_t data)  
*I2C slave receive data callback This callback is called from the I2C slave handler when an I2C slave address has receive data.  
If you want to NAK the master, return I2C\_SLVCTL\_SLVNACK to the caller. Return I2C\_SLVCTL\_SLVCONTINUE or 0 to the caller for normal non-DMA data transfer. If you've setup a DMA descriptor for the transfer, return I2C\_SLVCTL\_SLVDMA to the caller.*
- typedef void(\* [I2CSlaveXferDone](#)) (void)  
*I2C slave service done callback This callback is called from the I2C slave handler when an I2C slave transfer is completed. It's used to indicate the end of a slave transfer.*

#### Functions

- \_\_STATIC\_INLINE void [Chip\\_I2CS\\_Enable](#) (LPC\_I2C\_T \*pI2C)  
*Enable I2C slave interface.*
- \_\_STATIC\_INLINE void [Chip\\_I2CS\\_Disable](#) (LPC\_I2C\_T \*pI2C)  
*Disable I2C slave interface.*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_I2CS\\_GetStatus](#) (LPC\_I2C\_T \*pI2C)  
*Get I2C Status.*
- \_\_STATIC\_INLINE void [Chip\\_I2CS\\_ClearStatus](#) (LPC\_I2C\_T \*pI2C, uint32\_t clrStatus)  
*Clear I2C status bits (slave)*
- \_\_STATIC\_INLINE bool [Chip\\_I2CS\\_IsSlavePending](#) (LPC\_I2C\_T \*pI2C)  
*Check if I2C slave is pending.*
- \_\_STATIC\_INLINE bool [Chip\\_I2CS\\_IsSlaveSelected](#) (LPC\_I2C\_T \*pI2C)  
*Check if I2C slave is selected.*
- \_\_STATIC\_INLINE bool [Chip\\_I2CS\\_IsSlaveDeSelected](#) (LPC\_I2C\_T \*pI2C)  
*Check if I2C slave is deselected.*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_I2CS\\_GetSlaveState](#) (LPC\_I2C\_T \*pI2C)  
*Get current state of the I2C slave.*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_I2CS\\_GetSlaveMatchIndex](#) (LPC\_I2C\_T \*pI2C)

- Returns the current slave address match index.*
- `__STATIC_INLINE void Chip_I2CS_SlaveContinue (LPC_I2C_T *pI2C)`  
*Slave Continue transfer operation (ACK)*
- `__STATIC_INLINE void Chip_I2CS_SlaveNACK (LPC_I2C_T *pI2C)`  
*Slave NACK operation.*
- `__STATIC_INLINE void Chip_I2CS_SlaveEnableDMA (LPC_I2C_T *pI2C)`  
*Enable slave DMA operation.*
- `__STATIC_INLINE void Chip_I2CS_SlaveDisableDMA (LPC_I2C_T *pI2C)`  
*Disable slave DMA operation.*
- `__STATIC_INLINE void Chip_I2CS_WriteByte (LPC_I2C_T *pI2C, uint8_t data)`  
*Transmit a single data byte through the I2C peripheral (slave)*
- `__STATIC_INLINE uint8_t Chip_I2CS_ReadByte (LPC_I2C_T *pI2C)`  
*Read a single byte data from the I2C peripheral (slave)*
- `__STATIC_INLINE void Chip_I2CS_SetSlaveAddr (LPC_I2C_T *pI2C, uint8_t slvNum, uint8_t slvAddr)`  
*Set a I2C slave address for slave operation.*
- `__STATIC_INLINE uint8_t Chip_I2CS_GetSlaveAddr (LPC_I2C_T *pI2C, uint8_t slvNum)`  
*Return a I2C programmed slave address.*
- `__STATIC_INLINE void Chip_I2CS_EnableSlaveAddr (LPC_I2C_T *pI2C, uint8_t slvNum)`  
*Enable a I2C address.*
- `__STATIC_INLINE void Chip_I2CS_DisableSlaveAddr (LPC_I2C_T *pI2C, uint8_t slvNum)`  
*Disable a I2C address.*
- `__STATIC_INLINE void Chip_I2CS_SetSlaveQual0 (LPC_I2C_T *pI2C, bool extend, uint8_t slvAddr)`  
*Setup slave qualifier address.*
- `uint32_t Chip_I2CS_XferHandler (LPC_I2C_T *pI2C, const I2CS_XFER_T *xfers)`  
*Slave transfer state change handler.*

## 6.24.2 Typedef Documentation

### 6.24.2.1 typedef void(\* I2CSlaveXferDone) (void)

I2C slave service done callback This callback is called from the I2C slave handler when an I2C slave transfer is completed. It's used to indicate the end of a slave transfer.

Definition at line 76 of file `i2cs_5411x.h`.

### 6.24.2.2 typedef uint8\_t(\* I2CSlaveXferRecv) (uint8\_t data)

I2C slave receive data callback This callback is called from the I2C slave handler when an I2C slave address has receive data.

If you want to NAK the master, return `I2C_SLVCTL_SLVNACK` to the caller. Return `I2C_SLVCTL_SLVCONTINUE` or 0 to the caller for normal non-DMA data transfer. If you've setup a DMA descriptor for the transfer, return `I2C_SLVCTL_SLVDMA` to the caller.

Definition at line 70 of file `i2cs_5411x.h`.

### 6.24.2.3 typedef uint8\_t(\* I2CSlaveXferSend) (uint8\_t \*data)

I2C slave send data callback This callback is called from the I2C slave handler when an I2C slave address needs data to send.

If you want to NAK the master, return `I2C_SLVCTL_SLVNACK` to the caller. Return `I2C_SLVCTL_SLVCONTINUE` or 0 to the caller for normal non-DMA data transfer. If you've setup a DMA descriptor for the transfer, return

I2C\_SLVCTL\_SLVDMA to the caller.

Definition at line 61 of file i2cs\_5411x.h.

#### 6.24.2.4 typedef void(\* I2CSlaveXferStart) (uint8\_t addr)

I2C slave service start callback This callback is called from the I2C slave handler when an I2C slave address is received and needs servicing. It's used to indicate the start of a slave transfer that will happen on the slave bus.

Definition at line 52 of file i2cs\_5411x.h.

### 6.24.3 Function Documentation

#### 6.24.3.1 \_\_STATIC\_INLINE void Chip\_I2CS\_ClearStatus ( LPC\_I2C\_T \* pI2C, uint32\_t clrStatus )

Clear I2C status bits (slave)

##### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
<i>clrStatus</i>	: Status bit to clear, must be I2C_STAT_SLVDESEL

##### Returns

Nothing

##### Note

This function clears selected status flags.

Definition at line 149 of file i2cs\_5411x.h.

#### 6.24.3.2 \_\_STATIC\_INLINE void Chip\_I2CS\_Disable ( LPC\_I2C\_T \* pI2C )

Disable I2C slave interface.

##### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

##### Returns

Nothing

Definition at line 126 of file i2cs\_5411x.h.

#### 6.24.3.3 \_\_STATIC\_INLINE void Chip\_I2CS\_DisableSlaveAddr ( LPC\_I2C\_T \* pI2C, uint8\_t slvNum )

Disable a I2C address.

##### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

<i>slvNum</i>	: Possible slave address number, between 0 - 3
---------------	--

**Returns**

Nothing

Definition at line 326 of file i2cs\_5411x.h.

**6.24.3.4** `__STATIC_INLINE void Chip_I2CS_Enable ( LPC_I2C_T * pI2C )`

Enable I2C slave interface.

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

Nothing

**Note**

Do not call this function until the slave interface is fully configured.

Definition at line 116 of file i2cs\_5411x.h.

**6.24.3.5** `__STATIC_INLINE void Chip_I2CS_EnableSlaveAddr ( LPC_I2C_T * pI2C, uint8_t slvNum )`

Enable a I2C address.

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
<i>slvNum</i>	: Possible slave address number, between 0 - 3

**Returns**

Nothing

Definition at line 315 of file i2cs\_5411x.h.

**6.24.3.6** `__STATIC_INLINE uint8_t Chip_I2CS_GetSlaveAddr ( LPC_I2C_T * pI2C, uint8_t slvNum )`

Return a I2C programmed slave address.

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
<i>slvNum</i>	: Possible slave address number, between 0 - 3

**Returns**

Nothing

Definition at line 304 of file i2cs\_5411x.h.

**6.24.3.7** `__STATIC_INLINE uint32_t Chip_I2CS_GetSlaveMatchIndex ( LPC_I2C_T * pI2C )`

Returns the current slave address match index.

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

slave match index, 0 - 3

Definition at line 204 of file i2cs\_5411x.h.

**6.24.3.8   \_\_STATIC\_INLINE uint32\_t Chip\_I2CS\_GetSlaveState ( LPC\_I2C\_T \* *pI2C* )**

Get current state of the I2C slave.

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

slave State Code, a value of type I2C\_STAT\_SLVCODE\_\*

**Note**

After the slave is pending this state code tells the reason for slave pending.

Definition at line 194 of file i2cs\_5411x.h.

**6.24.3.9   \_\_STATIC\_INLINE uint32\_t Chip\_I2CS\_GetStatus ( LPC\_I2C\_T \* *pI2C* )**

Get I2C Status.

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

I2C Status register value

**Note**

This function returns the value of the status register.

Definition at line 137 of file i2cs\_5411x.h.

**6.24.3.10   \_\_STATIC\_INLINE bool Chip\_I2CS\_IsSlaveDeSelected ( LPC\_I2C\_T \* *pI2C* )**

Check if I2C slave is deselected.

**Parameters**

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

Returns TRUE if the slave is is deselected, otherwise FALSE

**Note**

Definition at line 182 of file i2cs\_5411x.h.

#### 6.24.3.11 `__STATIC_INLINE bool Chip_I2CS_IsSlavePending ( LPC_I2C_T * pI2C )`

Check if I2C slave is pending.

##### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

##### Returns

Returns TRUE if the slave is pending else returns FALSE

##### Note

Definition at line 160 of file i2cs\_5411x.h.

#### 6.24.3.12 `__STATIC_INLINE bool Chip_I2CS_IsSlaveSelected ( LPC_I2C_T * pI2C )`

Check if I2C slave is selected.

##### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

##### Returns

Returns TRUE if the slave is selected, otherwise FALSE

##### Note

Definition at line 171 of file i2cs\_5411x.h.

#### 6.24.3.13 `__STATIC_INLINE uint8_t Chip_I2CS_ReadByte ( LPC_I2C_T * pI2C )`

Read a single byte data from the I2C peripheral (slave)

##### Parameters

<i>pI2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

##### Returns

A single byte of data read

##### Note

This function reads a byte from the I2C receive hold register regardless of I2C state.

Definition at line 279 of file i2cs\_5411x.h.

#### 6.24.3.14 `__STATIC_INLINE void Chip_I2CS_SetSlaveAddr ( LPC_I2C_T * pI2C, uint8_t slvNum, uint8_t slvAddr )`

Set a I2C slave address for slave operation.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>slvNum</i>	: Possible slave address number, between 0 - 3
<i>slvAddr</i>	: Slave Address for the index (7-bits, bit 7 = 0)

**Returns**

Nothing

**Note**

Setting a slave address also enables the slave address. Do not 'pre-shift' the slave address.

Definition at line 293 of file i2cs\_5411x.h.

6.24.3.15 `__STATIC_INLINE void Chip_I2CS_SetSlaveQual0 ( LPC_I2C_T * pl2C, bool extend, uint8_t slvAddr )`

Setup slave qialifier address.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>extend</i>	: true to extend I2C slave detect address 0 range, or false to match to corresponding bits
<i>slvAddr</i>	: Slave address qualifier, see SLVQUAL0 register in User Manual

**Returns**

Nothing

**Note**

Do not 'pre-shift' the slave address.

Definition at line 339 of file i2cs\_5411x.h.

6.24.3.16 `__STATIC_INLINE void Chip_I2CS_SlaveContinue ( LPC_I2C_T * pl2C )`

Slave Continue transfer operation (ACK)

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

Nothing

**Note**

This function sets the slave controller to continue transmission. This should be called only when slave is pending. The function writes a complete value to slave Control register, ORing is not advised.

Definition at line 217 of file i2cs\_5411x.h.

6.24.3.17 `__STATIC_INLINE void Chip_I2CS_SlaveDisableDMA ( LPC_I2C_T * pl2C )`

Disable slave DMA operation.



## Parameters

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

## Returns

Nothing

## Note

This function disables DMA mode for the slave controller.

Definition at line 253 of file i2cs\_5411x.h.

#### 6.24.3.18 `__STATIC_INLINE void Chip_I2CS_SlaveEnableDMA ( LPC_I2C_T * pl2C )`

Enable slave DMA operation.

## Parameters

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

## Returns

Nothing

## Note

This function enables DMA mode for the slave controller. In DMA mode, the 'continue' and 'NACK' operations aren't used and the I2C slave controller will automatically NACK any bytes beyond the available DMA buffer size.

Definition at line 242 of file i2cs\_5411x.h.

#### 6.24.3.19 `__STATIC_INLINE void Chip_I2CS_SlaveNACK ( LPC_I2C_T * pl2C )`

Slave NACK operation.

## Parameters

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

## Returns

Nothing

## Note

This function sets the slave controller to NAK the master.

Definition at line 228 of file i2cs\_5411x.h.

#### 6.24.3.20 `__STATIC_INLINE void Chip_I2CS_WriteByte ( LPC_I2C_T * pl2C, uint8_t data )`

Transmit a single data byte through the I2C peripheral (slave)

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>data</i>	: Byte to transmit

**Returns**

Nothing

**Note**

This function attempts to place a byte into the I2C slave Data Register

Definition at line 267 of file i2cs\_5411x.h.

6.24.3.21 uint32\_t Chip\_I2CS\_XferHandler ( LPC\_I2C\_T \* *pl2C*, const I2CS\_XFER\_T \* *xfers* )

Slave transfer state change handler.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>xfers</i>	: Pointer to a I2CS_MULTI_XFER_T structure see notes below

**Returns**

Returns non-zero value on completion of transfer or NAK

**Note**

See [I2CS\\_XFER\\_T](#) for more information on this function. When using this function, the I2C\_INTENSET\_SLVPENDING and I2C\_INTENSET\_SLVDESEL interrupts should be enabled and setup in the I2C interrupt handler to call this function when they fire.

Definition at line 51 of file i2cs\_5411x.c.

## 6.25 CHIP: LPC5411X IOCON register block and driver

### 6.25.1 Detailed Description

#### Data Structures

- struct [LPC\\_IOCON\\_T](#)  
*LPC5411X IO Configuration Unit register block structure.*
- struct [PINMUX\\_GRP\\_T](#)  
*Array of IOCON pin definitions passed to [Chip\\_IOCON\\_SetPinMuxing\(\)](#) must be in this format.*

#### Macros

- `#define IOCON_FUNC0 0x0`
- `#define IOCON_FUNC1 0x1`
- `#define IOCON_FUNC2 0x2`
- `#define IOCON_FUNC3 0x3`
- `#define IOCON_FUNC4 0x4`
- `#define IOCON_FUNC5 0x5`
- `#define IOCON_FUNC6 0x6`
- `#define IOCON_FUNC7 0x7`
- `#define IOCON_MODE_INACT (0x0 << 3)`
- `#define IOCON_MODE_PULLDOWN (0x1 << 3)`
- `#define IOCON_MODE_PULLUP (0x2 << 3)`
- `#define IOCON_MODE_REPEATER (0x3 << 3)`
- `#define IOCON_HYS_EN (0x1 << 5)`
- `#define IOCON_GPIO_MODE (0x1 << 5)`
- `#define IOCON_I2C_SLEW (0x1 << 5)`
- `#define IOCON_INV_EN (0x1 << 6)`
- `#define IOCON_ANALOG_EN (0x0 << 7)`
- `#define IOCON_DIGITAL_EN (0x1 << 7)`
- `#define IOCON_STDI2C_EN (0x1 << 8)`
- `#define IOCON_FASTI2C_EN (0x3 << 8)`
- `#define IOCON_INPFILT_OFF (0x1 << 8)`
- `#define IOCON_INPFILT_ON (0x0 << 8)`
- `#define IOCON_OPENDRAIN_EN (0x1 << 10)`
- `#define IOCON_S_MODE_0CLK (0x0 << 11)`
- `#define IOCON_S_MODE_1CLK (0x1 << 11)`
- `#define IOCON_S_MODE_2CLK (0x2 << 11)`
- `#define IOCON_S_MODE_3CLK (0x3 << 11)`
- `#define IOCON_S_MODE(clks) ((clks) << 11)`
- `#define IOCON_CLKDIV(div) ((div) << 13)`

#### Functions

- `__STATIC_INLINE void Chip_IOCON_PinMuxSet (LPC_IOCON_T *pIOCON, uint8_t port, uint8_t pin, uint32_t modefunc)`  
*Sets I/O Control pin mux.*
- `__STATIC_INLINE void Chip_IOCON_PinMux (LPC_IOCON_T *pIOCON, uint8_t port, uint8_t pin, uint16_t mode, uint8_t func)`  
*I/O Control pin mux.*
- `__STATIC_INLINE void Chip_IOCON_SetPinMuxing (LPC_IOCON_T *pIOCON, const PINMUX_GRP_T *pinArray, uint32_t arrayLength)`  
*Set all I/O Control pin muxing.*

## 6.25.2 Macro Definition Documentation

### 6.25.2.1 `#define IOCON_ANALOG_EN (0x0 << 7)`

Enables analog function by setting 0 to bit 7

Definition at line 81 of file `iocon_5411x.h`.

### 6.25.2.2 `#define IOCON_CLKDIV( div ) ((div) << 13)`

Select peripheral clock divider for input filter sampling clock,  $2^n$ ,  $n=0-6$

Definition at line 93 of file `iocon_5411x.h`.

### 6.25.2.3 `#define IOCON_DIGITAL_EN (0x1 << 7)`

Enables digital function by setting 1 to bit 7(default)

Definition at line 82 of file `iocon_5411x.h`.

### 6.25.2.4 `#define IOCON_FASTI2C_EN (0x3 << 8)`

I2C Fast-mode Plus and high-speed slave

Definition at line 84 of file `iocon_5411x.h`.

### 6.25.2.5 `#define IOCON_FUNC0 0x0`

IOCON function and mode selection definitions See the User Manual for specific modes and functions supported by the various LPC15XX pins. Selects pin function 0

Definition at line 65 of file `iocon_5411x.h`.

### 6.25.2.6 `#define IOCON_FUNC1 0x1`

Selects pin function 1

Definition at line 66 of file `iocon_5411x.h`.

### 6.25.2.7 `#define IOCON_FUNC2 0x2`

Selects pin function 2

Definition at line 67 of file `iocon_5411x.h`.

### 6.25.2.8 `#define IOCON_FUNC3 0x3`

Selects pin function 3

Definition at line 68 of file `iocon_5411x.h`.

### 6.25.2.9 `#define IOCON_FUNC4 0x4`

Selects pin function 4

Definition at line 69 of file `iocon_5411x.h`.

**6.25.2.10 #define IOCON\_FUNC5 0x5**

Selects pin function 5

Definition at line 70 of file iocon\_5411x.h.

**6.25.2.11 #define IOCON\_FUNC6 0x6**

Selects pin function 6

Definition at line 71 of file iocon\_5411x.h.

**6.25.2.12 #define IOCON\_FUNC7 0x7**

Selects pin function 7

Definition at line 72 of file iocon\_5411x.h.

**6.25.2.13 #define IOCON\_GPIO\_MODE (0x1 << 5)**

GPIO Mode

Definition at line 78 of file iocon\_5411x.h.

**6.25.2.14 #define IOCON\_HYS\_EN (0x1 << 5)**

Enables hysteresis

Definition at line 77 of file iocon\_5411x.h.

**6.25.2.15 #define IOCON\_I2C\_SLEW (0x1 << 5)**

I2C Slew Rate Control

Definition at line 79 of file iocon\_5411x.h.

**6.25.2.16 #define IOCON\_INPFILT\_OFF (0x1 << 8)**

Input filter Off for GPIO pins

Definition at line 85 of file iocon\_5411x.h.

**6.25.2.17 #define IOCON\_INPFILT\_ON (0x0 << 8)**

Input filter On for GPIO pins

Definition at line 86 of file iocon\_5411x.h.

**6.25.2.18 #define IOCON\_INV\_EN (0x1 << 6)**

Enables invert function on input

Definition at line 80 of file iocon\_5411x.h.

**6.25.2.19 #define IOCON\_MODE\_INACT (0x0 << 3)**

No addition pin function

Definition at line 73 of file iocon\_5411x.h.

**6.25.2.20 #define IOCON\_MODE\_PULLDOWN (0x1 << 3)**

Selects pull-down function

Definition at line 74 of file iocon\_5411x.h.

**6.25.2.21 #define IOCON\_MODE\_PULLUP (0x2 << 3)**

Selects pull-up function

Definition at line 75 of file iocon\_5411x.h.

**6.25.2.22 #define IOCON\_MODE\_REPEATER (0x3 << 3)**

Selects pin repeater function

Definition at line 76 of file iocon\_5411x.h.

**6.25.2.23 #define IOCON\_OPENDRAIN\_EN (0x1 << 10)**

Enables open-drain function

Definition at line 87 of file iocon\_5411x.h.

**6.25.2.24 #define IOCON\_S\_MODE( *clks* ) ((*clks*) << 11)**

Select clocks for digital input filter mode

Definition at line 92 of file iocon\_5411x.h.

**6.25.2.25 #define IOCON\_S\_MODE\_0CLK (0x0 << 11)**

Bypass input filter

Definition at line 88 of file iocon\_5411x.h.

**6.25.2.26 #define IOCON\_S\_MODE\_1CLK (0x1 << 11)**

Input pulses shorter than 1 filter clock are rejected

Definition at line 89 of file iocon\_5411x.h.

**6.25.2.27 #define IOCON\_S\_MODE\_2CLK (0x2 << 11)**

Input pulses shorter than 2 filter clock2 are rejected

Definition at line 90 of file iocon\_5411x.h.

6.25.2.28 `#define IOCON_S_MODE_3CLK (0x3 << 11)`

Input pulses shorter than 3 filter clock2 are rejected

Definition at line 91 of file `iocon_5411x.h`.

6.25.2.29 `#define IOCON_STDI2C_EN (0x1 << 8)`

I2C standard mode/fast-mode

Definition at line 83 of file `iocon_5411x.h`.

### 6.25.3 Function Documentation

#### 6.25.3.1 `__STATIC_INLINE void Chip_IOCON_PinMux ( LPC_IOCON_T * pIOCON, uint8_t port, uint8_t pin, uint16_t mode, uint8_t func )`

I/O Control pin mux.

Parameters

<i>pIOCON</i>	: The base of IOCON peripheral on the chip
<i>port</i>	: GPIO port to mux
<i>pin</i>	: GPIO pin to mux
<i>mode</i>	: OR'ed values or type <code>IOCON_*</code>
<i>func</i>	: Pin function, value of type <code>IOCON_FUNC?</code>

Returns

Nothing

Definition at line 117 of file `iocon_5411x.h`.

#### 6.25.3.2 `__STATIC_INLINE void Chip_IOCON_PinMuxSet ( LPC_IOCON_T * pIOCON, uint8_t port, uint8_t pin, uint32_t modefunc )`

Sets I/O Control pin mux.

Parameters

<i>pIOCON</i>	: The base of IOCON peripheral on the chip
<i>port</i>	: GPIO port to mux
<i>pin</i>	: GPIO pin to mux
<i>modefunc</i>	: OR'ed values or type <code>IOCON_*</code>

Returns

Nothing

Definition at line 103 of file `iocon_5411x.h`.

#### 6.25.3.3 `__STATIC_INLINE void Chip_IOCON_SetPinMuxing ( LPC_IOCON_T * pIOCON, const PINMUX_GRP_T * pinArray, uint32_t arrayLength )`

Set all I/O Control pin muxing.

**Parameters**

<i>pIOCON</i>	: The base of IOCON peripheral on the chip
<i>pinArray</i>	: Pointer to array of pin mux selections
<i>arrayLength</i>	: Number of entries in pinArray

**Returns**

Nothing

Definition at line 129 of file iocon\_5411x.h.



## 6.26 CHIP: LPC5411X Input Mux Registers and Driver

### 6.26.1 Detailed Description

#### Data Structures

- struct [LPC\\_INMUX\\_T](#)  
*LPC5411X Input Mux Register Block Structure.*

#### Enumerations

- enum [DMA\\_TRIGSRC\\_T](#) {  
[DMATRIG\\_ADC0\\_SEQA\\_IRQ](#) = 0, [DMATRIG\\_ADC0\\_SEQB\\_IRQ](#), [DMATRIG\\_SCT0\\_DMA0](#), [DMATRIG\\_SCT0\\_DMA1](#),  
[DMATRIG\\_TIMER0\\_MATCH0](#), [DMATRIG\\_TIMER0\\_MATCH1](#), [DMATRIG\\_TIMER1\\_MATCH0](#), [DMATRIG\\_TIMER1\\_MATCH1](#),  
[DMATRIG\\_TIMER2\\_MATCH0](#), [DMATRIG\\_TIMER2\\_MATCH1](#), [DMATRIG\\_TIMER3\\_MATCH0](#), [DMATRIG\\_TIMER3\\_MATCH1](#),  
[DMATRIG\\_TIMER4\\_MATCH0](#), [DMATRIG\\_TIMER4\\_MATCH1](#),  
[DMATRIG\\_PININT0](#), [DMATRIG\\_PININT1](#), [DMATRIG\\_PININT2](#), [DMATRIG\\_PININT3](#),  
[DMATRIG\\_OUTMUX0](#), [DMATRIG\\_OUTMUX1](#), [DMATRIG\\_OUTMUX2](#), [DMATRIG\\_OUTMUX3](#) }
- enum [FREQMSR\\_SRC\\_T](#) {  
[FREQMSR\\_CLKIN](#) = 0, [FREQMSR\\_FRO12MHZ](#), [FREQMSR\\_WDOSC](#), [FREQMSR\\_32KHZOSC](#),  
[FREQ\\_MEAS\\_MAIN\\_CLK](#), [FREQMSR\\_PIO0\\_4](#), [FREQMSR\\_PIO0\\_20](#), [FREQMSR\\_PIO0\\_24](#),  
[FREQMSR\\_PIO1\\_4](#) }

#### Functions

- `__STATIC_INLINE void Chip\_INMUX\_PinIntSel (uint8_t pintSel, uint8_t portNum, uint8_t pinNum)`  
*GPIO Pin Interrupt Pin Select (sets PINTSEL register)*
- `__STATIC_INLINE void Chip\_INMUX\_SetDMATrigger (uint8_t ch, DMA\_TRIGSRC\_T trig)`  
*Select a trigger source for a DMA channel.*
- `__STATIC_INLINE void Chip\_INMUX\_SetDMAOutMux (uint8_t index, uint8_t dmaCh)`  
*Selects a DMA trigger source for the DMATRIG\_OUTMUXn IDs.*
- `__STATIC_INLINE void Chip\_INMUX\_SetFreqMeasRefClock (FREQMSR\_SRC\_T ref)`  
*Selects a reference clock used with the frequency measure function.*
- `__STATIC_INLINE void Chip\_INMUX\_SetFreqMeasTargClock (FREQMSR\_SRC\_T targ)`  
*Selects a target clock used with the frequency measure function.*

### 6.26.2 Enumeration Type Documentation

#### 6.26.2.1 enum [DMA\\_TRIGSRC\\_T](#)

##### Enumerator

**[DMATRIG\\_ADC0\\_SEQA\\_IRQ](#)** ADC0 sequencer A interrupt as trigger  
**[DMATRIG\\_ADC0\\_SEQB\\_IRQ](#)** ADC0 sequencer B interrupt as trigger  
**[DMATRIG\\_SCT0\\_DMA0](#)** SCT 0, DMA 0 as trigger  
**[DMATRIG\\_SCT0\\_DMA1](#)** SCT 1, DMA 1 as trigger  
**[DMATRIG\\_TIMER0\\_MATCH0](#)** Timer 0, match 0 trigger  
**[DMATRIG\\_TIMER0\\_MATCH1](#)** Timer 0, match 1 trigger  
**[DMATRIG\\_TIMER1\\_MATCH0](#)** Timer 1, match 0 trigger  
**[DMATRIG\\_TIMER2\\_MATCH0](#)** Timer 2, match 0 trigger

**DMATRIG\_TIMER2\_MATCH1** Timer 2, match 1 trigger  
**DMATRIG\_TIMER3\_MATCH0** Timer 3, match 0 trigger  
**DMATRIG\_TIMER4\_MATCH0** Timer 4, match 0 trigger  
**DMATRIG\_TIMER4\_MATCH1** Timer 4, match 1 trigger  
**DMATRIG\_PININT0** Pin interrupt 0 trigger  
**DMATRIG\_PININT1** Pin interrupt 1 trigger  
**DMATRIG\_PININT2** Pin interrupt 2 trigger  
**DMATRIG\_PININT3** Pin interrupt 3 trigger  
**DMATRIG\_OUTMUX0** DMA trigger tied to this source, Select with Chip\_INMUX\_SetDMAOutMux  
**DMATRIG\_OUTMUX1** DMA trigger tied to this source, Select with Chip\_INMUX\_SetDMAOutMux  
**DMATRIG\_OUTMUX2** DMA trigger tied to this source, Select with Chip\_INMUX\_SetDMAOutMux  
**DMATRIG\_OUTMUX3** DMA trigger tied to this source, Select with Chip\_INMUX\_SetDMAOutMux

Definition at line 71 of file inmux\_5411x.h.

#### 6.26.2.2 enum **FREQMSR\_SRC\_T**

Enumerator

**FREQMSR\_CLKIN** CLKIN pin  
**FREQMSR\_FRO12MHZ** FRO 12-MHz  
**FREQMSR\_WDOSC** Watchdog oscillator  
**FREQMSR\_32KHZOSC** 32KHz (RTC) oscillator rate  
**FREQ\_MEAS\_MAIN\_CLK** main system clock  
**FREQMSR\_PIO0\_4** External pin PIO0\_4 as input rate  
**FREQMSR\_PIO0\_20** External pin PIO0\_20 as input rate  
**FREQMSR\_PIO0\_24** External pin PIO0\_24 as input rate  
**FREQMSR\_PIO1\_4** External pin PIO1\_4 as input rate

Definition at line 119 of file inmux\_5411x.h.

### 6.26.3 Function Documentation

#### 6.26.3.1 **\_\_STATIC\_INLINE void Chip\_INMUX\_PinIntSel ( uint8\_t pintSel, uint8\_t portNum, uint8\_t pinNum )**

GPIO Pin Interrupt Pin Select (sets PINTSEL register)

Parameters

<i>pintSel</i>	: GPIO PINTSEL interrupt, should be: 0 to 7
<i>portNum</i>	: GPIO port number interrupt, should be: 0 to 1
<i>pinNum</i>	: GPIO pin number Interrupt, should be: 0 to 31

Returns

Nothing

Definition at line 65 of file inmux\_5411x.h.

#### 6.26.3.2 **\_\_STATIC\_INLINE void Chip\_INMUX\_SetDMAOutMux ( uint8\_t index, uint8\_t dmaCh )**

Selects a DMA trigger source for the DMATRIG\_OUTMUXn IDs.

## Parameters

<i>index</i>	: Select 0 to 3 to sets the source for DMATRIG_OUTMUX0 to DMATRIG_OUTMUX3
<i>dmaCh</i>	: DMA channel to select for DMATRIG_OUTMUXn source

## Returns

Nothing

## Note

This function sets the DMA trigger (out) source used with the DMATRIG\_OUTMUXn trigger source.

Definition at line 113 of file inmux\_5411x.h.

### 6.26.3.3 `__STATIC_INLINE void Chip_INMUX_SetDMATrigger ( uint8_t ch, DMA_TRIGSRC_T trig )`

Select a trigger source for a DMA channel.

## Parameters

<i>ch</i>	: DMA channel number
<i>trig</i>	: Trigger source for the DMA channel

## Returns

Nothing

Definition at line 100 of file inmux\_5411x.h.

### 6.26.3.4 `__STATIC_INLINE void Chip_INMUX_SetFreqMeasRefClock ( FREQMSR_SRC_T ref )`

Selects a reference clock used with the frequency measure function.

## Parameters

<i>ref</i>	: Frequency measure function reference clock
------------	--

## Returns

Nothing

Definition at line 136 of file inmux\_5411x.h.

### 6.26.3.5 `__STATIC_INLINE void Chip_INMUX_SetFreqMeasTargClock ( FREQMSR_SRC_T targ )`

Selects a target clock used with the frequency measure function.

## Parameters

<i>targ</i>	: Frequency measure function reference clock
-------------	--

## Returns

Nothing

Definition at line 146 of file inmux\_5411x.h.

## 6.27 CHIP: LPC5411X M0 core CMSIS include file

### 6.27.1 Detailed Description

#### Modules

- [CHIP: LPC5411X M0 core Cortex CMSIS definitions](#)
- [CHIP\\_5411X: LPC5411X M0 core peripheral interrupt numbers](#)

## 6.28 CHIP: LPC5411X M0 core Cortex CMSIS definitions

### 6.28.1 Detailed Description

#### Macros

- #define `__CM0PLUS_REV` 0x0001
- #define `__MPU_PRESENT` 0
- #define `__NVIC_PRIO_BITS` 2
- #define `__Vendor_SysTickConfig` 0
- #define `__VTOR_PRESENT` 1
- #define `TIMER0_IRQn` CT32B0\_IRQn  
*interrupt Alias*
- #define `TIMER1_IRQn` CT32B1\_IRQn
- #define `TIMER2_IRQn` CT32B2\_IRQn
- #define `TIMER3_IRQn` CT32B3\_IRQn
- #define `TIMER4_IRQn` CT32B4\_IRQn
- #define `SCT_IRQn` SCT0\_IRQn
- #define `ADC0_SEQA_IRQn` ADC\_SEQA\_IRQn
- #define `ADC0_SEQB_IRQn` ADC\_SEQB\_IRQn
- #define `ADC0_THCMP_IRQn` ADC\_THCMP\_IRQn
- #define `TIMER0_IRQHandler` CT32B0\_IRQHandler  
*Interrupt handler Alias.*
- #define `TIMER1_IRQHandler` CT32B1\_IRQHandler
- #define `TIMER2_IRQHandler` CT32B2\_IRQHandler
- #define `TIMER3_IRQHandler` CT32B3\_IRQHandler
- #define `TIMER4_IRQHandler` CT32B4\_IRQHandler
- #define `SCT_IRQHandler` SCT0\_IRQHandler
- #define `ADC0_SEQA_IRQHandler` ADC\_SEQA\_IRQHandler
- #define `ADC0_SEQB_IRQHandler` ADC\_SEQB\_IRQHandler
- #define `ADC0_THCMP_IRQHandler` ADC\_THCMP\_IRQHandler

### 6.28.2 Macro Definition Documentation

#### 6.28.2.1 #define `__CM0PLUS_REV` 0x0001

Cortex-M0PLUS Core Revision

Definition at line 136 of file cmsis\_5411x\_m0.h.

#### 6.28.2.2 #define `__MPU_PRESENT` 0

MPU present or not

Definition at line 137 of file cmsis\_5411x\_m0.h.

#### 6.28.2.3 #define `__NVIC_PRIO_BITS` 2

Number of Bits used for Priority Levels

Definition at line 138 of file cmsis\_5411x\_m0.h.

#### 6.28.2.4 `#define __Vendor_SysTickConfig 0`

Set to 1 if different SysTick Config is used

Definition at line 139 of file `cmsis_5411x_m0.h`.

#### 6.28.2.5 `#define __VTOR_PRESENT 1`

Definition at line 140 of file `cmsis_5411x_m0.h`.

#### 6.28.2.6 `#define ADC0_SEQA_IRQHandler ADC_SEQA_IRQHandler`

Definition at line 160 of file `cmsis_5411x_m0.h`.

#### 6.28.2.7 `#define ADC0_SEQA_IRQn ADC_SEQA_IRQn`

Definition at line 149 of file `cmsis_5411x_m0.h`.

#### 6.28.2.8 `#define ADC0_SEQB_IRQHandler ADC_SEQB_IRQHandler`

Definition at line 161 of file `cmsis_5411x_m0.h`.

#### 6.28.2.9 `#define ADC0_SEQB_IRQn ADC_SEQB_IRQn`

Definition at line 150 of file `cmsis_5411x_m0.h`.

#### 6.28.2.10 `#define ADC0_THCMP_IRQHandler ADC_THCMP_IRQHandler`

Definition at line 162 of file `cmsis_5411x_m0.h`.

#### 6.28.2.11 `#define ADC0_THCMP_IRQn ADC_THCMP_IRQn`

Definition at line 151 of file `cmsis_5411x_m0.h`.

#### 6.28.2.12 `#define SCT_IRQHandler SCT0_IRQHandler`

Definition at line 159 of file `cmsis_5411x_m0.h`.

#### 6.28.2.13 `#define SCT_IRQn SCT0_IRQn`

Definition at line 148 of file `cmsis_5411x_m0.h`.

#### 6.28.2.14 `#define TIMER0_IRQHandler CT32B0_IRQHandler`

Interrupt handler Alias.

Definition at line 154 of file `cmsis_5411x_m0.h`.

**6.28.2.15 #define TIMER0\_IRQn CT32B0\_IRQn**

interrupt Alias

Definition at line 143 of file cmsis\_5411x\_m0.h.

**6.28.2.16 #define TIMER1\_IRQHandler CT32B1\_IRQHandler**

Definition at line 155 of file cmsis\_5411x\_m0.h.

**6.28.2.17 #define TIMER1\_IRQn CT32B1\_IRQn**

Definition at line 144 of file cmsis\_5411x\_m0.h.

**6.28.2.18 #define TIMER2\_IRQHandler CT32B2\_IRQHandler**

Definition at line 156 of file cmsis\_5411x\_m0.h.

**6.28.2.19 #define TIMER2\_IRQn CT32B2\_IRQn**

Definition at line 145 of file cmsis\_5411x\_m0.h.

**6.28.2.20 #define TIMER3\_IRQHandler CT32B3\_IRQHandler**

Definition at line 157 of file cmsis\_5411x\_m0.h.

**6.28.2.21 #define TIMER3\_IRQn CT32B3\_IRQn**

Definition at line 146 of file cmsis\_5411x\_m0.h.

**6.28.2.22 #define TIMER4\_IRQHandler CT32B4\_IRQHandler**

Definition at line 158 of file cmsis\_5411x\_m0.h.

**6.28.2.23 #define TIMER4\_IRQn CT32B4\_IRQn**

Definition at line 147 of file cmsis\_5411x\_m0.h.

## 6.29 CHIP: LPC5411X M4 core CMSIS include file

### 6.29.1 Detailed Description

#### Modules

- [CHIP: LPC5411X M4 core Cortex CMSIS definitions](#)
- [CHIP\\_5411X: LPC5411X M4 core peripheral interrupt numbers](#)



## 6.30 CHIP: LPC5411X M4 core Cortex CMSIS definitions

### 6.30.1 Detailed Description

#### Macros

- #define `__CM4_REV` 0x0001
- #define `__MPU_PRESENT` 1
- #define `__NVIC_PRIO_BITS` 3
- #define `__Vendor_SysTickConfig` 0
- #define `__FPU_PRESENT` 1
- #define `TIMER0_IRQn` CT32B0\_IRQn  
*interrupt Alias*
- #define `TIMER1_IRQn` CT32B1\_IRQn
- #define `TIMER2_IRQn` CT32B2\_IRQn
- #define `TIMER3_IRQn` CT32B3\_IRQn
- #define `TIMER4_IRQn` CT32B4\_IRQn
- #define `SCT_IRQn` SCT0\_IRQn
- #define `ADC0_SEQA_IRQn` ADC\_SEQA\_IRQn
- #define `ADC0_SEQB_IRQn` ADC\_SEQB\_IRQn
- #define `ADC0_THCMP_IRQn` ADC\_THCMP\_IRQn
- #define `TIMER0_IRQHandler` CT32B0\_IRQHandler  
*Interrupt handler Alias.*
- #define `TIMER1_IRQHandler` CT32B1\_IRQHandler
- #define `TIMER2_IRQHandler` CT32B2\_IRQHandler
- #define `TIMER3_IRQHandler` CT32B3\_IRQHandler
- #define `TIMER4_IRQHandler` CT32B4\_IRQHandler
- #define `SCT_IRQHandler` SCT0\_IRQHandler
- #define `ADC0_SEQA_IRQHandler` ADC\_SEQA\_IRQHandler
- #define `ADC0_SEQB_IRQHandler` ADC\_SEQB\_IRQHandler
- #define `ADC0_THCMP_IRQHandler` ADC\_THCMP\_IRQHandler

### 6.30.2 Macro Definition Documentation

#### 6.30.2.1 #define `__CM4_REV` 0x0001

Cortex-M4 Core Revision

Definition at line 146 of file cmsis\_5411x.h.

#### 6.30.2.2 #define `__FPU_PRESENT` 1

Definition at line 150 of file cmsis\_5411x.h.

#### 6.30.2.3 #define `__MPU_PRESENT` 1

MPU present or not

Definition at line 147 of file cmsis\_5411x.h.

#### 6.30.2.4 #define `__NVIC_PRIO_BITS` 3

Number of Bits used for Priority Levels

Definition at line 148 of file cmsis\_5411x.h.

**6.30.2.5 #define \_\_Vendor\_SysTickConfig 0**

Set to 1 if different SysTick Config is used

Definition at line 149 of file cmsis\_5411x.h.

**6.30.2.6 #define ADC0\_SEQA\_IRQHandler ADC\_SEQA\_IRQHandler**

Definition at line 170 of file cmsis\_5411x.h.

**6.30.2.7 #define ADC0\_SEQA\_IRQn ADC\_SEQA\_IRQn**

Definition at line 159 of file cmsis\_5411x.h.

**6.30.2.8 #define ADC0\_SEQB\_IRQHandler ADC\_SEQB\_IRQHandler**

Definition at line 171 of file cmsis\_5411x.h.

**6.30.2.9 #define ADC0\_SEQB\_IRQn ADC\_SEQB\_IRQn**

Definition at line 160 of file cmsis\_5411x.h.

**6.30.2.10 #define ADC0\_THCMP\_IRQHandler ADC\_THCMP\_IRQHandler**

Definition at line 172 of file cmsis\_5411x.h.

**6.30.2.11 #define ADC0\_THCMP\_IRQn ADC\_THCMP\_IRQn**

Definition at line 161 of file cmsis\_5411x.h.

**6.30.2.12 #define SCT\_IRQHandler SCT0\_IRQHandler**

Definition at line 169 of file cmsis\_5411x.h.

**6.30.2.13 #define SCT\_IRQn SCT0\_IRQn**

Definition at line 158 of file cmsis\_5411x.h.

**6.30.2.14 #define TIMER0\_IRQHandler CT32B0\_IRQHandler**

Interrupt handler Alias.

Definition at line 164 of file cmsis\_5411x.h.

**6.30.2.15 #define TIMER0\_IRQn CT32B0\_IRQn**

interrupt Alias

Definition at line 153 of file cmsis\_5411x.h.

6.30.2.16 **#define TIMER1\_IRQHandler CT32B1\_IRQHandler**

Definition at line 165 of file cmsis\_5411x.h.

6.30.2.17 **#define TIMER1\_IRQn CT32B1\_IRQn**

Definition at line 154 of file cmsis\_5411x.h.

6.30.2.18 **#define TIMER2\_IRQHandler CT32B2\_IRQHandler**

Definition at line 166 of file cmsis\_5411x.h.

6.30.2.19 **#define TIMER2\_IRQn CT32B2\_IRQn**

Definition at line 155 of file cmsis\_5411x.h.

6.30.2.20 **#define TIMER3\_IRQHandler CT32B3\_IRQHandler**

Definition at line 167 of file cmsis\_5411x.h.

6.30.2.21 **#define TIMER3\_IRQn CT32B3\_IRQn**

Definition at line 156 of file cmsis\_5411x.h.

6.30.2.22 **#define TIMER4\_IRQHandler CT32B4\_IRQHandler**

Definition at line 168 of file cmsis\_5411x.h.

6.30.2.23 **#define TIMER4\_IRQn CT32B4\_IRQn**

Definition at line 157 of file cmsis\_5411x.h.

## 6.31 CHIP: LPC5411X Mailbox M4/M0+ driver

### 6.31.1 Detailed Description

#### Data Structures

- struct [LPC\\_MBOXIRQ\\_T](#)
- struct [LPC\\_MBOX\\_T](#)

#### Macros

- `#define MAILBOX_AVAIL (MAILBOX_CM4 + 1) /* Number of available mailboxes */`

#### Enumerations

- enum [MBOX\\_IDX\\_T](#) { [MAILBOX\\_CM0PLUS](#) = 0, [MAILBOX\\_CM4](#) }

#### Functions

- `__STATIC_INLINE void Chip_MBOX_Init (LPC_MBOX_T *pMBOX)`  
*Initialize mailbox.*
- `__STATIC_INLINE void Chip_MBOX_DeInit (LPC_MBOX_T *pMBOX)`  
*Shutdown mailbox.*
- `__STATIC_INLINE void Chip_MBOX_SetValue (LPC_MBOX_T *pMBOX, uint32_t cpu_id, uint32_t mbox↵  
Data)`  
*Set data value in the mailbox based on the CPU ID.*
- `__STATIC_INLINE void Chip_MBOX_SetValueBits (LPC_MBOX_T *pMBOX, uint32_t cpu_id, uint32_t ↵  
t mboxSetBits)`  
*Set data bits in the mailbox based on the CPU ID.*
- `__STATIC_INLINE void Chip_MBOX_ClearValueBits (LPC_MBOX_T *pMBOX, uint32_t cpu_id, uint32_t ↵  
t mboxClrBits)`  
*Clear data bits in the mailbox based on the CPU ID.*
- `__STATIC_INLINE uint32_t Chip_MBOX_GetValue (LPC_MBOX_T *pMBOX, uint32_t cpu_id)`  
*Get data in the mailbox based on the cpu\_id.*
- `__STATIC_INLINE uint32_t Chip_MBOX_GetMutex (LPC_MBOX_T *pMBOX)`  
*Get MUTEX state and lock mutex.*
- `__STATIC_INLINE void Chip_MBOX_SetMutex (LPC_MBOX_T *pMBOX)`  
*Set MUTEX state.*

### 6.31.2 Macro Definition Documentation

#### 6.31.2.1 `#define MAILBOX_AVAIL (MAILBOX_CM4 + 1) /* Number of available mailboxes */`

Definition at line 49 of file mailbox\_5411x.h.

### 6.31.3 Enumeration Type Documentation

#### 6.31.3.1 enum [MBOX\\_IDX\\_T](#)

##### Enumerator

**`MAILBOX_CM0PLUS`**

**MAILBOX\_CM4**

Definition at line 45 of file mailbox\_5411x.h.

**6.31.4 Function Documentation**
**6.31.4.1 `__STATIC_INLINE void Chip_MBOX_ClearValueBits ( LPC_MBOX_T * pMBOX, uint32_t cpu_id, uint32_t mboxClrBits )`**

Clear data bits in the mailbox based on the CPU ID.

**Parameters**

<i>pMBOX</i>	: Pointer to the mailbox register structure
<i>cpu_id</i>	: MAILBOX_CM0PLUS is M0+ or MAILBOX_CM4 is M4
<i>mboxClrBits</i>	: data bits to clear in the mailbox

**Returns**

Nothing

**Note**

Clear data bits to send via the MBOX to the other core. A value of 0 will do nothing. Only clears bits selected with a 1 in it's bit position.

Definition at line 124 of file mailbox\_5411x.h.

**6.31.4.2 `__STATIC_INLINE void Chip_MBOX_DeInit ( LPC_MBOX_T * pMBOX )`**

Shutdown mailbox.

**Parameters**

<i>pMBOX</i>	: Pointer to the mailbox register structure
--------------	---

**Returns**

Nothing

Definition at line 83 of file mailbox\_5411x.h.

**6.31.4.3 `__STATIC_INLINE uint32_t Chip_MBOX_GetMutex ( LPC_MBOX_T * pMBOX )`**

Get MUTEX state and lock mutex.

**Parameters**

<i>pMBOX</i>	: Pointer to the mailbox register structure
--------------	---

**Returns**

See note

**Note**

Returns '1' if the mutex was taken or '0' if another resources has the mutex locked. Once a mutex is taken, it can be returned with the [Chip\\_MBOX\\_SetMutex\(\)](#) function.

Definition at line 148 of file mailbox\_5411x.h.

6.31.4.4 `__STATIC_INLINE uint32_t Chip_MBOX_GetValue ( LPC_MBOX_T * pMBOX, uint32_t cpu_id )`

Get data in the mailbox based on the `cpu_id`.

## Parameters

<i>pMBOX</i>	: Pointer to the mailbox register structure
<i>cpu_id</i>	: MAILBOX_CM0PLUS is M0+ or MAILBOX_CM4 is M4

## Returns

Current mailbox data

Definition at line 135 of file mailbox\_5411x.h.

6.31.4.5 `__STATIC_INLINE void Chip_MBOX_Init ( LPC_MBOX_T * pMBOX )`

Initialize mailbox.

## Parameters

<i>pMBOX</i>	: Pointer to the mailbox register structure
--------------	---

## Returns

Nothing

## Note

Even if both cores use the mailbox, only 1 core should initialize it.

Definition at line 73 of file mailbox\_5411x.h.

6.31.4.6 `__STATIC_INLINE void Chip_MBOX_SetMutex ( LPC_MBOX_T * pMBOX )`

Set MUTEX state.

## Parameters

<i>pMBOX</i>	: Pointer to the mailbox register structure
--------------	---

## Returns

Nothing

## Note

Sets mutex state to '1' and allows other resources to get the mutex

Definition at line 159 of file mailbox\_5411x.h.

6.31.4.7 `__STATIC_INLINE void Chip_MBOX_SetValue ( LPC_MBOX_T * pMBOX, uint32_t cpu_id, uint32_t mboxData )`

Set data value in the mailbox based on the CPU ID.

## Parameters

<i>pMBOX</i>	: Pointer to the mailbox register structure
<i>cpu_id</i>	: MAILBOX_CM0PLUS is M0+ or MAILBOX_CM4 is M4
<i>mboxData</i>	: data to send in the mailbox

**Returns**

Nothing

**Note**

Sets a data value to send via the MBOX to the other core.

Definition at line 96 of file mailbox\_5411x.h.

6.31.4.8 `__STATIC_INLINE void Chip_MBOX_SetValueBits ( LPC_MBOX_T * pMBOX, uint32_t cpu_id, uint32_t mboxSetBits )`

Set data bits in the mailbox based on the CPU ID.

**Parameters**

<i>pMBOX</i>	: Pointer to the mailbox register structure
<i>cpu_id</i>	: MAILBOX_CM0PLUS is M0+ or MAILBOX_CM4 is M4
<i>mboxSetBits</i>	: data bits to set in the mailbox

**Returns**

Nothing

**Note**

Sets data bits to send via the MBOX to the other core, A value of 0 will do nothing. Only sets bits selected with a 1 in it's bit position.

Definition at line 110 of file mailbox\_5411x.h.



## 6.32 CHIP: LPC5411X Micro Tick driver

### 6.32.1 Detailed Description

#### Data Structures

- struct [LPC\\_UTICK\\_T](#)  
*Micro Tick register block structure.*

#### Macros

- #define [UTICK\\_CTRL\\_REPEAT](#) ((uint32\_t) 1UL << 31)  
*UTick register definitions.*
- #define [UTICK\\_CTRL\\_DELAY\\_MASK](#) ((uint32\_t) 0x7FFFFFFF)
- #define [UTICK\\_STATUS\\_INTR](#) ((uint32\_t) 1 << 0)
- #define [UTICK\\_STATUS\\_ACTIVE](#) ((uint32\_t) 1 << 1)
- #define [UTICK\\_STATUS\\_MASK](#) ((uint32\_t) 0x03)

#### Functions

- \_\_STATIC\_INLINE void [Chip\\_UTICK\\_Init](#) (LPC\_UTICK\_T \*pUTICK)  
*Initialize the UTICK peripheral.*
- \_\_STATIC\_INLINE void [Chip\\_UTICK\\_DeInit](#) (LPC\_UTICK\_T \*pUTICK)  
*De-initialize the UTICK peripheral.*
- \_\_STATIC\_INLINE void [Chip\\_UTICK\\_SetTick](#) (LPC\_UTICK\_T \*pUTICK, uint32\_t tick\_value, bool repeat)  
*Setup UTICK.*
- \_\_STATIC\_INLINE void [Chip\\_UTICK\\_SetDelayMs](#) (LPC\_UTICK\_T \*pUTICK, uint32\_t delayMs, bool repeat)  
*Setup UTICK for the passed delay (in mS)*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_UTICK\\_GetTick](#) (LPC\_UTICK\_T \*pUTICK)  
*Read UTICK Value.*
- \_\_STATIC\_INLINE void [Chip\\_UTICK\\_Halt](#) (LPC\_UTICK\_T \*pUTICK)  
*Halt UTICK timer.*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_UTICK\\_GetStatus](#) (LPC\_UTICK\_T \*pUTICK)  
*Returns the status of UTICK.*
- \_\_STATIC\_INLINE void [Chip\\_UTICK\\_ClearInterrupt](#) (LPC\_UTICK\_T \*pUTICK)  
*Clears UTICK Interrupt flag.*

### 6.32.2 Macro Definition Documentation

#### 6.32.2.1 #define UTICK\_CTRL\_DELAY\_MASK ((uint32\_t) 0x7FFFFFFF)

UTick Delay Value Mask

Definition at line 58 of file utick\_5411x.h.

#### 6.32.2.2 #define UTICK\_CTRL\_REPEAT ((uint32\_t) 1UL << 31)

UTick register definitions.

UTick repeat delay bit

Definition at line 56 of file utick\_5411x.h.

### 6.32.2.3 `#define UTICK_STATUS_ACTIVE ((uint32_t) 1 << 1)`

UTick Active Status bit

Definition at line 62 of file `utick_5411x.h`.

### 6.32.2.4 `#define UTICK_STATUS_INTR ((uint32_t) 1 << 0)`

UTick Interrupt Status bit

Definition at line 60 of file `utick_5411x.h`.

### 6.32.2.5 `#define UTICK_STATUS_MASK ((uint32_t) 0x03)`

UTick Status Register Mask

Definition at line 64 of file `utick_5411x.h`.

## 6.32.3 Function Documentation

### 6.32.3.1 `__STATIC_INLINE void Chip_UTICK_ClearInterrupt ( LPC_UTICK_T * pUTICK )`

Clears UTICK Interrupt flag.

Parameters

<i>pUTICK</i>	: The base address of UTICK block
---------------	-----------------------------------

Returns

Nothing

Definition at line 161 of file `utick_5411x.h`.

### 6.32.3.2 `__STATIC_INLINE void Chip_UTICK_DeInit ( LPC_UTICK_T * pUTICK )`

De-initialize the UTICK peripheral.

Parameters

<i>pUTICK</i>	: UTICK peripheral selected
---------------	-----------------------------

Returns

Nothing

Definition at line 82 of file `utick_5411x.h`.

### 6.32.3.3 `__STATIC_INLINE uint32_t Chip_UTICK_GetStatus ( LPC_UTICK_T * pUTICK )`

Returns the status of UTICK.

Parameters

---

<i>pUTICK</i>	: The base address of UTICK block
---------------	-----------------------------------

**Returns**

Micro tick timer status register value

Definition at line 151 of file utick\_5411x.h.

#### 6.32.3.4 `__STATIC_INLINE uint32_t Chip_UTICK_GetTick ( LPC_UTICK_T * pUTICK )`

Read UTICK Value.

**Parameters**

<i>pUTICK</i>	: The base address of UTICK block
---------------	-----------------------------------

**Returns**

Current tick value

Definition at line 131 of file utick\_5411x.h.

#### 6.32.3.5 `__STATIC_INLINE void Chip_UTICK_Halt ( LPC_UTICK_T * pUTICK )`

Halt UTICK timer.

**Parameters**

<i>pUTICK</i>	: The base address of UTICK block
---------------	-----------------------------------

**Returns**

Nothing

Definition at line 141 of file utick\_5411x.h.

#### 6.32.3.6 `__STATIC_INLINE void Chip_UTICK_Init ( LPC_UTICK_T * pUTICK )`

Initialize the UTICK peripheral.

**Parameters**

<i>pUTICK</i>	: UTICK peripheral selected
---------------	-----------------------------

**Returns**

Nothing

Definition at line 71 of file utick\_5411x.h.

#### 6.32.3.7 `__STATIC_INLINE void Chip_UTICK_SetDelayMs ( LPC_UTICK_T * pUTICK, uint32_t delayMs, bool repeat )`

Setup UTICK for the passed delay (in mS)

**Parameters**

<i>pUTICK</i>	: The base address of UTICK block
<i>delayMs</i>	: Delay value in mS (Maximum is 1000mS)
<i>repeat</i>	: If true then delay repeats continuously else it is one time

**Returns**

Nothing

**Note**

The WDT oscillator runs at about 500KHz, so delays in uS won't be too accurate.

Definition at line 112 of file utick\_5411x.h.

6.32.3.8 `__STATIC_INLINE void Chip_UTICK_SetTick ( LPC_UTICK_T * pUTICK, uint32_t tick_value, bool repeat )`

Setup UTICK.

**Parameters**

<i>pUTICK</i>	: The base address of UTICK block
<i>tick_value</i>	: Tick value, should not exceed UTICK_CTRL_DELAY_MASK
<i>repeat</i>	: If true then delay repeats continuously else it is one time

**Returns**

Nothing

Definition at line 94 of file utick\_5411x.h.

## 6.33 CHIP: LPC5411X Multi-Rate Timer driver

### 6.33.1 Detailed Description

#### Data Structures

- struct [LPC\\_MRT\\_CH\\_T](#)  
*MRT register block structure.*
- struct [LPC\\_MRT\\_T](#)  
*MRT register block structure.*

#### Macros

- #define [MRT\\_CHANNELS\\_NUM](#) (4)  
*LPC5411X MRT chip configuration.*
- #define [MRT\\_NO\\_IDLE\\_CHANNEL](#) (0x40)
- #define [MRT\\_INTVAL\\_IVALUE](#) (0xFFFFFFF) /\* Maximum interval load value and mask \*/  
*MRT register bit fields & masks.*
- #define [MRT\\_INTVAL\\_LOAD](#) (0x80000000UL) /\* Force immediate load of timer interval register bit \*/
- #define [MRT\\_CTRL\\_INTEN\\_MASK](#) (0x01)
- #define [MRT\\_CTRL\\_MODE\\_MASK](#) (0x06)
- #define [MRT\\_STAT\\_INTFLAG](#) (0x01)
- #define [MRT\\_STAT\\_RUNNING](#) (0x02)
- #define [LPC\\_MRT\\_CH0](#) (([LPC\\_MRT\\_CH\\_T](#) \*) &[LPC\\_MRT](#)->CHANNEL[0])
- #define [LPC\\_MRT\\_CH1](#) (([LPC\\_MRT\\_CH\\_T](#) \*) &[LPC\\_MRT](#)->CHANNEL[1])
- #define [LPC\\_MRT\\_CH2](#) (([LPC\\_MRT\\_CH\\_T](#) \*) &[LPC\\_MRT](#)->CHANNEL[2])
- #define [LPC\\_MRT\\_CH3](#) (([LPC\\_MRT\\_CH\\_T](#) \*) &[LPC\\_MRT](#)->CHANNEL[3])
- #define [LPC\\_MRT\\_CH](#)(ch) (([LPC\\_MRT\\_CH\\_T](#) \*) &[LPC\\_MRT](#)->CHANNEL[(ch)])
- #define [MRT0\\_INTFLAG](#) (1)
- #define [MRT1\\_INTFLAG](#) (2)
- #define [MRT2\\_INTFLAG](#) (4)
- #define [MRT3\\_INTFLAG](#) (8)
- #define [MRTn\\_INTFLAG](#)(ch) (1 << (ch))

#### Enumerations

- enum [MRT\\_MODE\\_T](#) { [MRT\\_MODE\\_REPEAT](#) = (0 << 1), [MRT\\_MODE\\_ONESHOT](#) = (1 << 1) }  
*MRT Interrupt Modes enum.*

#### Functions

- `__STATIC_INLINE void Chip\_MRT\_Init (void)`  
*Initializes the MRT.*
- `__STATIC_INLINE void Chip\_MRT\_DeInit (void)`  
*De-initializes the MRT Channel.*
- `__STATIC_INLINE LPC\_MRT\_CH\_T * Chip\_MRT\_GetRegPtr (uint8_t ch)`  
*Returns a pointer to the register block for a MRT channel.*
- `__STATIC_INLINE uint32_t Chip\_MRT\_GetInterval (LPC\_MRT\_CH\_T *pMRT)`  
*Returns the timer time interval value.*
- `__STATIC_INLINE void Chip\_MRT\_SetInterval (LPC\_MRT\_CH\_T *pMRT, uint32_t interval)`  
*Sets the timer time interval value.*

- `__STATIC_INLINE uint32_t Chip_MRT_GetTimer (LPC_MRT_CH_T *pMRT)`  
*Returns the current timer value.*
- `__STATIC_INLINE bool Chip_MRT_GetEnabled (LPC_MRT_CH_T *pMRT)`  
*Returns true if the timer is enabled.*
- `__STATIC_INLINE void Chip_MRT_SetEnabled (LPC_MRT_CH_T *pMRT)`  
*Enables the timer.*
- `__STATIC_INLINE void Chip_MRT_SetDisabled (LPC_MRT_CH_T *pMRT)`  
*Disables the timer.*
- `__STATIC_INLINE MRT_MODE_T Chip_MRT_GetMode (LPC_MRT_CH_T *pMRT)`  
*Returns the timer mode (repeat or one-shot)*
- `__STATIC_INLINE void Chip_MRT_SetMode (LPC_MRT_CH_T *pMRT, MRT_MODE_T mode)`  
*Sets the timer mode (repeat or one-shot)*
- `__STATIC_INLINE bool Chip_MRT_IsRepeatMode (LPC_MRT_CH_T *pMRT)`  
*Check if the timer is configured in repeat mode.*
- `__STATIC_INLINE bool Chip_MRT_IsOneShotMode (LPC_MRT_CH_T *pMRT)`  
*Check if the timer is configured in one-shot mode.*
- `__STATIC_INLINE bool Chip_MRT_IntPending (LPC_MRT_CH_T *pMRT)`  
*Check if the timer has an interrupt pending.*
- `__STATIC_INLINE void Chip_MRT_IntClear (LPC_MRT_CH_T *pMRT)`  
*Clears the pending interrupt (if any)*
- `__STATIC_INLINE bool Chip_MRT_Running (LPC_MRT_CH_T *pMRT)`  
*Check if the timer is running.*
- `__STATIC_INLINE uint8_t Chip_MRT_GetIdleChannel (void)`  
*Returns the IDLE channel value.*
- `__STATIC_INLINE uint8_t Chip_MRT_GetIdleChannelShifted (void)`  
*Returns the IDLE channel value.*
- `__STATIC_INLINE uint32_t Chip_MRT_GetIntPending (void)`  
*Returns the interrupt pending status for all MRT channels.*
- `__STATIC_INLINE bool Chip_MRT_GetIntPendingByChannel (uint8_t ch)`  
*Returns the interrupt pending status for a singel MRT channel.*
- `__STATIC_INLINE void Chip_MRT_ClearIntPending (uint32_t mask)`  
*Clears the interrupt pending status for one or more MRT channels.*

## 6.33.2 Macro Definition Documentation

6.33.2.1 `#define LPC_MRT_CH( ch ) ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[(ch)])`

Definition at line 99 of file `mrt_5411x.h`.

6.33.2.2 `#define LPC_MRT_CH0 ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[0])`

Definition at line 95 of file `mrt_5411x.h`.

6.33.2.3 `#define LPC_MRT_CH1 ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[1])`

Definition at line 96 of file `mrt_5411x.h`.

6.33.2.4 `#define LPC_MRT_CH2 ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[2])`

Definition at line 97 of file `mrt_5411x.h`.

6.33.2.5 `#define LPC_MRT_CH3 ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[3])`

Definition at line 98 of file `mrt_5411x.h`.

6.33.2.6 `#define MRT0_INTFLAG (1)`

Definition at line 102 of file `mrt_5411x.h`.

6.33.2.7 `#define MRT1_INTFLAG (2)`

Definition at line 103 of file `mrt_5411x.h`.

6.33.2.8 `#define MRT2_INTFLAG (4)`

Definition at line 104 of file `mrt_5411x.h`.

6.33.2.9 `#define MRT3_INTFLAG (8)`

Definition at line 105 of file `mrt_5411x.h`.

6.33.2.10 `#define MRT_CHANNELS_NUM (4)`

LPC5411X MRT chip configuration.

Definition at line 47 of file `mrt_5411x.h`.

6.33.2.11 `#define MRT_CTRL_INTEN_MASK (0x01)`

Definition at line 87 of file `mrt_5411x.h`.

6.33.2.12 `#define MRT_CTRL_MODE_MASK (0x06)`

Definition at line 88 of file `mrt_5411x.h`.

6.33.2.13 `#define MRT_INTVAL_IVALUE (0xFFFFF) /* Maximum interval load value and mask */`

MRT register bit fields & masks.

Definition at line 83 of file `mrt_5411x.h`.

6.33.2.14 `#define MRT_INTVAL_LOAD (0x80000000UL) /* Force immediate load of timer interval register bit */`

Definition at line 84 of file `mrt_5411x.h`.

6.33.2.15 `#define MRT_NO_IDLE_CHANNEL (0x40)`

Definition at line 48 of file `mrt_5411x.h`.

6.33.2.16 `#define MRT_STAT_INTFLAG (0x01)`

Definition at line 91 of file `mrt_5411x.h`.

#### 6.33.2.17 `#define MRT_STAT_RUNNING (0x02)`

Definition at line 92 of file `mrt_5411x.h`.

#### 6.33.2.18 `#define MRTn_INTFLAG( ch )(1 << (ch))`

Definition at line 106 of file `mrt_5411x.h`.

### 6.33.3 Enumeration Type Documentation

#### 6.33.3.1 `enum MRT_MODE_T`

MRT Interrupt Modes enum.

Enumerator

***MRT\_MODE\_REPEAT*** MRT Repeat interrupt mode

***MRT\_MODE\_ONESHOT*** MRT One-shot interrupt mode

Definition at line 74 of file `mrt_5411x.h`.

### 6.33.4 Function Documentation

#### 6.33.4.1 `__STATIC_INLINE void Chip_MRT_ClearIntPending ( uint32_t mask )`

Clears the interrupt pending status for one or more MRT channels.

Parameters

<i>mask</i>	: Channels to clear (bit 0 = MRT0, bit 1 = MRT1, etc.)
-------------	--

Returns

Nothing

Note

Use this function to clear multiple interrupt pending states in a single call via the `IRQ_FLAG` register. Performs the same function for all MRT channels in a single call as the [Chip\\_MRT\\_IntClear\(\)](#) does for a single channel.

Definition at line 327 of file `mrt_5411x.h`.

#### 6.33.4.2 `__STATIC_INLINE void Chip_MRT_DeInit ( void )`

De-initializes the MRT Channel.

Returns

Nothing

Definition at line 125 of file `mrt_5411x.h`.

#### 6.33.4.3 `__STATIC_INLINE bool Chip_MRT_GetEnabled ( LPC_MRT_CH_T * pMRT )`

Returns true if the timer is enabled.



## Parameters

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

## Returns

True if enabled, False if not enabled

Definition at line 182 of file mrt\_5411x.h.

6.33.4.4 `__STATIC_INLINE uint8_t Chip_MRT_GetIdleChannel ( void )`

Returns the IDLE channel value.

## Returns

IDLE channel value (unshifted in bits 7..4)

Definition at line 285 of file mrt\_5411x.h.

6.33.4.5 `__STATIC_INLINE uint8_t Chip_MRT_GetIdleChannelShifted ( void )`

Returns the IDLE channel value.

## Returns

IDLE channel value (shifted in bits 3..0)

Definition at line 294 of file mrt\_5411x.h.

6.33.4.6 `__STATIC_INLINE uint32_t Chip_MRT_GetInterval ( LPC_MRT_CH_T * pMRT )`

Returns the timer time interval value.

## Parameters

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

## Returns

Timer time interval value (IVALUE)

Definition at line 146 of file mrt\_5411x.h.

6.33.4.7 `__STATIC_INLINE uint32_t Chip_MRT_GetIntPending ( void )`

Returns the interrupt pending status for all MRT channels.

## Returns

IRQ pending channel bitfield(bit 0 = MRT0, bit 1 = MRT1, etc.)

Definition at line 303 of file mrt\_5411x.h.

6.33.4.8 `__STATIC_INLINE bool Chip_MRT_GetIntPendingByChannel ( uint8_t ch )`

Returns the interrupt pending status for a single MRT channel.

**Parameters**

<i>ch</i>	: Channel to check pending interrupt status for
-----------	---

**Returns**

IRQ pending channel number

Definition at line 313 of file mrt\_5411x.h.

#### 6.33.4.9 `__STATIC_INLINE MRT_MODE_T Chip_MRT_GetMode ( LPC_MRT_CH_T * pMRT )`

Returns the timer mode (repeat or one-shot)

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

**Returns**

The current timer mode

Definition at line 212 of file mrt\_5411x.h.

#### 6.33.4.10 `__STATIC_INLINE LPC_MRT_CH_T* Chip_MRT_GetRegPtr ( uint8_t ch )`

Returns a pointer to the register block for a MRT channel.

**Parameters**

<i>ch</i>	: MRT channel to get register block for (0..3)
-----------	--

**Returns**

Pointer to the MRT register block for the channel

Definition at line 136 of file mrt\_5411x.h.

#### 6.33.4.11 `__STATIC_INLINE uint32_t Chip_MRT_GetTimer ( LPC_MRT_CH_T * pMRT )`

Returns the current timer value.

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

**Returns**

The current timer value

Definition at line 172 of file mrt\_5411x.h.

#### 6.33.4.12 `__STATIC_INLINE void Chip_MRT_Init ( void )`

Initializes the MRT.

**Returns**

Nothing

Definition at line 112 of file mrt\_5411x.h.

**6.33.4.13** `__STATIC_INLINE void Chip_MRT_IntClear ( LPC_MRT_CH_T * pMRT )`

Clears the pending interrupt (if any)

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

**Returns**

Nothing

Definition at line 266 of file mrt\_5411x.h.

**6.33.4.14** `__STATIC_INLINE bool Chip_MRT_IntPending ( LPC_MRT_CH_T * pMRT )`

Check if the timer has an interrupt pending.

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

**Returns**

True if interrupt is pending, False if no interrupt is pending

Definition at line 256 of file mrt\_5411x.h.

**6.33.4.15** `__STATIC_INLINE bool Chip_MRT_IsOneShotMode ( LPC_MRT_CH_T * pMRT )`

Check if the timer is configured in one-shot mode.

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

**Returns**

True if in one-shot mode, False if in repeat mode

Definition at line 246 of file mrt\_5411x.h.

**6.33.4.16** `__STATIC_INLINE bool Chip_MRT_IsRepeatMode ( LPC_MRT_CH_T * pMRT )`

Check if the timer is configured in repeat mode.

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

**Returns**

True if in repeat mode, False if in one-shot mode

Definition at line 236 of file mrt\_5411x.h.

**6.33.4.17** `__STATIC_INLINE bool Chip_MRT_Running ( LPC_MRT_CH_T * pMRT )`

Check if the timer is running.

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

**Returns**

True if running, False if stopped

Definition at line 276 of file mrt\_5411x.h.

#### 6.33.4.18 `__STATIC_INLINE void Chip_MRT_SetDisabled ( LPC_MRT_CH_T * pMRT )`

Disables the timer.

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

**Returns**

Nothing

Definition at line 202 of file mrt\_5411x.h.

#### 6.33.4.19 `__STATIC_INLINE void Chip_MRT_SetEnabled ( LPC_MRT_CH_T * pMRT )`

Enables the timer.

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
-------------	-----------------------------------

**Returns**

Nothing

Definition at line 192 of file mrt\_5411x.h.

#### 6.33.4.20 `__STATIC_INLINE void Chip_MRT_SetInterval ( LPC_MRT_CH_T * pMRT, uint32_t interval )`

Sets the timer time interval value.

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
<i>interval</i>	: The interval timeout (31-bits)

**Returns**

Nothing

**Note**

Setting bit 31 in timer time interval register causes the time interval value to load immediately, otherwise the time interval value will be loaded in next timer cycle.

Example: `Chip_MRT_SetInterval(pMRT, 0x500 | MRT_INTVAL_LOAD);` // Will load timer interval immediately

Example: `Chip_MRT_SetInterval(pMRT, 0x500);` // Will load timer interval after interval expires

Definition at line 162 of file mrt\_5411x.h.

6.33.4.21 `__STATIC_INLINE void Chip_MRT_SetMode ( LPC_MRT_CH_T * pMRT, MRT_MODE_T mode )`

Sets the timer mode (repeat or one-shot)

**Parameters**

<i>pMRT</i>	: Pointer to selected MRT Channel
<i>mode</i>	: Timer mode

**Returns**

Nothing

Definition at line 223 of file mrt\_5411x.h.

## 6.34 CHIP: LPC5411X PLL Driver

### 6.34.1 Detailed Description

The PLL in the LPC5411x is flexible, but can be complex to use. This driver provides functions to help setup and use the PLL in its various supported modes.

This driver does not alter PLL clock source or system clocks outside the PLL (like the main clock source) that may be referenced from the PLL. It may optionally setup system voltages, wait for PLL lock, and power cycle the PLL during setup based on setup flags.

The driver works by first generating a PLL setup structure from a desired PLL configuration structure. The PLL setup structure is then passed to the PLL setup function to setup the PLL. In a user application, the PLL setup structure can be pre-populated with PLL setup data to avoid using the PLL configuration structure (or multiple PLL setup structures can be used to more dynamically control PLL output rate).

#### How to use this driver

```
// Setup PLL configuration
PLL_CONFIG_T pllConfig = {
    75000000, // desiredRate = 75MHz
    0,       // InputRate = 0Hz (not used)
    0       // No flags, function will determine best setup to get closest rate
};

// Get closest PLL setup to get the desired configuration
PLL_SETUP_T pllSetup;
uint32_t actualPllRate;
PLL_ERROR_T pllError;
pllError = Chip_Clock_SetupPLLData(&pllConfig, &pllSetup, &actualPllRate);
if (pllError != PLL_ERROR_SUCCESS) {
    printf("PLL setup error %#x\r\n", (uint32_t) pllError);
    while (1);
}
else {
    printf("PLL config successful, actual config rate = %uHz\r\n", actualPllRate);
}

// Make sure main system clock is not using PLL, as the PLL setup
// function will power off and optionally power on the PLL
Chip_Clock_SetMainClockSource(SYSCON_MAINCLKSRC_IRC);

// Setup PLL source
Chip_Clock_SetSystemPLLSource(SYSCON_PLLCLKSRC_IRC);

// Now to apply the configuration to the PLL
pllSetup.flags = PLL_SETUPFLAG_WAITLOCK;
Chip_Clock_SetupSystemPLLPrec(&pllSetup);

// Switch main system clock to PLL
Chip_Clock_SetMainClockSource(SYSCON_MAINCLKSRC_PLLOUT);
```

#### Data Structures

- struct [PLL\\_CONFIG\\_T](#)

*PLL configuration structure This structure can be used to configure the settings for a PLL setup structure. Fill in the desired configuration for the PLL and call the PLL setup function to fill in a PLL setup structure.*

- struct [PLL\\_SETUP\\_T](#)

*PLL setup structure This structure can be used to pre-build a PLL setup configuration at run-time and quickly set the PLL to the configuration. It can be populated with the PLL setup function. If powering up or waiting for PLL lock, the PLL input clock source should be configured prior to PLL setup.*

## Macros

- #define `PLL_CONFIGFLAG_USEINRATE` (1 << 0)  
*PLL configuration structure flags for 'flags' field These flags control how the PLL configuration function sets up the PLL setup structure.*
- #define `PLL_CONFIGFLAG_FORCENOFRACT` (1 << 2)
- #define `PLL_SETUPFLAG_POWERUP` (1 << 0)  
*PLL setup structure flags for 'flags' field These flags control how the PLL setup function sets up the PLL.*
- #define `PLL_SETUPFLAG_WAITLOCK` (1 << 1)
- #define `PLL_SETUPFLAG_ADGVOLT` (1 << 2)

## Enumerations

- enum `CHIP_SYSCON_PLLCLKSRC_T` {  
`SYSCON_PLLCLKSRC_FRO12MHZ` = 0, `SYSCON_PLLCLKSRC_CLKIN`, `SYSCON_PLLCLKSRC_WD`,  
`T`, `SYSCON_PLLCLKSRC_RTC`,  
`SYSCON_PLLCLKSRC_DISABLED` = 7 }
- enum `SS_PROGMODFM_T` {  
`SS_MF_512` = (0 << 20), `SS_MF_384` = (1 << 20), `SS_MF_256` = (2 << 20), `SS_MF_128` = (3 << 20),  
`SS_MF_64` = (4 << 20), `SS_MF_32` = (5 << 20), `SS_MF_24` = (6 << 20), `SS_MF_16` = (7 << 20) }  
*PLL Spread Spectrum (SS) Programmable modulation frequency See (MF) field in the SYSPLLSSCTRL1 register in the UM.*
- enum `SS_PROGMODDP_T` {  
`SS_MR_K0` = (0 << 23), `SS_MR_K1` = (1 << 23), `SS_MR_K1_5` = (2 << 23), `SS_MR_K2` = (3 << 23),  
`SS_MR_K3` = (4 << 23), `SS_MR_K4` = (5 << 23), `SS_MR_K6` = (6 << 23), `SS_MR_K8` = (7 << 23) }  
*PLL Spread Spectrum (SS) Programmable frequency modulation depth See (MR) field in the SYSPLLSSCTRL1 register in the UM.*
- enum `SS_MODWVCTRL_T` { `SS_MC_NOC` = (0 << 26), `SS_MC_RECC` = (2 << 26), `SS_MC_MAXC` = (3 << 26) }  
*PLL Spread Spectrum (SS) Modulation waveform control See (MC) field in the SYSPLLSSCTRL1 register in the UM. Compensation for low pass filtering of the PLL to get a triangular modulation at the output of the PLL, giving a flat frequency spectrum.*
- enum `PLL_ERROR_T` {  
`PLL_ERROR_SUCCESS` = 0, `PLL_ERROR_OUTPUT_TOO_LOW`, `PLL_ERROR_OUTPUT_TOO_HIGH`,  
`PLL_ERROR_INPUT_TOO_LOW`,  
`PLL_ERROR_INPUT_TOO_HIGH`, `PLL_ERROR_OUTSIDE_INTLIMIT` }  
*PLL status definitions.*

## Functions

- \_\_STATIC\_INLINE void `Chip_Clock_SetSystemPLLSource` (`CHIP_SYSCON_PLLCLKSRC_T` src)  
*Set System PLL clock source.*
- uint32\_t `Chip_Clock_GetSystemPLLInClockRate` (void)  
*Return System PLL input clock rate.*
- uint32\_t `Chip_Clock_GetSystemPLLOutClockRate` (bool recompute)  
*Return System PLL output clock rate.*
- void `Chip_Clock_SetBypassPLL` (bool bypass)  
*Enables and disables PLL bypass mode.*
- \_\_STATIC\_INLINE bool `Chip_Clock_IsSystemPLLLocked` (void)  
*Check if PLL is locked or not.*
- uint32\_t `Chip_Clock_GetStoredPLLClockRate` (void)  
*Get the rate of pll from the stored value.*



- void `Chip_Clock_SetStoredPLLClockRate` (uint32\_t rate)  
*Store the current PLL rate.*
- uint32\_t `Chip_Clock_GetSystemPLLOutFromSetup` (PLL\_SETUP\_T \*pSetup)  
*Return System PLL output clock rate from setup structure.*
- PLL\_ERROR\_T `Chip_Clock_SetupPLLData` (PLL\_CONFIG\_T \*pControl, PLL\_SETUP\_T \*pSetup)  
*Set PLL output based on the passed PLL setup data.*
- PLL\_ERROR\_T `Chip_Clock_SetupSystemPLLPrec` (PLL\_SETUP\_T \*pSetup)  
*Set PLL output from PLL setup structure (precise frequency)*
- PLL\_ERROR\_T `Chip_Clock_SetPLLFreq` (const PLL\_SETUP\_T \*pSetup)  
*Set PLL output from PLL setup structure (precise frequency)*
- void `Chip_Clock_SetupSystemPLL` (uint32\_t multiply\_by, uint32\_t input\_freq)  
*Set PLL output based on the multiplier and input frequency.*

## 6.34.2 Macro Definition Documentation

### 6.34.2.1 #define PLL\_CONFIGFLAG\_FORCENOFRACT (1 << 2)

Force non-fractional output mode, PLL output will not use the fractional, automatic bandwidth, or SS hardware  
Definition at line 180 of file pll\_5411x.h.

### 6.34.2.2 #define PLL\_CONFIGFLAG\_USEINRATE (1 << 0)

PLL configuration structure flags for 'flags' field These flags control how the PLL configuration function sets up the PLL setup structure.

When the PLL\_CONFIGFLAG\_USEINRATE flag is selected, the 'InputRate' field in the configuration structure must be assigned with the expected PLL frequency. If the PLL\_CONFIGFLAG\_USEINRATE is not used, 'InputRate' is ignored in the configuration function and the driver will determine the PLL rate from the currently selected PLL source. This flag might be used to configure the PLL input clock more accurately when using the WDT oscillator or a more dynamic CLKIN source.

When the PLL\_CONFIGFLAG\_FORCENOFRACT flag is selected, the PLL hardware for the automatic bandwidth selection, Spread Spectrum (SS) support, and fractional M-divider are not used.

Flag to use InputRate in PLL configuration structure for setup

Definition at line 179 of file pll\_5411x.h.

### 6.34.2.3 #define PLL\_SETUPFLAG\_ADGVOLT (1 << 2)

Optimize system voltage for the new PLL rate

Definition at line 241 of file pll\_5411x.h.

### 6.34.2.4 #define PLL\_SETUPFLAG\_POWERUP (1 << 0)

PLL setup structure flags for 'flags' field These flags control how the PLL setup function sets up the PLL.

Setup will power on the PLL after setup

Definition at line 239 of file pll\_5411x.h.

#### 6.34.2.5 `#define PLL_SETUPFLAG_WAITLOCK (1 << 1)`

Setup will wait for PLL lock, implies the PLL will be powered on

Definition at line 240 of file pll\_5411x.h.

### 6.34.3 Enumeration Type Documentation

#### 6.34.3.1 `enum CHIP_SYSCON_PLLCLKSRC_T`

Clock sources for system PLLs

Enumerator

**`SYSCON_PLLCLKSRC_FRO12MHZ`** 12MHz FRO  
**`SYSCON_PLLCLKSRC_CLKIN`** External clock input pin  
**`SYSCON_PLLCLKSRC_WDT`** Watchdog oscillator  
**`SYSCON_PLLCLKSRC_RTC`** RTC 32KHz oscillator  
**`SYSCON_PLLCLKSRC_DISABLED`** PLL input clock is disabled

Definition at line 100 of file pll\_5411x.h.

#### 6.34.3.2 `enum PLL_ERROR_T`

PLL status definitions.

Enumerator

**`PLL_ERROR_SUCCESS`** PLL operation was successful  
**`PLL_ERROR_OUTPUT_TOO_LOW`** PLL output rate request was too low  
**`PLL_ERROR_OUTPUT_TOO_HIGH`** PLL output rate request was too high  
**`PLL_ERROR_INPUT_TOO_LOW`** PLL input rate is too low  
**`PLL_ERROR_INPUT_TOO_HIGH`** PLL input rate is too high  
**`PLL_ERROR_OUTSIDE_INTLIMIT`** Requested output rate isn't possible

Definition at line 261 of file pll\_5411x.h.

#### 6.34.3.3 `enum SS_MODWVCTRL_T`

PLL Spread Spectrum (SS) Modulation waveform control See (MC) field in the SYSPLLSSCTRL1 register in the UM.

Compensation for low pass filtering of the PLL to get a triangular modulation at the output of the PLL, giving a flat frequency spectrum.

Enumerator

**`SS_MC_NOC`** no compensation  
**`SS_MC_RECC`** recommended setting  
**`SS_MC_MAXC`** max. compensation

Definition at line 215 of file pll\_5411x.h.

6.34.3.4 enum **SS\_PROGMODDP\_T**

PLL Spread Spectrum (SS) Programmable frequency modulation depth See (MR) field in the SYSPLLSSCTRL1 register in the UM.

## Enumerator

**SS\_MR\_K0** k = 0 (no spread spectrum)  
**SS\_MR\_K1** k = 1  
**SS\_MR\_K1\_5** k = 1.5  
**SS\_MR\_K2** k = 2  
**SS\_MR\_K3** k = 3  
**SS\_MR\_K4** k = 4  
**SS\_MR\_K6** k = 6  
**SS\_MR\_K8** k = 8

Definition at line 199 of file pll\_5411x.h.

6.34.3.5 enum **SS\_PROGMODFM\_T**

PLL Spread Spectrum (SS) Programmable modulation frequency See (MF) field in the SYSPLLSSCTRL1 register in the UM.

## Enumerator

**SS\_MF\_512** Nss = 512 (fm 3.9 - 7.8 kHz)  
**SS\_MF\_384** Nss = 384 (fm 5.2 - 10.4 kHz)  
**SS\_MF\_256** Nss = 256 (fm 7.8 - 15.6 kHz)  
**SS\_MF\_128** Nss = 128 (fm 15.6 - 31.3 kHz)  
**SS\_MF\_64** Nss = 64 (fm 32.3 - 64.5 kHz)  
**SS\_MF\_32** Nss = 32 (fm 62.5- 125 kHz)  
**SS\_MF\_24** Nss = 24 (fm 83.3- 166.6 kHz)  
**SS\_MF\_16** Nss = 16 (fm 125- 250 kHz)

Definition at line 185 of file pll\_5411x.h.

## 6.34.4 Function Documentation

## 6.34.4.1 uint32\_t Chip\_Clock\_GetStoredPLLClockRate ( void )

Get the rate of pll from the stored value.

## Returns

Current rate of the PLL from the storage

Definition at line 726 of file pll\_5411x.c.

## 6.34.4.2 uint32\_t Chip\_Clock\_GetSystemPLLInClockRate ( void )

Return System PLL input clock rate.

## Returns

System PLL input clock rate

Definition at line 653 of file pll\_5411x.c.

#### 6.34.4.3 uint32\_t Chip\_Clock\_GetSystemPLLOutClockRate ( bool *recompute* )

Return System PLL output clock rate.

##### Parameters

<i>recompute</i>	: Forces a PLL rate recomputation if true
------------------	---

##### Returns

System PLL output clock rate

##### Note

The PLL rate is cached in the driver in a variable as the rate computation function can take some time to perform. It is recommended to use 'false' with the 'recompute' parameter.

Definition at line 738 of file pll\_5411x.c.

#### 6.34.4.4 uint32\_t Chip\_Clock\_GetSystemPLLOutFromSetup ( PLL\_SETUP\_T \* *pSetup* )

Return System PLL output clock rate from setup structure.

##### Parameters

<i>pSetup</i>	: Pointer to a PLL setup structure
---------------	------------------------------------

##### Returns

System PLL output clock rate the setup structure will generate

Definition at line 683 of file pll\_5411x.c.

#### 6.34.4.5 \_\_STATIC\_INLINE bool Chip\_Clock\_IsSystemPLLLocked ( void )

Check if PLL is locked or not.

##### Returns

true if the PLL is locked, false if not locked

Definition at line 146 of file pll\_5411x.h.

#### 6.34.4.6 void Chip\_Clock\_SetBypassPLL ( bool *bypass* )

Enables and disables PLL bypass mode.

bypass : true to bypass PLL (PLL output = PLL input, false to disable bypass

##### Returns

System PLL output clock rate

Definition at line 759 of file pll\_5411x.c.

#### 6.34.4.7 PLL\_ERROR\_T Chip\_Clock\_SetPLLFreq ( const PLL\_SETUP\_T \* *pSetup* )

Set PLL output from PLL setup structure (precise frequency)

## Parameters

<i>pSetup</i>	: Pointer to populated PLL setup structure
---------------	--

## Returns

PLL\_ERROR\_SUCCESS on success, or PLL setup error code

## Note

This function will power off the PLL, setup the PLL with the new setup data, and then optionally powerup the PLL, wait for PLL lock, and adjust system voltages to the new PLL rate. The function will not alter any source clocks (ie, main system clock) that may use the PLL, so these should be setup prior to and after exiting the function.

Definition at line 835 of file pll\_5411x.c.

#### 6.34.4.8 void Chip\_Clock\_SetStoredPLLClockRate ( uint32\_t rate )

Store the current PLL rate.

## Parameters

<i>rate</i>	Current rate of the PLL
-------------	-------------------------

## Returns

Nothing

Definition at line 732 of file pll\_5411x.c.

#### 6.34.4.9 \_\_STATIC\_INLINE void Chip\_Clock\_SetSystemPLLSource ( CHIP\_SYSCON\_PLLCLKSRC\_T src )

Set System PLL clock source.

## Parameters

<i>src</i>	: Clock source for system PLL
------------	-------------------------------

## Returns

Nothing

## Note

The PLL should be powered down prior to changing the source.

Definition at line 114 of file pll\_5411x.h.

#### 6.34.4.10 PLL\_ERROR\_T Chip\_Clock\_SetupPLLData ( PLL\_CONFIG\_T \* pControl, PLL\_SETUP\_T \* pSetup )

Set PLL output based on the passed PLL setup data.

**Parameters**

<i>pControl</i>	: Pointer to populated PLL control structure to generate setup with
<i>pSetup</i>	: Pointer to PLL setup structure to be filled

**Returns**

PLL\_ERROR\_SUCCESS on success, or PLL setup error code

**Note**

Actual frequency for setup may vary from the desired frequency based on the accuracy of input clocks, rounding, non-fractional PLL mode, etc.

Definition at line 770 of file pll\_5411x.c.

#### 6.34.4.11 void Chip\_Clock\_SetupSystemPLL ( uint32\_t *multiply\_by*, uint32\_t *input\_freq* )

Set PLL output based on the multiplier and input frequency.

**Parameters**

<i>multiply_by</i>	: multiplier
<i>input_freq</i>	: Clock input frequency of the PLL

**Returns**

Nothing

**Note**

Unlike the [Chip\\_Clock\\_SetupSystemPLLPrec\(\)](#) function, this function does not disable or enable PLL power, wait for PLL lock, or adjust system voltages. These must be done in the application. The function will not alter any source clocks (ie, main system clock) that may use the PLL, so these should be setup prior to and after exiting the function.

Definition at line 866 of file pll\_5411x.c.

#### 6.34.4.12 PLL\_ERROR\_T Chip\_Clock\_SetupSystemPLLPrec ( PLL\_SETUP\_T \* *pSetup* )

Set PLL output from PLL setup structure (precise frequency)

**Parameters**

<i>pSetup</i>	: Pointer to populated PLL setup structure
---------------	--

**Returns**

PLL\_ERROR\_SUCCESS on success, or PLL setup error code

**Note**

This function will power off the PLL, setup the PLL with the new setup data, and then optionally powerup the PLL, wait for PLL lock, and adjust system voltages to the new PLL rate. The function will not alter any source clocks (ie, main system clock) that may use the PLL, so these should be setup prior to and after exiting the function.

Definition at line 799 of file pll\_5411x.c.

## 6.35 CHIP: LPC5411X Peripheral addresses and register set declarations

### 6.35.1 Detailed Description

#### Macros

- #define `LPC_FLASHMEM_BASE` 0x00000000UL
- #define `LPC_SRAMX_BASE` 0x04000000UL
- #define `LPC_SRAM0_BASE` 0x20000000UL
- #define `LPC_SRAM1_BASE` 0x20010000UL
- #define `LPC_SRAM2_BASE` 0x20018000UL
- #define `LPC_ROM_BASE` 0x03000000UL
- #define `LPC_SYSCON_BASE` 0x40000000UL
- #define `LPC_IOCON_BASE` 0x40001000UL
- #define `LPC_GPIO_GROUPINT0_BASE` 0x40002000UL
- #define `LPC_GPIO_GROUPINT1_BASE` 0x40003000UL
- #define `LPC_PIN_INT_BASE` 0x40004000UL
- #define `LPC_INMUX_BASE` 0x40005000UL
- #define `LPC_TIMER0_BASE` 0x40008000UL
- #define `LPC_TIMER1_BASE` 0x40009000UL
- #define `LPC_WWDT_BASE` 0x4000C000UL
- #define `LPC_MRT_BASE` 0x4000D000UL
- #define `LPC_UTICK_BASE` 0x4000E000UL
- #define `LPC_PMU_BASE` 0x40020000UL
- #define `LPC_TIMER2_BASE` 0x40028000UL
- #define `LPC_RTC_BASE` 0x4002C000UL
- #define `LPC_FMC_BASE` 0x40034000UL
- #define `LPC_ASYNC_SYSCON_BASE` 0x40040000UL
- #define `LPC_TIMER3_BASE` 0x40048000UL
- #define `LPC_TIMER4_BASE` 0x40049000UL
- #define `LPC_SPIFI_BASE` 0x40080000UL
- #define `LPC_DMA_BASE` 0x40082000UL
- #define `LPC_USB_BASE` 0x40084000UL
- #define `LPC_SCT_BASE` 0x40085000UL
- #define `LPC_FLEXCOMM0_BASE` 0x40086000UL
- #define `LPC_FLEXCOMM1_BASE` 0x40087000UL
- #define `LPC_FLEXCOMM2_BASE` 0x40088000UL
- #define `LPC_FLEXCOMM3_BASE` 0x40089000UL
- #define `LPC_FLEXCOMM4_BASE` 0x4008A000UL
- #define `LPC_MBOX_BASE` 0x4008B000UL
- #define `LPC_GPIO_PORT_BASE` 0x4008C000UL
- #define `LPC_DMIC_BASE` 0x40090000UL
- #define `LPC_CRC_BASE` 0x40095000UL
- #define `LPC_FLEXCOMM5_BASE` 0x40096000UL
- #define `LPC_FLEXCOMM6_BASE` 0x40097000UL
- #define `LPC_FLEXCOMM7_BASE` 0x40098000UL
- #define `LPC_ISPAP_BASE` 0x4009C000UL
- #define `LPC_ADC_BASE` 0x400A0000UL
- #define `LPC_GPIO` ((LPC\_GPIO\_T \*) `LPC_GPIO_PORT_BASE`)
- #define `LPC_DMA` ((LPC\_DMA\_T \*) `LPC_DMA_BASE`)
- #define `LPC_CRC` ((LPC\_CRC\_T \*) `LPC_CRC_BASE`)
- #define `LPC_SCT` ((LPC\_SCT\_T \*) `LPC_SCT_BASE`)
- #define `LPC_MBOX` ((LPC\_MBOX\_T \*) `LPC_MBOX_BASE`)
- #define `LPC_ADC` ((LPC\_ADC\_T \*) `LPC_ADC_BASE`)

- `#define LPC_PMU ((LPC_PMU_T *) LPC_PMU_BASE)`
- `#define LPC_DMIC ((LPC_DMIC_T *) LPC_DMIC_BASE)`
- `#define LPC_USB ((LPC_USB_T *) LPC_USB_BASE)`
- `#define LPC_SYSCON ((LPC_SYSCON_T *) LPC_SYSCON_BASE)`
- `#define LPC_TIMER2 ((LPC_TIMER_T *) LPC_TIMER2_BASE)`
- `#define LPC_TIMER3 ((LPC_TIMER_T *) LPC_TIMER3_BASE)`
- `#define LPC_TIMER4 ((LPC_TIMER_T *) LPC_TIMER4_BASE)`
- `#define LPC_GINT ((LPC_GPIOGROUPINT_T *) LPC_GPIO_GROUPINT0_BASE)`
- `#define LPC_PININT ((LPC_PIN_INT_T *) LPC_PIN_INT_BASE)`
- `#define LPC_IOCON ((LPC_IOCON_T *) LPC_IOCON_BASE)`
- `#define LPC_UTICK ((LPC_UTICK_T *) LPC_UTICK_BASE)`
- `#define LPC_WWDT ((LPC_WWDT_T *) LPC_WWDT_BASE)`
- `#define LPC_RTC ((LPC_RTC_T *) LPC_RTC_BASE)`
- `#define LPC_ASYNC_SYSCON ((LPC_ASYNC_SYSCON_T *) LPC_ASYNC_SYSCON_BASE)`
- `#define LPC_TIMER0 ((LPC_TIMER_T *) LPC_TIMER0_BASE)`
- `#define LPC_TIMER1 ((LPC_TIMER_T *) LPC_TIMER1_BASE)`
- `#define LPC_INMUX ((LPC_INMUX_T *) LPC_INMUX_BASE)`
- `#define LPC_MRT ((LPC_MRT_T *) LPC_MRT_BASE)`

## 6.35.2 Macro Definition Documentation

### 6.35.2.1 `#define LPC_ADC ((LPC_ADC_T *) LPC_ADC_BASE)`

Definition at line 130 of file chip.h.

### 6.35.2.2 `#define LPC_ADC_BASE 0x400A0000UL`

Definition at line 120 of file chip.h.

### 6.35.2.3 `#define LPC_ASYNC_SYSCON ((LPC_ASYNC_SYSCON_T *) LPC_ASYNC_SYSCON_BASE)`

Definition at line 148 of file chip.h.

### 6.35.2.4 `#define LPC_ASYNC_SYSCON_BASE 0x40040000UL`

Definition at line 98 of file chip.h.

### 6.35.2.5 `#define LPC_CRC ((LPC_CRC_T *) LPC_CRC_BASE)`

Definition at line 127 of file chip.h.

### 6.35.2.6 `#define LPC_CRC_BASE 0x40095000UL`

Definition at line 115 of file chip.h.

### 6.35.2.7 `#define LPC_DMA ((LPC_DMA_T *) LPC_DMA_BASE)`

Definition at line 126 of file chip.h.



6.35.2.8 `#define LPC_DMA_BASE 0x40082000UL`

Definition at line 104 of file chip.h.

6.35.2.9 `#define LPC_DMIC ((LPC_DMIC_T *) LPC_DMIC_BASE)`

Definition at line 132 of file chip.h.

6.35.2.10 `#define LPC_DMIC_BASE 0x40090000UL`

Definition at line 114 of file chip.h.

6.35.2.11 `#define LPC_FLASHMEM_BASE 0x00000000UL`

Definition at line 71 of file chip.h.

6.35.2.12 `#define LPC_FLEXCOMM0_BASE 0x40086000UL`

Definition at line 107 of file chip.h.

6.35.2.13 `#define LPC_FLEXCOMM1_BASE 0x40087000UL`

Definition at line 108 of file chip.h.

6.35.2.14 `#define LPC_FLEXCOMM2_BASE 0x40088000UL`

Definition at line 109 of file chip.h.

6.35.2.15 `#define LPC_FLEXCOMM3_BASE 0x40089000UL`

Definition at line 110 of file chip.h.

6.35.2.16 `#define LPC_FLEXCOMM4_BASE 0x4008A000UL`

Definition at line 111 of file chip.h.

6.35.2.17 `#define LPC_FLEXCOMM5_BASE 0x40096000UL`

Definition at line 116 of file chip.h.

6.35.2.18 `#define LPC_FLEXCOMM6_BASE 0x40097000UL`

Definition at line 117 of file chip.h.

6.35.2.19 `#define LPC_FLEXCOMM7_BASE 0x40098000UL`

Definition at line 118 of file chip.h.

6.35.2.20 `#define LPC_FMC_BASE 0x40034000UL`

Definition at line 95 of file chip.h.

6.35.2.21 `#define LPC_GINT ((LPC_GPIOGROUPINT_T *) LPC_GPIO_GROUPINT0_BASE)`

Definition at line 140 of file chip.h.

6.35.2.22 `#define LPC_GPIO ((LPC_GPIO_T *) LPC_GPIO_PORT_BASE)`

Definition at line 125 of file chip.h.

6.35.2.23 `#define LPC_GPIO_GROUPINT0_BASE 0x40002000UL`

Definition at line 81 of file chip.h.

6.35.2.24 `#define LPC_GPIO_GROUPINT1_BASE 0x40003000UL`

Definition at line 82 of file chip.h.

6.35.2.25 `#define LPC_GPIO_PORT_BASE 0x4008C000UL`

Definition at line 113 of file chip.h.

6.35.2.26 `#define LPC_INMUX ((LPC_INMUX_T *) LPC_INMUX_BASE)`

Definition at line 151 of file chip.h.

6.35.2.27 `#define LPC_INMUX_BASE 0x40005000UL`

Definition at line 84 of file chip.h.

6.35.2.28 `#define LPC_IOCON ((LPC_IOCON_T *) LPC_IOCON_BASE)`

Definition at line 142 of file chip.h.

6.35.2.29 `#define LPC_IOCON_BASE 0x40001000UL`

Definition at line 80 of file chip.h.

6.35.2.30 `#define LPC_ISPAP_BASE 0x4009C000UL`

Definition at line 119 of file chip.h.

6.35.2.31 `#define LPC_MBOX ((LPC_MBOX_T *) LPC_MBOX_BASE)`

Definition at line 129 of file chip.h.

6.35.2.32 `#define LPC_MBOX_BASE 0x4008B000UL`

Definition at line 112 of file chip.h.

6.35.2.33 `#define LPC_MRT ((LPC_MRT_T *) LPC_MRT_BASE)`

Definition at line 152 of file chip.h.

6.35.2.34 `#define LPC_MRT_BASE 0x4000D000UL`

Definition at line 88 of file chip.h.

6.35.2.35 `#define LPC_PIN_INT_BASE 0x40004000UL`

Definition at line 83 of file chip.h.

6.35.2.36 `#define LPC_PININT ((LPC_PIN_INT_T *) LPC_PIN_INT_BASE)`

Definition at line 141 of file chip.h.

6.35.2.37 `#define LPC_PMU ((LPC_PMU_T *) LPC_PMU_BASE)`

Definition at line 131 of file chip.h.

6.35.2.38 `#define LPC_PMU_BASE 0x40020000UL`

Definition at line 92 of file chip.h.

6.35.2.39 `#define LPC_ROM_BASE 0x03000000UL`

Definition at line 76 of file chip.h.

6.35.2.40 `#define LPC_RTC ((LPC_RTC_T *) LPC_RTC_BASE)`

Definition at line 145 of file chip.h.

6.35.2.41 `#define LPC_RTC_BASE 0x4002C000UL`

Definition at line 94 of file chip.h.

6.35.2.42 `#define LPC_SCT ((LPC_SCT_T *) LPC_SCT_BASE)`

Definition at line 128 of file chip.h.

6.35.2.43 `#define LPC_SCT_BASE 0x40085000UL`

Definition at line 106 of file chip.h.

6.35.2.44 `#define LPC_SPIFI_BASE 0x40080000UL`

Definition at line 103 of file chip.h.

6.35.2.45 `#define LPC_SRAM0_BASE 0x20000000UL`

Definition at line 73 of file chip.h.

6.35.2.46 `#define LPC_SRAM1_BASE 0x20010000UL`

Definition at line 74 of file chip.h.

6.35.2.47 `#define LPC_SRAM2_BASE 0x20018000UL`

Definition at line 75 of file chip.h.

6.35.2.48 `#define LPC_SRAMX_BASE 0x04000000UL`

Definition at line 72 of file chip.h.

6.35.2.49 `#define LPC_SYSCON ((LPC_SYSCON_T *) LPC_SYSCON_BASE)`

Definition at line 136 of file chip.h.

6.35.2.50 `#define LPC_SYSCON_BASE 0x40000000UL`

Definition at line 79 of file chip.h.

6.35.2.51 `#define LPC_TIMER0 ((LPC_TIMER_T *) LPC_TIMER0_BASE)`

Definition at line 149 of file chip.h.

6.35.2.52 `#define LPC_TIMER0_BASE 0x40008000UL`

Definition at line 85 of file chip.h.

6.35.2.53 `#define LPC_TIMER1 ((LPC_TIMER_T *) LPC_TIMER1_BASE)`

Definition at line 150 of file chip.h.

6.35.2.54 `#define LPC_TIMER1_BASE 0x40009000UL`

Definition at line 86 of file chip.h.

6.35.2.55 `#define LPC_TIMER2 ((LPC_TIMER_T *) LPC_TIMER2_BASE)`

Definition at line 137 of file chip.h.

6.35.2.56 `#define LPC_TIMER2_BASE 0x40028000UL`

Definition at line 93 of file chip.h.

6.35.2.57 `#define LPC_TIMER3 ((LPC_TIMER_T *) LPC_TIMER3_BASE)`

Definition at line 138 of file chip.h.

6.35.2.58 `#define LPC_TIMER3_BASE 0x40048000UL`

Definition at line 99 of file chip.h.

6.35.2.59 `#define LPC_TIMER4 ((LPC_TIMER_T *) LPC_TIMER4_BASE)`

Definition at line 139 of file chip.h.

6.35.2.60 `#define LPC_TIMER4_BASE 0x40049000UL`

Definition at line 100 of file chip.h.

6.35.2.61 `#define LPC_USB ((LPC_USB_T *) LPC_USB_BASE)`

Definition at line 133 of file chip.h.

6.35.2.62 `#define LPC_USB_BASE 0x40084000UL`

Definition at line 105 of file chip.h.

6.35.2.63 `#define LPC_UTICK ((LPC_UTICK_T *) LPC_UTICK_BASE)`

Definition at line 143 of file chip.h.

6.35.2.64 `#define LPC_UTICK_BASE 0x4000E000UL`

Definition at line 89 of file chip.h.

6.35.2.65 `#define LPC_WWDT ((LPC_WWDT_T *) LPC_WWDT_BASE)`

Definition at line 144 of file chip.h.

6.35.2.66 `#define LPC_WWDT_BASE 0x4000C000UL`

Definition at line 87 of file chip.h.

## 6.36 CHIP: LPC5411X Pin Interrupt and Pattern Match driver

### 6.36.1 Detailed Description

#### Data Structures

- struct [LPC\\_PIN\\_INT\\_T](#)  
*LPC5411X Pin Interrupt and Pattern Match register block structure.*

#### Macros

- #define [PININT\\_ISEL\\_PMODE\\_MASK](#) ((uint32\_t) 0x00FF)
- #define [PININT\\_PMCTRL\\_MASK](#) ((uint32\_t) 0xFF000003)
- #define [PININT\\_PMCTRL\\_PMATCH\\_SEL](#) (1 << 0)
- #define [PININT\\_PMCTRL\\_RXEV\\_ENA](#) (1 << 1)
- #define [PININT\\_SRC\\_BITSOURCE\\_START](#) 8
- #define [PININT\\_SRC\\_BITSOURCE\\_MASK](#) 7
- #define [PININT\\_SRC\\_BITCFG\\_START](#) 8
- #define [PININT\\_SRC\\_BITCFG\\_MASK](#) 7
- #define [PININTCH0](#) (1 << 0)
- #define [PININTCH1](#) (1 << 1)
- #define [PININTCH2](#) (1 << 2)
- #define [PININTCH3](#) (1 << 3)
- #define [PININTCH4](#) (1 << 4)
- #define [PININTCH5](#) (1 << 5)
- #define [PININTCH6](#) (1 << 6)
- #define [PININTCH7](#) (1 << 7)
- #define [PININTCH](#)(ch) (1 << (ch))

#### Enumerations

- enum [Chip\\_PININT\\_SELECT\\_T](#) {  
  [PININTSELECT0](#) = 0, [PININTSELECT1](#) = 1, [PININTSELECT2](#) = 2, [PININTSELECT3](#) = 3,  
  [PININTSELECT4](#) = 4, [PININTSELECT5](#) = 5, [PININTSELECT6](#) = 6, [PININTSELECT7](#) = 7 }
- enum [Chip\\_PININT\\_BITSLICE\\_T](#) {  
  [PININTBITSLICE0](#) = 0, [PININTBITSLICE1](#) = 1, [PININTBITSLICE2](#) = 2, [PININTBITSLICE3](#) = 3,  
  [PININTBITSLICE4](#) = 4, [PININTBITSLICE5](#) = 5, [PININTBITSLICE6](#) = 6, [PININTBITSLICE7](#) = 7 }
- enum [Chip\\_PININT\\_BITSLICE\\_CFG\\_T](#) {  
  [PININT\\_PATTERNCONST1](#) = 0x0, [PININT\\_PATTERNRISING](#) = 0x1, [PININT\\_PATTERNFALLING](#) = 0x2,  
  [PININT\\_PATTERNRISINGORFALLING](#) = 0x3,  
  [PININT\\_PATTERNHIGH](#) = 0x4, [PININT\\_PATTERNLOW](#) = 0x5, [PININT\\_PATTERNCONST0](#) = 0x6, [PININT\\_PATTERNEVENT](#) = 0x7 }

#### Functions

- `__STATIC_INLINE void Chip\_PININT\_Init (LPC\_PIN\_INT\_T *pPININT)`  
*Initialize Pin interrupt block.*
- `__STATIC_INLINE void Chip\_PININT\_DeInit (LPC\_PIN\_INT\_T *pPININT)`  
*De-Initialize Pin interrupt block.*
- `__STATIC_INLINE void Chip\_PININT\_SetPinModeEdge (LPC\_PIN\_INT\_T *pPININT, uint32_t pins)`  
*Configure the pins as edge sensitive in Pin interrupt block.*
- `__STATIC_INLINE void Chip\_PININT\_SetPinModeLevel (LPC\_PIN\_INT\_T *pPININT, uint32_t pins)`

- Configure the pins as level sensitive in Pin interrupt block.*
- `__STATIC_INLINE uint32_t Chip_PININT_GetPinMode (LPC_PIN_INT_T *pPININT)`  
*Return current PININT edge or level sensitive interrupt selection state.*
  - `__STATIC_INLINE uint32_t Chip_PININT_GetHighEnabled (LPC_PIN_INT_T *pPININT)`  
*Return current PININT rising edge or level interrupt enable state.*
  - `__STATIC_INLINE void Chip_PININT_EnableIntHigh (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Enable rising edge/level PININT interrupts for pins.*
  - `__STATIC_INLINE void Chip_PININT_DisableIntHigh (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Disable rising edge/level PININT interrupts for pins.*
  - `__STATIC_INLINE uint32_t Chip_PININT_GetLowEnabled (LPC_PIN_INT_T *pPININT)`  
*Return current PININT falling edge or level interrupt active level enable state.*
  - `__STATIC_INLINE void Chip_PININT_EnableIntLow (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Enable falling edge/level active level PININT interrupts for pins.*
  - `__STATIC_INLINE void Chip_PININT_DisableIntLow (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Disable low edge/level active level PININT interrupts for pins.*
  - `__STATIC_INLINE uint32_t Chip_PININT_GetRiseStates (LPC_PIN_INT_T *pPININT)`  
*Return pin states that have a detected latched rising edge (RISE) state.*
  - `__STATIC_INLINE void Chip_PININT_ClearRiseStates (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Clears pin states that had a latched rising edge (RISE) state.*
  - `__STATIC_INLINE uint32_t Chip_PININT_GetFallStates (LPC_PIN_INT_T *pPININT)`  
*Return pin states that have a detected latched falling edge (FALL) state.*
  - `__STATIC_INLINE void Chip_PININT_ClearFallStates (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Clears pin states that had a latched falling edge (FALL) state.*
  - `__STATIC_INLINE uint32_t Chip_PININT_GetIntStatus (LPC_PIN_INT_T *pPININT)`  
*Get interrupt status from Pin interrupt block.*
  - `__STATIC_INLINE void Chip_PININT_ClearIntStatus (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Clear interrupt status in Pin interrupt block.*
  - `__STATIC_INLINE void Chip_PININT_SetPatternMatchSrc (LPC_PIN_INT_T *pPININT, Chip_PININT_SELECT_T channelNum, Chip_PININT_BITSLICE_T sliceNum)`  
*Set source for pattern match in Pin interrupt block.*
  - `__STATIC_INLINE void Chip_PININT_SetPatternMatchConfig (LPC_PIN_INT_T *pPININT, Chip_PININT_BITSLICE_T sliceNum, Chip_PININT_BITSLICE_CFG_T slice_cfg, bool end_point)`  
*Configure the pattern match in Pin interrupt block.*
  - `__STATIC_INLINE void Chip_PININT_EnablePatternMatch (LPC_PIN_INT_T *pPININT)`  
*Enable pattern match interrupts in Pin interrupt block.*
  - `__STATIC_INLINE void Chip_PININT_DisablePatternMatch (LPC_PIN_INT_T *pPININT)`  
*Disable pattern match interrupts in Pin interrupt block.*
  - `__STATIC_INLINE void Chip_PININT_EnablePatternMatchRxEv (LPC_PIN_INT_T *pPININT)`  
*Enable RXEV output in Pin interrupt block.*
  - `__STATIC_INLINE void Chip_PININT_DisablePatternMatchRxEv (LPC_PIN_INT_T *pPININT)`  
*Disable RXEV output in Pin interrupt block.*
  - `__STATIC_INLINE uint32_t Chip_PININT_GetPatternMatchState (LPC_PIN_INT_T *pPININT)`  
*Return pattern match state.*

## 6.36.2 Macro Definition Documentation

### 6.36.2.1 `#define PININT_ISEL_PMODE_MASK ((uint32_t) 0x00FF)`

LPC5411X Pin Interrupt and Pattern match engine register bit fields and macros

Definition at line 69 of file `pinint_5411x.h`.

**6.36.2.2   #define PININT\_PMCTRL\_MASK ((uint32\_t) 0xFF000003)**

Definition at line 72 of file pinint\_5411x.h.

**6.36.2.3   #define PININT\_PMCTRL\_PMATCH\_SEL (1 << 0)**

Definition at line 75 of file pinint\_5411x.h.

**6.36.2.4   #define PININT\_PMCTRL\_RXEV\_ENA (1 << 1)**

Definition at line 76 of file pinint\_5411x.h.

**6.36.2.5   #define PININT\_SRC\_BITCFG\_MASK 7**

Definition at line 84 of file pinint\_5411x.h.

**6.36.2.6   #define PININT\_SRC\_BITCFG\_START 8**

Definition at line 83 of file pinint\_5411x.h.

**6.36.2.7   #define PININT\_SRC\_BITSOURCE\_MASK 7**

Definition at line 80 of file pinint\_5411x.h.

**6.36.2.8   #define PININT\_SRC\_BITSOURCE\_START 8**

Definition at line 79 of file pinint\_5411x.h.

**6.36.2.9   #define PININTCH( ch ) (1 << (ch))**

Definition at line 97 of file pinint\_5411x.h.

**6.36.2.10   #define PININTCH0 (1 << 0)**

LPC5411X Pin Interrupt channel values

Definition at line 89 of file pinint\_5411x.h.

**6.36.2.11   #define PININTCH1 (1 << 1)**

Definition at line 90 of file pinint\_5411x.h.

**6.36.2.12   #define PININTCH2 (1 << 2)**

Definition at line 91 of file pinint\_5411x.h.

**6.36.2.13   #define PININTCH3 (1 << 3)**

Definition at line 92 of file pinint\_5411x.h.



6.36.2.14 `#define PININTCH4 (1 << 4)`

Definition at line 93 of file `pinint_5411x.h`.

6.36.2.15 `#define PININTCH5 (1 << 5)`

Definition at line 94 of file `pinint_5411x.h`.

6.36.2.16 `#define PININTCH6 (1 << 6)`

Definition at line 95 of file `pinint_5411x.h`.

6.36.2.17 `#define PININTCH7 (1 << 7)`

Definition at line 96 of file `pinint_5411x.h`.

### 6.36.3 Enumeration Type Documentation

6.36.3.1 `enum Chip_PININT_BITSLICE_CFG_T`

LPC5411X Pin Matching Interrupt bit slice configuration enum values

Enumerator

***PININT\_PATTERNCONST1*** Contributes to product term match  
***PININT\_PATTERNRISING*** Rising edge  
***PININT\_PATTERNFALLING*** Falling edge  
***PININT\_PATTERNRISINGORFALLING*** Rising or Falling edge  
***PININT\_PATTERNHIGH*** High level  
***PININT\_PATTERNLOW*** Low level  
***PININT\_PATTERNCONST0*** Never contributes for match  
***PININT\_PATTERNEVENT*** Match occurs on event

Definition at line 130 of file `pinint_5411x.h`.

6.36.3.2 `enum Chip_PININT_BITSLICE_T`

LPC5411X Pin Matching Interrupt bit slice enum values

Enumerator

***PININTBITSLICE0*** PININT Bit slice 0  
***PININTBITSLICE1*** PININT Bit slice 1  
***PININTBITSLICE2*** PININT Bit slice 2  
***PININTBITSLICE3*** PININT Bit slice 3  
***PININTBITSLICE4*** PININT Bit slice 4  
***PININTBITSLICE5*** PININT Bit slice 5  
***PININTBITSLICE6*** PININT Bit slice 6  
***PININTBITSLICE7*** PININT Bit slice 7

Definition at line 116 of file `pinint_5411x.h`.

### 6.36.3.3 enum Chip\_PININT\_SELECT\_T

LPC5411X Pin Interrupt select enum values

Enumerator

***PININTSELECT0***  
***PININTSELECT1***  
***PININTSELECT2***  
***PININTSELECT3***  
***PININTSELECT4***  
***PININTSELECT5***  
***PININTSELECT6***  
***PININTSELECT7***

Definition at line 102 of file pinint\_5411x.h.

## 6.36.4 Function Documentation

### 6.36.4.1 \_\_STATIC\_INLINE void Chip\_PININT\_ClearFallStates ( LPC\_PIN\_INT\_T \* *pPININT*, uint32\_t *pins* )

Clears pin states that had a latched falling edge (FALL) state.

Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
<i>pins</i>	: Pins with latched states to clear

Returns

Nothing

Definition at line 305 of file pinint\_5411x.h.

### 6.36.4.2 \_\_STATIC\_INLINE void Chip\_PININT\_ClearIntStatus ( LPC\_PIN\_INT\_T \* *pPININT*, uint32\_t *pins* )

Clear interrupt status in Pin interrupt block.

Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
<i>pins</i>	: Pin interrupts to clear (ORed value of PININTCH*)

Returns

Nothing

Definition at line 326 of file pinint\_5411x.h.

### 6.36.4.3 \_\_STATIC\_INLINE void Chip\_PININT\_ClearRiseStates ( LPC\_PIN\_INT\_T \* *pPININT*, uint32\_t *pins* )

Clears pin states that had a latched rising edge (RISE) state.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
<i>pins</i>	: Pins with latched states to clear

## Returns

Nothing

Definition at line 284 of file pinint\_5411x.h.

#### 6.36.4.4 `__STATIC_INLINE void Chip_PININT_DeInit ( LPC_PIN_INT_T * pPININT )`

De-Initialize Pin interrupt block.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

Nothing

Definition at line 158 of file pinint\_5411x.h.

#### 6.36.4.5 `__STATIC_INLINE void Chip_PININT_DisableIntHigh ( LPC_PIN_INT_T * pPININT, uint32_t pins )`

Disable rising edge/level PININT interrupts for pins.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
<i>pins</i>	: Pins to disable (ORed value of PININTCH*)

## Returns

Nothing

Definition at line 228 of file pinint\_5411x.h.

#### 6.36.4.6 `__STATIC_INLINE void Chip_PININT_DisableIntLow ( LPC_PIN_INT_T * pPININT, uint32_t pins )`

Disable low edge/level active level PININT interrupts for pins.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
<i>pins</i>	: Pins to disable (ORed value of PININTCH*)

## Returns

Nothing

Definition at line 263 of file pinint\_5411x.h.

#### 6.36.4.7 `__STATIC_INLINE void Chip_PININT_DisablePatternMatch ( LPC_PIN_INT_T * pPININT )`

Disable pattern match interrupts in Pin interrupt block.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

Nothing

Definition at line 395 of file pinint\_5411x.h.

#### 6.36.4.8 `__STATIC_INLINE void Chip_PININT_DisablePatternMatchRxEv ( LPC_PIN_INT_T * pPININT )`

Disable RXEV output in Pin interrupt block.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

Nothing

Definition at line 415 of file pinint\_5411x.h.

#### 6.36.4.9 `__STATIC_INLINE void Chip_PININT_EnableIntHigh ( LPC_PIN_INT_T * pPININT, uint32_t pins )`

Enable rising edge/level PININT interrupts for pins.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
<i>pins</i>	: Pins to enable (ORed value of PININTCH*)

## Returns

Nothing

Definition at line 217 of file pinint\_5411x.h.

#### 6.36.4.10 `__STATIC_INLINE void Chip_PININT_EnableIntLow ( LPC_PIN_INT_T * pPININT, uint32_t pins )`

Enable falling edge/level active level PININT interrupts for pins.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
<i>pins</i>	: Pins to enable (ORed value of PININTCH*)

## Returns

Nothing

Definition at line 252 of file pinint\_5411x.h.

#### 6.36.4.11 `__STATIC_INLINE void Chip_PININT_EnablePatternMatch ( LPC_PIN_INT_T * pPININT )`

Enable pattern match interrupts in Pin interrupt block.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

Nothing

Definition at line 385 of file pinint\_5411x.h.

#### 6.36.4.12 `__STATIC_INLINE void Chip_PININT_EnablePatternMatchRxEv ( LPC_PIN_INT_T * pPININT )`

Enable RXEV output in Pin interrupt block.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

Nothing

Definition at line 405 of file pinint\_5411x.h.

#### 6.36.4.13 `__STATIC_INLINE uint32_t Chip_PININT_GetFallStates ( LPC_PIN_INT_T * pPININT )`

Return pin states that have a detected latched falling edge (FALL) state.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

PININT states (bit n = high) with a latched rise state detected

Definition at line 294 of file pinint\_5411x.h.

#### 6.36.4.14 `__STATIC_INLINE uint32_t Chip_PININT_GetHighEnabled ( LPC_PIN_INT_T * pPININT )`

Return current PININT rising edge or level interrupt enable state.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

A bifield containing the rising edge/level enable for each interrupt. Bit 0 = PININT0, 1 = PININT1, etc. For each bit, a 0 means the rising edge/level interrupt is disabled, while a 1 means it's enabled.

Definition at line 206 of file pinint\_5411x.h.

#### 6.36.4.15 `__STATIC_INLINE uint32_t Chip_PININT_GetIntStatus ( LPC_PIN_INT_T * pPININT )`

Get interrupt status from Pin interrupt block.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

Interrupt status (bit n for PININTn = high means interrupt ie pending)

Definition at line 315 of file pinint\_5411x.h.

#### 6.36.4.16 `__STATIC_INLINE uint32_t Chip_PININT_GetLowEnabled ( LPC_PIN_INT_T * pPININT )`

Return current PININT falling edge or level interrupt active level enable state.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

A bifield containing the falling edge/level interrupt active level enable for each interrupt. Bit 0 = PININT0, 1 = PININT1, etc. For each bit, a 0 means the falling edge is disabled/level interrupt active low is enabled, while a 1 means the falling edge is enabled/level interrupt active high is enabled.

Definition at line 241 of file pinint\_5411x.h.

#### 6.36.4.17 `__STATIC_INLINE uint32_t Chip_PININT_GetPatternMatchState ( LPC_PIN_INT_T * pPININT )`

Return pattern match state.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

8 bit pattern match state, where a 1 in any bit indicates that the corresponding product term has matched by the current state of its inputs.

Definition at line 427 of file pinint\_5411x.h.

#### 6.36.4.18 `__STATIC_INLINE uint32_t Chip_PININT_GetPinMode ( LPC_PIN_INT_T * pPININT )`

Return current PININT edge or level sensitive interrupt selection state.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

A bifield containing the edge/level sensitive selection for each interrupt. Bit 0 = PININT0, 1 = PININT1, etc. For each bit, a 0 means the edge sensitive interrupt is selected, while a 1 means the level sensitive interrupt is selected.

Definition at line 193 of file pinint\_5411x.h.

#### 6.36.4.19 `__STATIC_INLINE uint32_t Chip_PININT_GetRiseStates ( LPC_PIN_INT_T * pPININT )`

Return pin states that have a detected latched rising edge (RISE) state.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

PININT states (bit n = high) with a latched rise state detected

Definition at line 273 of file pinint\_5411x.h.

#### 6.36.4.20 `__STATIC_INLINE void Chip_PININT_Init ( LPC_PIN_INT_T * pPININT )`

Initialize Pin interrupt block.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
----------------	---

## Returns

Nothing

## Note

This function should be used after the [Chip\\_GPIO\\_Init\(\)](#) function.

Definition at line 147 of file pinint\_5411x.h.

#### 6.36.4.21 `__STATIC_INLINE void Chip_PININT_SetPatternMatchConfig ( LPC_PIN_INT_T * pPININT, Chip_PININT_BITSLICE_T sliceNum, Chip_PININT_BITSLICE_CFG_T slice_cfg, bool end_point )`

Configure the pattern match in Pin interrupt block.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
<i>sliceNum</i>	: PININT slice number
<i>slice_cfg</i>	: PININT slice configuration value (enum <code>Chip_PININT_BITSLICE_CFG_T</code> )
<i>end_point</i>	: If true, current slice is final component

## Returns

Nothing

Definition at line 357 of file pinint\_5411x.h.

#### 6.36.4.22 `__STATIC_INLINE void Chip_PININT_SetPatternMatchSrc ( LPC_PIN_INT_T * pPININT, Chip_PININT_SELECT_T channelNum, Chip_PININT_BITSLICE_T sliceNum )`

Set source for pattern match in Pin interrupt block.

## Parameters

<i>pPININT</i>	: The base address of Pin interrupt block
<i>channelNum</i>	: PININT channel number (From 0 to 7)
<i>sliceNum</i>	: PININT slice number

**Returns**

Nothing

Definition at line 338 of file pinint\_5411x.h.

6.36.4.23 `__STATIC_INLINE void Chip_PININT_SetPinModeEdge ( LPC_PIN_INT_T * pPININT, uint32_t pins )`

Configure the pins as edge sensitive in Pin interrupt block.

**Parameters**

<i>pPININT</i>	: The base address of Pin interrupt block
<i>pins</i>	: Pins (ORed value of PININTCH*)

**Returns**

Nothing

Definition at line 169 of file pinint\_5411x.h.

6.36.4.24 `__STATIC_INLINE void Chip_PININT_SetPinModeLevel ( LPC_PIN_INT_T * pPININT, uint32_t pins )`

Configure the pins as level sensitive in Pin interrupt block.

**Parameters**

<i>pPININT</i>	: The base address of Pin interrupt block
<i>pins</i>	: Pins (ORed value of PININTCH*)

**Returns**

Nothing

Definition at line 180 of file pinint\_5411x.h.



## 6.37 CHIP: LPC5411X Power LIBRARY functions

### 6.37.1 Detailed Description

The power library provides functions to control system power usage and place the device into low power modes.

#### Clock shutdown in sleep and power down modes

When using the [Chip\\_POWER\\_EnterPowerMode\(\)](#) function, system clocks are shutdown based on the selected sleep or power down mode and the device version being used. The following list details which clocks are shut down in which modes for which device versions. You can keep a clock enabled for a sleep or power down mode by enabling it in the 'peripheral\_ctrl' field in the [Chip\\_POWER\\_EnterPowerMode\(\)](#) function.

Mode: Sleep

No clocks are disabled for any chip version.

Mode: Deep sleep

SYSCON\_PDRUNCFG\_PD\_IRC\_OSC  
 SYSCON\_PDRUNCFG\_PD\_IRC  
 SYSCON\_PDRUNCFG\_PD\_FLASH (v17.1 and later only)  
 SYSCON\_PDRUNCFG\_PD\_BOD\_INTR  
 SYSCON\_PDRUNCFG\_PD\_ADC0  
 SYSCON\_PDRUNCFG\_PD\_ROM  
 SYSCON\_PDRUNCFG\_PD\_VDDA\_ENA  
 SYSCON\_PDRUNCFG\_PD\_SYS\_PLL  
 SYSCON\_PDRUNCFG\_PD\_VREFP

Mode: Power down

SYSCON\_PDRUNCFG\_PD\_IRC\_OSC  
 SYSCON\_PDRUNCFG\_PD\_IRC  
 SYSCON\_PDRUNCFG\_PD\_FLASH (v17.1 and later only)  
 SYSCON\_PDRUNCFG\_PD\_BOD\_RST  
 SYSCON\_PDRUNCFG\_PD\_BOD\_INTR  
 SYSCON\_PDRUNCFG\_PD\_ADC0  
 SYSCON\_PDRUNCFG\_PD\_SRAM0B  
 SYSCON\_PDRUNCFG\_PD\_SRAM1  
 SYSCON\_PDRUNCFG\_PD\_SRAM2  
 SYSCON\_PDRUNCFG\_PD\_ROM  
 SYSCON\_PDRUNCFG\_PD\_VDDA\_ENA  
 SYSCON\_PDRUNCFG\_PD\_WDT\_OSC  
 SYSCON\_PDRUNCFG\_PD\_SYS\_PLL  
 SYSCON\_PDRUNCFG\_PD\_VREFP  
 SYSCON\_PDRUNCFG\_PD\_32K\_OSC

Mode: Deep power down

All clocks are disabled for all chip versions.

If you are using a peripheral as a wakeup source for a power down mode, it needs to be kept active with the call to [Chip\\_POWER\\_EnterPowerMode\(\)](#). For example, if you are using the RTC to wake the system up from power down mode, the 32KHz RTC oscillator needs to remain active, so the power down call would look like this:

```
Chip_POWER_EnterPowerMode(POWER_POWER_DOWN, SYSCON_PDRUNCFG_PD_32K_OSC);
```

If your application uses internal RAM beyond the first 8K, you will also need to prevent power down of the IIRAM like this:

```
Chip_POWER_EnterPowerMode(POWER_POWER_DOWN, (SYSCON_PDRUNCFG_PD_32K_OSC | SYSCON_PDRUNCFG_PD_SRAM0A));
```

## Macros

- #define `LPC5411X_ROMVER_0` (0x1100)
- #define `LPC5411X_ROMVER_1` (0x1101)
- #define `LPC5411X_ROMVER_2` (0x1102)

## Enumerations

- enum `POWER_MODE_T` { `POWER_SLEEP` = 0, `POWER_DEEP_SLEEP`, `POWER_DEEP_POWER_DOWN` }

## Functions

- void `Chip_POWER_SetFROHFRate` (uint32\_t freq)  
*Sets the High Frequency FRO rate to (48MHz or 96MHz)*
- uint32\_t `Chip_POWER_SetPLL` (uint32\_t multiply\_by, uint32\_t input\_freq)  
*Sets up the System PLL given the PLL input frequency and feedback multiplier.*
- uint32\_t `Chip_POWER_SetVoltage` (uint32\_t desired\_freq)  
*Set optimal system voltage based on passed system frequency.*
- void `Chip_POWER_SetLowPowerVoltage` (uint32\_t freq)  
*Set low-power voltage levels for LP mode.*
- void `Chip_POWER_EnterPowerMode` (`POWER_MODE_T` mode, uint32\_t peripheral\_ctrl)  
*Enters the selected power state.*
- uint32\_t `Chip_POWER_GetROMVersion` (void)  
*Return ROM version.*

### 6.37.2 Macro Definition Documentation

#### 6.37.2.1 #define `LPC5411X_ROMVER_0` (0x1100)

Definition at line 152 of file `power_lib_5411x.h`.

#### 6.37.2.2 #define `LPC5411X_ROMVER_1` (0x1101)

Definition at line 153 of file `power_lib_5411x.h`.

#### 6.37.2.3 #define `LPC5411X_ROMVER_2` (0x1102)

Definition at line 154 of file `power_lib_5411x.h`.

### 6.37.3 Enumeration Type Documentation

#### 6.37.3.1 enum `POWER_MODE_T`

Enumerator

**`POWER_SLEEP`**

**`POWER_DEEP_SLEEP`**

**`POWER_DEEP_POWER_DOWN`**

Definition at line 100 of file `power_lib_5411x.h`.

### 6.37.4 Function Documentation

#### 6.37.4.1 void Chip\_POWER\_EnterPowerMode ( POWER\_MODE\_T mode, uint32\_t peripheral\_ctrl )

Enters the selected power state.

##### Parameters

<i>mode</i>	: Power mode
<i>peripheral_ctrl</i>	: Peripherals that will remain powered up in the power state

##### Returns

Nothing

##### Note

The 'peripheral\_ctrl' field is a bitmask of bits from the PDRUNCFG register (SYSCON\_PDRUNCFG\_PD\_\*) that describe which peripherals can wake up the chip from the power state. These peripherals are not powered down during the power state.

#### 6.37.4.2 uint32\_t Chip\_POWER\_GetROMVersion ( void )

Return ROM version.

##### Returns

ROM version

##### Note

Will return one of the following version numbers:  
(0x1100) for v17.0 ROMs.  
(0x1101) for v17.1 ROMs.  
(0x1102) for v17.2 ROMs.

#### 6.37.4.3 void Chip\_POWER\_SetFROHFRate ( uint32\_t freq )

Sets the High Frequency FRO rate to (48MHz or 96MHz)

##### Parameters

<i>freq</i>	: Frequency selection for FRO
-------------	-------------------------------

##### Returns

none

#### 6.37.4.4 void Chip\_POWER\_SetLowPowerVoltage ( uint32\_t freq )

Set low-power voltage levels for LP mode.

**Parameters**

<i>frequency</i>	: This is the frequency at which CPU is running.
------------------	--

**Note**

Low power mode is only possible at 12MHz and 48MHz FRO

**Returns**

Nothing

**6.37.4.5 uint32\_t Chip\_POWER\_SetPLL ( uint32\_t *multiply\_by*, uint32\_t *input\_freq* )**

Sets up the System PLL given the PLL input frequency and feedback multiplier.

**Parameters**

<i>multiply_by</i>	: PLL multiplier, minimum of 1, maximum of 16
<i>input_freq</i>	: Input frequency into the PLL

**Returns**

LPC\_OK on success, or an error code (see [error.h](#))

**6.37.4.6 uint32\_t Chip\_POWER\_SetVoltage ( uint32\_t *desired\_freq* )**

Set optimal system voltage based on passed system frequency.

**Parameters**

<i>desired_freq</i>	: System (CPU) frequency
---------------------	--------------------------

**Returns**

LPC\_OK on success, or an error code (see [error.h](#))

**Note**

This function will adjust the system voltages to the lowest levels that will support the passed CPU frequency.

## 6.38 CHIP: LPC5411X Power Management declarations and functions

### 6.38.1 Detailed Description

#### Data Structures

- struct [LPC\\_PMU\\_T](#)  
*PMU register block structure.*

#### Macros

- #define [PMU\\_BOD\\_RST](#) (1 << 6)
- #define [PMU\\_BOD\\_INT](#) (1 << 7)

#### Enumerations

- enum [CHIP\\_PMU\\_BODRSTLVL\\_T](#) {  
[PMU\\_BODRSTLVL\\_0](#), [PMU\\_BODRSTLVL\\_1\\_50V](#) = [PMU\\_BODRSTLVL\\_0](#), [PMU\\_BODRSTLVL\\_1](#), [PMU\\_BODRSTLVL\\_1\\_85V](#) = [PMU\\_BODRSTLVL\\_1](#),  
[PMU\\_BODRSTLVL\\_2](#), [PMU\\_BODRSTLVL\\_2\\_00V](#) = [PMU\\_BODRSTLVL\\_2](#), [PMU\\_BODRSTLVL\\_3](#), [PMU\\_BODRSTLVL\\_2\\_30V](#) = [PMU\\_BODRSTLVL\\_3](#) }
- enum [CHIP\\_PMU\\_BODRINTVAL\\_T](#) {  
[PMU\\_BODINTVAL\\_LVL0](#), [PMU\\_BODINTVAL\\_2\\_05v](#) = [PMU\\_BODINTVAL\\_LVL0](#), [PMU\\_BODINTVAL\\_LVL1](#), [PMU\\_BODINTVAL\\_2\\_45v](#) = [PMU\\_BODINTVAL\\_LVL1](#),  
[PMU\\_BODINTVAL\\_LVL2](#), [PMU\\_BODINTVAL\\_2\\_75v](#) = [PMU\\_BODINTVAL\\_LVL2](#), [PMU\\_BODINTVAL\\_LVL3](#), [PMU\\_BODINTVAL\\_3\\_05v](#) = [PMU\\_BODINTVAL\\_LVL3](#) }

#### Functions

- `__STATIC_INLINE void Chip\_PMU\_SetBODLevels (CHIP\_PMU\_BODRSTLVL\_T rstlvl, CHIP\_PMU\_BODRINTVAL\_T intlvl)`  
*Set brown-out detection interrupt and reset levels.*
- `__STATIC_INLINE void Chip\_PMU\_EnableBODReset (void)`  
*Enable brown-out detection reset.*
- `__STATIC_INLINE void Chip\_PMU\_DisableBODReset (void)`  
*Disable brown-out detection reset.*
- `__STATIC_INLINE void Chip\_PMU\_EnableBODInt (void)`  
*Enable brown-out detection interrupt.*
- `__STATIC_INLINE void Chip\_PMU\_DisableBODInt (void)`  
*Disable brown-out detection interrupt.*

### 6.38.2 Macro Definition Documentation

#### 6.38.2.1 #define [PMU\\_BOD\\_INT](#) (1 << 7)

brown-out detection interrupt status (in BODCTRL register)

Definition at line 88 of file `pmu_5411x.h`.

#### 6.38.2.2 #define [PMU\\_BOD\\_RST](#) (1 << 6)

brown-out detection reset status (in BODCTRL register)

Definition at line 84 of file `pmu_5411x.h`.

### 6.38.3 Enumeration Type Documentation

#### 6.38.3.1 enum CHIP\_PMU\_BODRINTVAL\_T

Brown-out detector interrupt level

Enumerator

***PMU\_BODINTVAL\_LVL0*** Brown-out interrupt at ~2.05v  
***PMU\_BODINTVAL\_2\_05v***  
***PMU\_BODINTVAL\_LVL1*** Brown-out interrupt at ~2.45v  
***PMU\_BODINTVAL\_2\_45v***  
***PMU\_BODINTVAL\_LVL2*** Brown-out interrupt at ~2.75v  
***PMU\_BODINTVAL\_2\_75v***  
***PMU\_BODINTVAL\_LVL3*** Brown-out interrupt at ~3.05v  
***PMU\_BODINTVAL\_3\_05v***

Definition at line 70 of file pmu\_5411x.h.

#### 6.38.3.2 enum CHIP\_PMU\_BODRSTLVL\_T

Brown-out detector reset level

Enumerator

***PMU\_BODRSTLVL\_0*** Brown-out reset at ~1.5v  
***PMU\_BODRSTLVL\_1\_50V***  
***PMU\_BODRSTLVL\_1*** Brown-out reset at ~1.85v  
***PMU\_BODRSTLVL\_1\_85V***  
***PMU\_BODRSTLVL\_2*** Brown-out reset at ~2.0v  
***PMU\_BODRSTLVL\_2\_00V***  
***PMU\_BODRSTLVL\_3*** Brown-out reset at ~2.3v  
***PMU\_BODRSTLVL\_2\_30V***

Definition at line 56 of file pmu\_5411x.h.

### 6.38.4 Function Documentation

#### 6.38.4.1 \_\_STATIC\_INLINE void Chip\_PMU\_DisableBODInt ( void )

Disable brown-out detection interrupt.

Returns

Nothing

Definition at line 135 of file pmu\_5411x.h.

#### 6.38.4.2 \_\_STATIC\_INLINE void Chip\_PMU\_DisableBODReset ( void )

Disable brown-out detection reset.

Returns

Nothing

Definition at line 117 of file pmu\_5411x.h.

**6.38.4.3** `__STATIC_INLINE void Chip_PMU_EnableBODInt ( void )`

Enable brown-out detection interrupt.

**Returns**

Nothing

Definition at line 126 of file pmu\_5411x.h.

**6.38.4.4** `__STATIC_INLINE void Chip_PMU_EnableBODReset ( void )`

Enable brown-out detection reset.

**Returns**

Nothing

Definition at line 108 of file pmu\_5411x.h.

**6.38.4.5** `__STATIC_INLINE void Chip_PMU_SetBODLevels ( CHIP_PMU_BODRSTLVL_T rstlvl,  
CHIP_PMU_BODRINTVAL_T intlvl )`

Set brown-out detection interrupt and reset levels.

**Parameters**

<i>rstlvl</i>	: Brown-out detector reset level
<i>intlvl</i>	: Brown-out interrupt level

**Returns**

Nothing

**Note**

Brown-out detection reset will be disabled upon exiting this function. Use [Chip\\_PMU\\_EnableBODReset\(\)](#) to re-enable.

Definition at line 98 of file pmu\_5411x.h.

## 6.39 CHIP: LPC5411X ROM API declarations and functions

### 6.39.1 Detailed Description

#### Data Structures

- struct [LPC\\_ROM\\_API\\_T](#)  
*High level ROM API structure.*

#### Macros

- `#define LPC_ROM_API_BASE_LOC 0x03000200UL`
- `#define LPC_ROM_API (*(LPC_ROM_API_T *) LPC_ROM_API_BASE_LOC)`
- `#define IAP_ENTRY_LOCATION 0x03000205`

#### Functions

- static `INLINE` void [iap\\_entry](#) (unsigned int cmd\_param[5], unsigned int status\_result[4])  
*LPC5410x IAP\_ENTRY API function type.*

### 6.39.2 Macro Definition Documentation

#### 6.39.2.1 `#define IAP_ENTRY_LOCATION 0x03000205`

Definition at line 78 of file romapi\_5411x.h.

#### 6.39.2.2 `#define LPC_ROM_API (*(LPC_ROM_API_T *) LPC_ROM_API_BASE_LOC)`

Definition at line 72 of file romapi\_5411x.h.

#### 6.39.2.3 `#define LPC_ROM_API_BASE_LOC 0x03000200UL`

Definition at line 71 of file romapi\_5411x.h.

### 6.39.3 Function Documentation

#### 6.39.3.1 `static INLINE void iap_entry ( unsigned int cmd_param[5], unsigned int status_result[4] ) [static]`

LPC5410x IAP\_ENTRY API function type.

Definition at line 83 of file romapi\_5411x.h.



## 6.40 CHIP: LPC5411X Real Time clock

### 6.40.1 Detailed Description

#### Data Structures

- struct [LPC\\_RTC\\_T](#)  
*LPC5411X Real Time clock register block structure.*

#### Macros

- #define [RTC\\_CTRL\\_SWRESET](#) (1 << 0)
- #define [RTC\\_CTRL\\_ALARM1HZ](#) (1 << 2)
- #define [RTC\\_CTRL\\_WAKE1KHZ](#) (1 << 3)
- #define [RTC\\_CTRL\\_ALARMDPD\\_EN](#) (1 << 4)
- #define [RTC\\_CTRL\\_WAKEDPD\\_EN](#) (1 << 5)
- #define [RTC\\_CTRL\\_RTC1KHZ\\_EN](#) (1 << 6)
- #define [RTC\\_CTRL\\_RTC\\_EN](#) (1 << 7)
- #define [RTC\\_CTRL\\_RTC\\_OSC\\_PD](#) (1 << 8)
- #define [RTC\\_CTRL\\_RTC\\_OSC\\_BYPASS](#) (1 << 9)
- #define [RTC\\_CTRL\\_MASK](#) ((uint32\_t) 0x0000003FD)

#### Functions

- `__STATIC_INLINE void Chip\_RTC\_Init (LPC\_RTC\_T *pRTC)`  
*Initialize the RTC peripheral.*
- `__STATIC_INLINE void Chip\_RTC\_DeInit (LPC\_RTC\_T *pRTC)`  
*De-initialize the RTC peripheral.*
- `__STATIC_INLINE void Chip\_RTC\_EnableOptions (LPC\_RTC\_T *pRTC, uint32_t flags)`  
*Enable RTC options.*
- `__STATIC_INLINE void Chip\_RTC\_DisableOptions (LPC\_RTC\_T *pRTC, uint32_t flags)`  
*Disable RTC options.*
- `__STATIC_INLINE void Chip\_RTC\_Reset (LPC\_RTC\_T *pRTC)`  
*Reset RTC.*
- `__STATIC_INLINE void Chip\_RTC\_Enable (LPC\_RTC\_T *pRTC)`  
*Enables the RTC.*
- `__STATIC_INLINE void Chip\_RTC\_Disable (LPC\_RTC\_T *pRTC)`  
*Disables the RTC.*
- `__STATIC_INLINE void Chip\_RTC\_PowerUp (LPC\_RTC\_T *pRTC)`  
*Power up the RTC.*
- `__STATIC_INLINE void Chip\_RTC\_PowerDown (LPC\_RTC\_T *pRTC)`  
*Disables the RTC.*
- `__STATIC_INLINE void Chip\_RTC\_Enable1KHZ (LPC\_RTC\_T *pRTC)`  
*Enables the RTC 1KHz high resolution timer.*
- `__STATIC_INLINE void Chip\_RTC\_Disable1KHZ (LPC\_RTC\_T *pRTC)`  
*Disables the RTC 1KHz high resolution timer.*
- `__STATIC_INLINE void Chip\_RTC\_EnableWakeup (LPC\_RTC\_T *pRTC, uint32_t ints)`  
*Enables selected RTC wakeup events.*
- `__STATIC_INLINE void Chip\_RTC\_DisableWakeup (LPC\_RTC\_T *pRTC, uint32_t ints)`  
*Disables selected RTC wakeup events.*
- `__STATIC_INLINE void Chip\_RTC\_ClearStatus (LPC\_RTC\_T *pRTC, uint32_t stsMask)`

- Clears latched RTC statuses.*
- `__STATIC_INLINE uint32_t Chip_RTC_GetStatus (LPC_RTC_T *pRTC)`  
*Return RTC control/status register.*
- `__STATIC_INLINE void Chip_RTC_SetAlarm (LPC_RTC_T *pRTC, uint32_t count)`  
*Set RTC match value for alarm status/interrupt.*
- `__STATIC_INLINE uint32_t Chip_RTC_GetAlarm (LPC_RTC_T *pRTC)`  
*Return the RTC match value used for alarm status/interrupt.*
- `__STATIC_INLINE void Chip_RTC_SetCount (LPC_RTC_T *pRTC, uint32_t count)`  
*Set RTC match count for 1 second timer count.*
- `__STATIC_INLINE uint32_t Chip_RTC_GetCount (LPC_RTC_T *pRTC)`  
*Get current RTC 1 second timer count.*
- `__STATIC_INLINE void Chip_RTC_SetWake (LPC_RTC_T *pRTC, uint16_t count)`  
*Set RTC wake count countdown value (in mS ticks)*
- `__STATIC_INLINE uint16_t Chip_RTC_GetWake (LPC_RTC_T *pRTC)`  
*Get RTC wake count countdown value.*

## 6.40.2 Macro Definition Documentation

### 6.40.2.1 `#define RTC_CTRL_ALARM1HZ (1 << 2)`

RTC 1 Hz timer alarm flag status (match) bit

Definition at line 56 of file rtc\_5411x.h.

### 6.40.2.2 `#define RTC_CTRL_ALARMDPD_EN (1 << 4)`

RTC 1 Hz timer alarm for Deep power-down enable bit

Definition at line 58 of file rtc\_5411x.h.

### 6.40.2.3 `#define RTC_CTRL_MASK ((uint32_t) 0x0000003FD)`

RTC Control register Mask for reserved bits

Definition at line 64 of file rtc\_5411x.h.

### 6.40.2.4 `#define RTC_CTRL_RTC1KHZ_EN (1 << 6)`

RTC 1 kHz clock enable bit

Definition at line 60 of file rtc\_5411x.h.

### 6.40.2.5 `#define RTC_CTRL_RTC_EN (1 << 7)`

RTC enable bit

Definition at line 61 of file rtc\_5411x.h.

### 6.40.2.6 `#define RTC_CTRL_RTC_OSC_BYPASS (1 << 9)`

RTC power-down bit

Definition at line 63 of file rtc\_5411x.h.

**6.40.2.7 #define RTC\_CTRL\_RTC\_OSC\_PD (1 << 8)**

RTC power-down bit

Definition at line 62 of file rtc\_5411x.h.

**6.40.2.8 #define RTC\_CTRL\_SWRESET (1 << 0)**

Apply reset to RTC

Definition at line 55 of file rtc\_5411x.h.

**6.40.2.9 #define RTC\_CTRL\_WAKE1KHZ (1 << 3)**

RTC 1 kHz timer wake-up flag status (timeout) bit

Definition at line 57 of file rtc\_5411x.h.

**6.40.2.10 #define RTC\_CTRL\_WAKEDPD\_EN (1 << 5)**

RTC 1 kHz timer wake-up for Deep power-down enable bit

Definition at line 59 of file rtc\_5411x.h.

**6.40.3 Function Documentation****6.40.3.1 \_\_STATIC\_INLINE void Chip\_RTC\_ClearStatus ( LPC\_RTC\_T \* pRTC, uint32\_t stsMask )**

Clears latched RTC statuses.

Parameters

<i>pRTC</i>	: The base address of RTC block
<i>stsMask</i>	: OR'ed status bits to clear

Returns

Nothing

Note

Use and OR'ed stsMask value of RTC\_CTRL\_ALARM1HZ, and RTC\_CTRL\_WAKE1KHZ to clear specific RTC states.

Definition at line 240 of file rtc\_5411x.h.

**6.40.3.2 \_\_STATIC\_INLINE void Chip\_RTC\_DeInit ( LPC\_RTC\_T \* pRTC )**

De-initialize the RTC peripheral.

Parameters

<i>pRTC</i>	: RTC peripheral selected
-------------	---------------------------

Returns

None

Definition at line 82 of file rtc\_5411x.h.

#### 6.40.3.3 `__STATIC_INLINE void Chip_RTC_Disable ( LPC_RTC_T * pRTC )`

Disables the RTC.

##### Parameters

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

##### Returns

Nothing

##### Note

You can also use [Chip\\_RTC\\_DisableOptions\(\)](#) with the `RTC_CTRL_RTC_EN` flag to enable the RTC.

Definition at line 148 of file `rtc_5411x.h`.

#### 6.40.3.4 `__STATIC_INLINE void Chip_RTC_Disable1KHZ ( LPC_RTC_T * pRTC )`

Disables the RTC 1KHz high resolution timer.

##### Parameters

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

##### Returns

Nothing

##### Note

You can also use [Chip\\_RTC\\_DisableOptions\(\)](#) with the `RTC_CTRL_RTC1KHZ_EN` flag to disable the high resolution timer.

Definition at line 197 of file `rtc_5411x.h`.

#### 6.40.3.5 `__STATIC_INLINE void Chip_RTC_DisableOptions ( LPC_RTC_T * pRTC, uint32_t flags )`

Disable RTC options.

##### Parameters

<i>pRTC</i>	: The base address of RTC block
<i>flags</i>	: And OR'ed value of <code>RTC_CTRL_*</code> definitions to disable

##### Returns

Nothing

##### Note

You can enable multiple RTC options at once using this function by OR'ing them together. It is recommended to only use the `RTC_CTRL_ALARMDPD_EN`, `RTC_CTRL_WAKEDPD_EN`, `RTC_CTRL_RTC1KHZ_EN`, and `RTC_CTRL_RTC_EN` flags with this function.

Definition at line 112 of file `rtc_5411x.h`.

#### 6.40.3.6 `__STATIC_INLINE void Chip_RTC_DisableWakeup ( LPC_RTC_T * pRTC, uint32_t ints )`

Disables selected RTC wakeup events.

## Parameters

<i>pRTC</i>	: The base address of RTC block
<i>ints</i>	: Wakeup events to disable

## Returns

Nothing

## Note

Select either one or both (OR'ed) RTC\_CTRL\_ALARMDPD\_EN and RTC\_CTRL\_WAKEDPD\_EN values to disabled. You can also use [Chip\\_RTC\\_DisableOptions\(\)](#) with the flags to disable the events.

Definition at line 227 of file rtc\_5411x.h.

#### 6.40.3.7 `__STATIC_INLINE void Chip_RTC_Enable ( LPC_RTC_T * pRTC )`

Enables the RTC.

## Parameters

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

## Returns

Nothing

## Note

You can also use [Chip\\_RTC\\_EnableOptions\(\)](#) with the RTC\_CTRL\_RTC\_EN flag to enable the RTC.

Definition at line 136 of file rtc\_5411x.h.

#### 6.40.3.8 `__STATIC_INLINE void Chip_RTC_Enable1KHZ ( LPC_RTC_T * pRTC )`

Enables the RTC 1KHz high resolution timer.

## Parameters

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

## Returns

Nothing

## Note

You can also use [Chip\\_RTC\\_EnableOptions\(\)](#) with the RTC\_CTRL\_RTC1KHZ\_EN flag to enable the high resolution timer.

Definition at line 184 of file rtc\_5411x.h.

#### 6.40.3.9 `__STATIC_INLINE void Chip_RTC_EnableOptions ( LPC_RTC_T * pRTC, uint32_t flags )`

Enable RTC options.

## Parameters

<i>pRTC</i>	: The base address of RTC block
<i>flags</i>	: And OR'ed value of RTC_CTRL_* definitions to enable

## Returns

Nothing

## Note

You can enable multiple RTC options at once using this function by OR'ing them together. It is recommended to only use the RTC\_CTRL\_ALARMDPD\_EN, RTC\_CTRL\_WAKEDPD\_EN, RTC\_CTRL\_RTC1KHZ\_EN, and RTC\_CTRL\_RTC\_EN flags with this function.

Definition at line 97 of file rtc\_5411x.h.

#### 6.40.3.10 `__STATIC_INLINE void Chip_RTC_EnableWakeup ( LPC_RTC_T * pRTC, uint32_t ints )`

Enables selected RTC wakeup events.

## Parameters

<i>pRTC</i>	: The base address of RTC block
<i>ints</i>	: Wakeup events to enable

## Returns

Nothing

## Note

Select either one or both (OR'ed) RTC\_CTRL\_ALARMDPD\_EN and RTC\_CTRL\_WAKEDPD\_EN values to enabled. You can also use [Chip\\_RTC\\_EnableOptions\(\)](#) with the flags to enable the events.

Definition at line 212 of file rtc\_5411x.h.

#### 6.40.3.11 `__STATIC_INLINE uint32_t Chip_RTC_GetAlarm ( LPC_RTC_T * pRTC )`

Return the RTC match value used for alarm status/interrupt.

## Parameters

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

## Returns

Alarm event time

Definition at line 274 of file rtc\_5411x.h.

#### 6.40.3.12 `__STATIC_INLINE uint32_t Chip_RTC_GetCount ( LPC_RTC_T * pRTC )`

Get current RTC 1 second timer count.

## Parameters

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

## Returns

current RTC 1 second timer count

Definition at line 298 of file rtc\_5411x.h.

#### 6.40.3.13 `__STATIC_INLINE uint32_t Chip_RTC_GetStatus ( LPC_RTC_T * pRTC )`

Return RTC control/status register.

## Parameters

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

## Returns

The current RTC control/status register

## Note

Mask the return value with a `RTC_CTRL_*` definitions to determine which bits are set. For example, mask the return value with `RTC_CTRL_ALARM1HZ` to determine if the alarm interrupt is pending.

Definition at line 253 of file rtc\_5411x.h.

#### 6.40.3.14 `__STATIC_INLINE uint16_t Chip_RTC_GetWake ( LPC_RTC_T * pRTC )`

Get RTC wake count countdown value.

## Parameters

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

## Returns

current RTC wake count countdown value (in mS)

Definition at line 321 of file rtc\_5411x.h.

#### 6.40.3.15 `__STATIC_INLINE void Chip_RTC_Init ( LPC_RTC_T * pRTC )`

Initialize the RTC peripheral.

## Parameters

<i>pRTC</i>	: RTC peripheral selected
-------------	---------------------------

## Returns

None

Definition at line 72 of file rtc\_5411x.h.

#### 6.40.3.16 `__STATIC_INLINE void Chip_RTC_PowerDown ( LPC_RTC_T * pRTC )`

Disables the RTC.

**Parameters**

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

**Returns**

Nothing

**Note**

You can also use [Chip\\_RTC\\_DisableOptions\(\)](#) with the RTC\_CTRL\_RTC\_EN flag to enable the RTC.

Definition at line 170 of file rtc\_5411x.h.

**6.40.3.17** `__STATIC_INLINE void Chip_RTC_PowerUp ( LPC_RTC_T * pRTC )`

Power up the RTC.

**Parameters**

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

**Returns**

Nothing

Definition at line 158 of file rtc\_5411x.h.

**6.40.3.18** `__STATIC_INLINE void Chip_RTC_Reset ( LPC_RTC_T * pRTC )`

Reset RTC.

**Parameters**

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------

**Returns**

Nothing

**Note**

The RTC state will be returned to it's default.

Definition at line 123 of file rtc\_5411x.h.

**6.40.3.19** `__STATIC_INLINE void Chip_RTC_SetAlarm ( LPC_RTC_T * pRTC, uint32_t count )`

Set RTC match value for alarm status/interrupt.

**Parameters**

<i>pRTC</i>	: The base address of RTC block
-------------	---------------------------------



<i>count</i>	: Alarm event time
--------------	--------------------

**Returns**

Nothing

Definition at line 264 of file rtc\_5411x.h.

**6.40.3.20** `__STATIC_INLINE void Chip_RTC_SetCount ( LPC_RTC_T * pRTC, uint32_t count )`

Set RTC match count for 1 second timer count.

**Parameters**

<i>pRTC</i>	: The base address of RTC block
<i>count</i>	: Initial count to set

**Returns**

Nothing

**Note**

Only write to this register when the RTC\_CTRL\_RTC\_EN bit in the CTRL Register is 0. The counter increments one second after the RTC\_CTRL\_RTC\_EN bit is set.

Definition at line 288 of file rtc\_5411x.h.

**6.40.3.21** `__STATIC_INLINE void Chip_RTC_SetWake ( LPC_RTC_T * pRTC, uint16_t count )`

Set RTC wake count countdown value (in mS ticks)

**Parameters**

<i>pRTC</i>	: The base address of RTC block
<i>count</i>	: wakeup time in milliSeconds

**Returns**

Nothing

**Note**

A write pre-loads a start count value into the wake-up timer and initializes a count-down sequence.

Definition at line 311 of file rtc\_5411x.h.

## 6.41 CHIP: LPC5411X SPI driver

### 6.41.1 Detailed Description

The SPI interface is provided via FlexCOMM module in the CHIP, to associate a SPI to a flexcom the source file must add a define like **#define SPI0\_FLEXCOMM 5** which will make LPC\_SPI0 use LPC\_FLEXCOMM5, interrupt service function name and IRQ number DMA REQUEST number will all be mapped automatically. **Note that this define must be available/visible to all the sources that uses SPI.**

The SPI driver by default will use FIFOs for TX and RX. TX and RX FIFOs has a depth of 8 entries each.

### Data Structures

- struct [LPC\\_SPI\\_T](#)  
*SPI register block structure.*
- struct [SPI\\_CFGSETUP\\_T](#)

### Modules

- [CHIP: LPC5411X SPI master driver](#)
- [CHIP: LPC5411X SPI slave driver](#)

### Macros

- #define [SPI\\_CFG\\_BITMASK](#) (0x0FBD) /\*\* SPI register bit mask \*/
- #define [SPI\\_CFG\\_SPI\\_EN](#) (1 << 0) /\*\* SPI Slave Mode Select \*/
- #define [SPI\\_CFG\\_SLAVE\\_EN](#) (0 << 0) /\*\* SPI Master Mode Select \*/
- #define [SPI\\_CFG\\_MASTER\\_EN](#) (1 << 2) /\*\* SPI MSB First mode enable \*/
- #define [SPI\\_CFG\\_MSB\\_FIRST\\_EN](#) (0 << 3) /\*\* SPI LSB First mode enable \*/
- #define [SPI\\_CFG\\_LSB\\_FIRST\\_EN](#) (1 << 3) /\*\* SPI Clock Phase Select \*/
- #define [SPI\\_CFG\\_CPHA\\_FIRST](#) (0 << 4) /\*\* Capture data on the first edge, Change data on the following edge \*/
- #define [SPI\\_CFG\\_CPHA\\_SECOND](#) (1 << 4) /\*\* SPI Clock Polarity Select \*/
- #define [SPI\\_CFG\\_CPOL\\_LO](#) (0 << 5) /\*\* The rest state of the clock (between frames) is low. \*/
- #define [SPI\\_CFG\\_CPOL\\_HI](#) (1 << 5) /\*\* The rest state of the clock (between frames) is high. \*/
- #define [SPI\\_CFG\\_LBM\\_EN](#) (1 << 7) /\*\* SPI control 1 loopback mode enable \*/
- #define [SPI\\_CFG\\_SPOL\\_LO](#) (0 << 8) /\*\* SPI SSEL0 Polarity Select \*/
- #define [SPI\\_CFG\\_SPOL\\_HI](#) (1 << 8) /\*\* SSEL0 is active High \*/
- #define [SPI\\_CFG\\_SPOLNUM\\_HI](#)(n) (1 << ((n) + 8)) /\*\* SSELN is active High, selects 0 - 3 \*/
- #define [SPI\\_DLY\\_BITMASK](#) (0xFFFF) /\*\* SPI DLY Register Mask \*/
- #define [SPI\\_DLY\\_PRE\\_DELAY](#)(n) (((n) & 0x0F) << 0) /\*\* Time in SPI clocks between SSEL assertion and the beginning of a data frame \*/
- #define [SPI\\_DLY\\_POST\\_DELAY](#)(n) (((n) & 0x0F) << 4) /\*\* Time in SPI clocks between the end of a data frame and SSEL deassertion. \*/
- #define [SPI\\_DLY\\_FRAME\\_DELAY](#)(n) (((n) & 0x0F) << 8) /\*\* Minimum time in SPI clocks between adjacent data frames. \*/
- #define [SPI\\_DLY\\_TRANSFER\\_DELAY](#)(n) (((n) & 0x0F) << 12) /\*\* Minimum time in SPI clocks that the SSEL is deasserted between transfers. \*/
- #define [SPI\\_STAT\\_BITMASK](#) (0x01F0) /\*\* SPI STAT Register BitMask \*/
- #define [SPI\\_STAT\\_SSA](#) (1 << 4) /\*\* Slave Select Assert \*/
- #define [SPI\\_STAT\\_SSD](#) (1 << 5) /\*\* Slave Select Deassert \*/
- #define [SPI\\_STAT\\_STALLED](#) (1 << 6) /\*\* Stalled status flag \*/
- #define [SPI\\_STAT\\_EOT](#) (1 << 7) /\*\* End Transfer flag \*/

- #define SPI\_STAT\_MSTIDLE (1 << 8) /\*\* Idle status flag \*/
- #define SPI\_INT\_BITMASK (0x0130) /\*\* SPI interrupt Enable/Disable bits \*/
- #define SPI\_INT\_SSAEN (1 << 4) /\*\* Slave Select is asserted interrupt [BIT-4 of INTENSET/INTENCLR/INTSTAT register] \*/
- #define SPI\_INT\_SSDEN (1 << 5) /\*\* Slave Select is deasserted interrupt [BIT-5 of INTENSET/INTENCLR/INTSTAT register] \*/
- #define SPI\_INT\_MSTIDLE (1 << 8) /\*\* SPI master is Idle [BIT-8 of INTENSET/INTENCLR/INTSTAT register] \*/
- #define SPI\_FIFOCFG\_ENABLETX (1 << 0)
- SPI FIFO Configuration register bits.*
- #define SPI\_FIFOCFG\_ENABLERX (1 << 1)
- #define SPI\_FIFOCFG\_DMATX (1 << 12)
- #define SPI\_FIFOCFG\_DMARX (1 << 13)
- #define SPI\_FIFOCFG\_WAKETX (1 << 14)
- #define SPI\_FIFOCFG\_WAKERX (1 << 15)
- #define SPI\_FIFOCFG\_EMPTYTX (1 << 16)
- #define SPI\_FIFOCFG\_EMPTYRX (1 << 17)
- #define SPI\_FIFO\_DEPTH (8) /\*\* SPI-FIFO How many entries are in the FIFO \*/
- Macro defines for FIFO Status register.*
- #define SPI\_FIFOSTAT\_BITMASK (0x1F1FFB) /\*\* SPI-FIFO STAT Register BitMask \*/
- #define SPI\_FIFOSTAT\_TXERR (1 << 0) /\*\* SPI-FIFO transmit error \*/
- #define SPI\_FIFOSTAT\_RXERR (1 << 1) /\*\* SPI-FIFO receive error \*/
- #define SPI\_FIFOSTAT\_PERINT (1 << 3) /\*\* SPI-FIFO peripheral (SPI) interrupt \*/
- #define SPI\_FIFOSTAT\_TXEMPTY (1 << 4) /\*\* SPI-FIFO transmitter empty \*/
- #define SPI\_FIFOSTAT\_TXNOTFULL (1 << 5) /\*\* SPI-FIFO transmitter not full \*/
- #define SPI\_FIFOSTAT\_RXNOTEMPTY (1 << 6) /\*\* SPI-FIFO receiver not empty \*/
- #define SPI\_FIFOSTAT\_RXFULL (1 << 7) /\*\* SPI-FIFO receiver not full \*/
- #define SPI\_FIFOSTAT\_TXLVL(val) (((val) >> 8) & 0x1F) /\*\* SPI-FIFO extract transmit level \*/
- #define SPI\_FIFOSTAT\_RXLVL(val) (((val) >> 16) & 0x1F) /\*\* SPI-FIFO extract receive level \*/
- #define SPI\_FIFOTRIG\_BITMASK (0x000f0f03) /\*\* SPI FIFO trigger settings Register BitMask \*/
- UART FIFO trigger settings register defines.*
- #define SPI\_FIFOTRIG\_TXLVLENA (1 << 0)
- #define SPI\_FIFOTRIG\_RXLVLENA (1 << 1)
- #define SPI\_FIFOTRIG\_TXLVL(lvl) ((lvl & 0x0f) << 8)
- #define SPI\_FIFOTRIG\_RXLVL(lvl) ((lvl & 0x0f) << 16)
- #define SPI\_FIFOTRIG\_TXLVL\_DEFAULT 4
- #define SPI\_FIFOTRIG\_RXLVL\_DEFAULT 0
- #define SPI\_FIFOINT\_BITMASK (0x001F) /\*\* FIFO interrupt Bit mask \*/
- Macro defines for SPI Interrupt enable/disable/status [INTSET/INTCLR/INTSTAT and FIFOINTSET/FIFOINTCLR/FIFOINTSTAT registers].*
- #define SPI\_FIFOINT\_TXERR (1 << 0) /\*\* TX error interrupt [BIT-0 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/
- #define SPI\_FIFOINT\_RXERR (1 << 1) /\*\* RX error interrupt [BIT-1 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/
- #define SPI\_FIFOINT\_TXLVL (1 << 2) /\*\* TX FIFO ready interrupt [BIT-2 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/
- #define SPI\_FIFOINT\_RXLVL (1 << 3) /\*\* RX Data ready interrupt [BIT-3 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/
- #define SPI\_FIFOINT\_PERINT (1 << 4) /\*\* SPI peripheral interrupt [BIT-4 of FIFOINTSTAT register] \*/
- #define SPI\_RXDAT\_BITMASK (0x1FFFFFF) /\*\* SPI RXDAT Register BitMask \*/
- #define SPI\_RXDAT\_DATA(n) ((n) & 0xFFFF) /\*\* Receiver Data \*/
- #define SPI\_RXDAT\_RXSSELN(n) (((n) >> 16) & 0x0f) & 0x0f /\*\* Determine the active SSEL pin \*/
- #define SPI\_RXDAT\_RXSSELN\_ACTIVE (0 << 16) /\*\* The state of SSEL pin is active \*/
- #define SPI\_RXDAT\_SOT (1 << 20) /\*\* Start of Transfer flag \*/

- #define `SPI_TXDAT_BITMASK` (0xF7FFFFFF) /\*\* SPI TXDATCTL Register BitMask \*/
- #define `SPI_TXDAT_DATA`(n) ((n) & 0xFFFF) /\*\* SPI Transmit Data \*/
- #define `SPI_TXDAT_DATA`(n) ((n) & 0xFFFF) /\*\* SPI Transmit Data \*/
- #define `SPI_TXDAT_CTRLMASK` (0xF7F) /\*\* SPI TXDATCTL Register BitMask for control bits only \*/
- #define `SPI_TXDAT_ASSERT_SSEL` (0) /\*\* Assert SSEL0 pin \*/
- #define `SPI_TXDAT_ASSERTNUM_SSEL`(n) ((~(1 << (n))) & 0x0f) /\*\* Assert SSELN pin \*/
- #define `SPI_TXDAT_DEASSERT_SSEL` (1) /\*\* Deassert SSEL0 pin \*/
- #define `SPI_TXDAT_DEASSERTNUM_SSEL`(n) (1 << (n)) /\*\* Deassert SSELN pin \*/
- #define `SPI_TXDAT_DEASSERT_ALL` (0xF) /\*\* Deassert all SSEL pins \*/
- #define `SPI_TXDAT_EOT` (1 << 4) /\*\* End of Transfer flag (TRANSFER\_DELAY is applied after sending the current frame) \*/
- #define `SPI_TXDAT_EOF` (1 << 5) /\*\* End of Frame flag (FRAME\_DELAY is applied after sending the current part) \*/
- #define `SPI_TXDAT_RXIGNORE` (1 << 6) /\*\* Receive Ignore Flag \*/
- #define `SPI_TXDAT_FLEN`(n) (((n) & 0x0F) << 8) /\*\* Frame length - 1 \*/
- #define `SPI_TXDAT_FLENMASK` (0xF << 8) /\*\* Frame length mask \*/
- #define `SPI_DIV_VAL`(n) ((n) & 0xFFFF) /\*\* Rate divider value mask (In Master Mode only)\*/
- #define `Chip_SPI_ReadFIFO` `Chip_SPI_ReadRawRXFifo`
- #define `Chip_SPI_ReadFIFOdata` `Chip_SPI_ReadRXData`
- #define `Chip_SPI_WriteFIFOcd` `Chip_SPI_SetTXCTRLData`
- #define `Chip_SPI_WriteFIFOdata` `Chip_SPI_WriteTXData`
- #define `Chip_SPI_FlushFIFOs` `Chip_SPI_FlushFifos`

## Enumerations

- enum `ROM_SPI_CLOCK_MODE_T` {  
`ROM_SPI_CLOCK_CPHA0_CPOL0` = 0, `ROM_SPI_CLOCK_MODE0` = `ROM_SPI_CLOCK_CPHA0_CPOL0`,  
`ROM_SPI_CLOCK_CPHA1_CPOL0` = 1, `ROM_SPI_CLOCK_MODE1` = `ROM_SPI_CLOCK_CPHA1_CPOL0`,  
`ROM_SPI_CLOCK_CPHA0_CPOL1` = 2, `ROM_SPI_CLOCK_MODE2` = `ROM_SPI_CLOCK_CPHA0_CPOL1`,  
`ROM_SPI_CLOCK_CPHA1_CPOL1` = 3, `ROM_SPI_CLOCK_MODE3` = `ROM_SPI_CLOCK_CPHA1_CPOL1` }  
*SPI Clock Mode.*
- enum `SPI_CLOCK_MODE_T` {  
`SPI_CLOCK_CPHA0_CPOL0` = `SPI_CFG_CPOL_LO | SPI_CFG_CPHA_FIRST`, `SPI_CLOCK_MODE0` = `SPI_CLOCK_CPHA0_CPOL0`,  
`SPI_CLOCK_CPHA1_CPOL0` = `SPI_CFG_CPOL_LO | SPI_CFG_CPHA_SECOND`, `SPI_CLOCK_MODE1` = `SPI_CLOCK_CPHA1_CPOL0`,  
`SPI_CLOCK_CPHA0_CPOL1` = `SPI_CFG_CPOL_HI | SPI_CFG_CPHA_FIRST`, `SPI_CLOCK_MODE2` = `SPI_CLOCK_CPHA0_CPOL1`,  
`SPI_CLOCK_CPHA1_CPOL1` = `SPI_CFG_CPOL_HI | SPI_CFG_CPHA_SECOND`, `SPI_CLOCK_MODE3` = `SPI_CLOCK_CPHA1_CPOL1` }  
*SPI Clock Mode.*

## Functions

- int `Chip_SPI_Init` (`LPC_SPI_T` \*pSPI)  
*Initialize the SPI.*
- \_\_STATIC\_INLINE void `Chip_SPI_DeInit` (`LPC_SPI_T` \*pSPI)  
*Disable SPI operation.*
- \_\_STATIC\_INLINE void `Chip_SPI_SetCFGRegBits` (`LPC_SPI_T` \*pSPI, uint32\_t bits)  
*Set SPI CFG register values.*
- \_\_STATIC\_INLINE void `Chip_SPI_ClearCFGRegBits` (`LPC_SPI_T` \*pSPI, uint32\_t bits)  
*Clear SPI CFG register values.*
- \_\_STATIC\_INLINE void `Chip_SPI_Enable` (`LPC_SPI_T` \*pSPI)

- Enable SPI peripheral.*

  - `__STATIC_INLINE void Chip_SPI_Disable (LPC_SPI_T *pSPI)`
- Disable SPI peripheral.*

  - `__STATIC_INLINE void Chip_SPI_EnableSlaveMode (LPC_SPI_T *pSPI)`
- Enable SPI slave mode.*

  - `__STATIC_INLINE void Chip_SPI_EnableLSBFirst (LPC_SPI_T *pSPI)`
- Enable LSB First transfers.*

  - `__STATIC_INLINE void Chip_SPI_EnableMSBFirst (LPC_SPI_T *pSPI)`
- Enable MSB First transfers.*

  - `__STATIC_INLINE void Chip_SPI_SetSPIMode (LPC_SPI_T *pSPI, SPI_CLOCK_MODE_T mode)`
- Set SPI mode.*

  - `__STATIC_INLINE void Chip_SPI_SetCSPolHigh (LPC_SPI_T *pSPI, uint8_t csNum)`
- Set polarity on the SPI chip select high.*

  - `__STATIC_INLINE void Chip_SPI_SetCSPolLow (LPC_SPI_T *pSPI, uint8_t csNum)`
- Set polarity on the SPI chip select low.*

  - `void Chip_SPI_ConfigureSPI (LPC_SPI_T *pSPI, SPI_CFGSETUP_T *pCFG)`
- Setup SPI configuration.*

  - `__STATIC_INLINE uint32_t Chip_SPI_GetStatus (LPC_SPI_T *pSPI)`
- Get the current status of SPI controller.*

  - `__STATIC_INLINE void Chip_SPI_ClearStatus (LPC_SPI_T *pSPI, uint32_t Flag)`
- Clear SPI status.*

  - `__STATIC_INLINE void Chip_SPI_EnableInts (LPC_SPI_T *pSPI, uint32_t intMask)`
- Enable a SPI interrupt.*

  - `__STATIC_INLINE void Chip_SPI_DisableInts (LPC_SPI_T *pSPI, uint32_t intMask)`
- Disable a SPI interrupt.*

  - `__STATIC_INLINE uint32_t Chip_SPI_GetEnabledInts (LPC_SPI_T *pSPI)`
- Return enabled SPI interrupts.*

  - `__STATIC_INLINE uint32_t Chip_SPI_GetPendingInts (LPC_SPI_T *pSPI)`
- Return pending SPI interrupts.*

  - `__STATIC_INLINE void Chip_SPI_SetFIFOCfg (LPC_SPI_T *pSPI, uint32_t cfg)`
- Set FIFO Configuration register.*

  - `__STATIC_INLINE void Chip_SPI_ClearFIFOCfg (LPC_SPI_T *pSPI, uint32_t cfg)`
- Clear FIFO Configuration register.*

  - `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOStatus (LPC_SPI_T *pSPI)`
- Get the current status of SPI controller FIFO.*

  - `__STATIC_INLINE void Chip_SPI_ClearFIFOStatus (LPC_SPI_T *pSPI, uint32_t mask)`
- Clear the FIFO status register.*

  - `__STATIC_INLINE void Chip_SPI_SetFIFOTrigLevel (LPC_SPI_T *pSPI, uint8_t tx_lvl, uint8_t rx_lvl)`
- Setup SPI FIFO trigger-level.*

  - `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOTrigLevel (LPC_SPI_T *pSPI)`
- Get SPI FIFO trigger-level.*

  - `__STATIC_INLINE void Chip_SPI_EnableFIFOInts (LPC_SPI_T *pSPI, uint32_t intMask)`
- Enable a SPI FIFO interrupt.*

  - `__STATIC_INLINE void Chip_SPI_DisableFIFOInts (LPC_SPI_T *pSPI, uint32_t intMask)`
- Disable a SPI FIFO interrupt.*

  - `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOEnabledInts (LPC_SPI_T *pSPI)`
- Return enabled SPI FIFO interrupts.*

  - `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOPendingInts (LPC_SPI_T *pSPI)`
- Return pending SPI FIFO interrupts.*

  - `__STATIC_INLINE uint32_t Chip_SPI_ReadRawRXFifo (LPC_SPI_T *pSPI)`
- Read raw data from receive FIFO with status bits.*

- `__STATIC_INLINE uint16_t Chip_SPI_ReadRXData (LPC_SPI_T *pSPI)`  
*Read data from receive FIFO masking off status bits.*
- `__STATIC_INLINE void Chip_SPI_WriteFIFO (LPC_SPI_T *pSPI, uint32_t data)`  
*Write FIFOWR register: writes 32-bit value to FIFO.*
- `__STATIC_INLINE void Chip_SPI_SetTXCTRLData (LPC_SPI_T *pSPI, uint16_t ctrl, uint16_t data)`  
*Write FIFOWR register: writes control options and data.*
- `__STATIC_INLINE void Chip_SPI_WriteTXData (LPC_SPI_T *pSPI, uint16_t data)`  
*Write data to transmit FIFO.*
- `__STATIC_INLINE void Chip_SPI_FlushFifos (LPC_SPI_T *pSPI)`  
*Flush FIFOs.*

## 6.41.2 Macro Definition Documentation

### 6.41.2.1 `#define Chip_SPI_FlushFIFOs Chip_SPI_FlushFifos`

Definition at line 676 of file `spi_common_5411x.h`.

### 6.41.2.2 `#define Chip_SPI_ReadFIFO Chip_SPI_ReadRawRXFifo`

Definition at line 611 of file `spi_common_5411x.h`.

### 6.41.2.3 `#define Chip_SPI_ReadFIFOdata Chip_SPI_ReadRXData`

Definition at line 625 of file `spi_common_5411x.h`.

### 6.41.2.4 `#define Chip_SPI_WriteFIFOcd Chip_SPI_SetTXCTRLData`

Definition at line 652 of file `spi_common_5411x.h`.

### 6.41.2.5 `#define Chip_SPI_WriteFIFOdata Chip_SPI_WriteTXData`

Definition at line 665 of file `spi_common_5411x.h`.

### 6.41.2.6 `#define SPI_CFG_BITMASK (0x0FBD) /** SPI register bit mask */`

Macro defines for SPI Configuration register

Definition at line 92 of file `spi_common_5411x.h`.

### 6.41.2.7 `#define SPI_CFG_CPHA_FIRST (0 << 4) /** Capture data on the first edge, Change data on the following edge */`

Definition at line 98 of file `spi_common_5411x.h`.

### 6.41.2.8 `#define SPI_CFG_CPHA_SECOND (1 << 4) /** SPI Clock Polarity Select */`

Definition at line 99 of file `spi_common_5411x.h`.

### 6.41.2.9 `#define SPI_CFG_CPOL_HI (1 << 5) /** The rest state of the clock (between frames) is high. */`

Definition at line 101 of file `spi_common_5411x.h`.

6.41.2.10 **#define SPI\_CFG\_CPOL\_LO (0 << 5) /\*\* The rest state of the clock (between frames) is low. \*/**

Definition at line 100 of file spi\_common\_5411x.h.

6.41.2.11 **#define SPI\_CFG\_LBM\_EN (1 << 7) /\*\* SPI control 1 loopback mode enable \*/**

Definition at line 102 of file spi\_common\_5411x.h.

6.41.2.12 **#define SPI\_CFG\_LSB\_FIRST\_EN (1 << 3) /\*\* SPI Clock Phase Select \*/**

Definition at line 97 of file spi\_common\_5411x.h.

6.41.2.13 **#define SPI\_CFG\_MASTER\_EN (1 << 2) /\*\* SPI MSB First mode enable \*/**

Definition at line 95 of file spi\_common\_5411x.h.

6.41.2.14 **#define SPI\_CFG\_MSB\_FIRST\_EN (0 << 3) /\*\* SPI LSB First mode enable \*/**

Definition at line 96 of file spi\_common\_5411x.h.

6.41.2.15 **#define SPI\_CFG\_SLAVE\_EN (0 << 0) /\*\* SPI Master Mode Select \*/**

Definition at line 94 of file spi\_common\_5411x.h.

6.41.2.16 **#define SPI\_CFG\_SPI\_EN (1 << 0) /\*\* SPI Slave Mode Select \*/**

Definition at line 93 of file spi\_common\_5411x.h.

6.41.2.17 **#define SPI\_CFG\_SPOL\_HI (1 << 8) /\*\* SSEL0 is active High \*/**

Definition at line 104 of file spi\_common\_5411x.h.

6.41.2.18 **#define SPI\_CFG\_SPOL\_LO (0 << 8) /\*\* SPI SSEL0 Polarity Select \*/**

Definition at line 103 of file spi\_common\_5411x.h.

6.41.2.19 **#define SPI\_CFG\_SPOLNUM\_HI( n ) (1 << ((n) + 8)) /\*\* SSELN is active High, selects 0 - 3 \*/**

Definition at line 105 of file spi\_common\_5411x.h.

6.41.2.20 **#define SPI\_DIV\_VAL( n ) ((n) & 0xFFFF) /\*\* Rate divider value mask (In Master Mode only)\*/**

Macro defines for SPI Divider register

Definition at line 219 of file spi\_common\_5411x.h.

6.41.2.21 **#define SPI\_DLY\_BITMASK (0xFFFF) /\*\* SPI DLY Register Mask \*/**

Macro defines for SPI Delay register

Definition at line 110 of file spi\_common\_5411x.h.

**6.41.2.22** `#define SPI_DLY_FRAME_DELAY( n ) (((n) & 0x0F) << 8) /** Minimum time in SPI clocks between adjacent data frames. */`

Definition at line 113 of file `spi_common_5411x.h`.

**6.41.2.23** `#define SPI_DLY_POST_DELAY( n ) (((n) & 0x0F) << 4) /** Time in SPI clocks between the end of a data frame and SSEL deassertion. */`

Definition at line 112 of file `spi_common_5411x.h`.

**6.41.2.24** `#define SPI_DLY_PRE_DELAY( n ) (((n) & 0x0F) << 0) /** Time in SPI clocks between SSEL assertion and the beginning of a data frame */`

Definition at line 111 of file `spi_common_5411x.h`.

**6.41.2.25** `#define SPI_DLY_TRANSFER_DELAY( n ) (((n) & 0x0F) << 12) /** Minimum time in SPI clocks that the SSEL is deasserted between transfers. */`

Definition at line 114 of file `spi_common_5411x.h`.

**6.41.2.26** `#define SPI_FIFO_DEPTH(8) /** SPI-FIFO How many entries are in the FIFO */`

Macro defines for FIFO Status register.

Definition at line 151 of file `spi_common_5411x.h`.

**6.41.2.27** `#define SPI_FIFOCFG_DMARX(1 << 13)`

Enable DMA RX

Definition at line 142 of file `spi_common_5411x.h`.

**6.41.2.28** `#define SPI_FIFOCFG_DMATX(1 << 12)`

Enable DMA TX

Definition at line 141 of file `spi_common_5411x.h`.

**6.41.2.29** `#define SPI_FIFOCFG_EMPTYRX(1 << 17)`

Empty the RX FIFO

Definition at line 146 of file `spi_common_5411x.h`.

**6.41.2.30** `#define SPI_FIFOCFG_EMPTYTX(1 << 16)`

Empty the TX FIFO

Definition at line 145 of file `spi_common_5411x.h`.

**6.41.2.31** `#define SPI_FIFOCFG_ENABLERX(1 << 1)`

Enable RX FIFO

Definition at line 140 of file `spi_common_5411x.h`.



**6.41.2.32 #define SPI\_FIFOCFG\_ENABLETX (1 << 0)**

SPI FIFO Configuration register bits.

Enable TX FIFO

Definition at line 139 of file spi\_common\_5411x.h.

**6.41.2.33 #define SPI\_FIFOCFG\_WAKERX (1 << 15)**

Enable wakeup triggered by RX

Definition at line 144 of file spi\_common\_5411x.h.

**6.41.2.34 #define SPI\_FIFOCFG\_WAKETX (1 << 14)**

Enable wakeup triggered by TX

Definition at line 143 of file spi\_common\_5411x.h.

**6.41.2.35 #define SPI\_FIFPOINT\_BITMASK (0x001F) /\*\* FIFO interrupt Bit mask \*/**

Macro defines for SPI Interrupt enable/disable/status [INTSET/INTCLR/INTSTAT and FIFPOINTSET/FIFPOINTCLR/FIFPOINTSTAT registers].

Definition at line 178 of file spi\_common\_5411x.h.

**6.41.2.36 #define SPI\_FIFPOINT\_PERINT (1 << 4) /\*\* SPI peripheral interrupt [BIT-4 of FIFPOINTSTAT register] \*/**

Definition at line 183 of file spi\_common\_5411x.h.

**6.41.2.37 #define SPI\_FIFPOINT\_RXERR (1 << 1) /\*\* RX error interrupt [BIT-1 of FIFPOINTENSET/FIFPOINTENCLR/FIFPOINTSTAT register] \*/**

Definition at line 180 of file spi\_common\_5411x.h.

**6.41.2.38 #define SPI\_FIFPOINT\_RXLVL (1 << 3) /\*\* RX Data ready interrupt [BIT-3 of FIFPOINTENSET/FIFPOINTENCLR/FIFPOINTSTAT register] \*/**

Definition at line 182 of file spi\_common\_5411x.h.

**6.41.2.39 #define SPI\_FIFPOINT\_TXERR (1 << 0) /\*\* TX error interrupt [BIT-0 of FIFPOINTENSET/FIFPOINTENCLR/FIFPOINTSTAT register] \*/**

Definition at line 179 of file spi\_common\_5411x.h.

**6.41.2.40 #define SPI\_FIFPOINT\_TXLVL (1 << 2) /\*\* TX FIFO ready interrupt [BIT-2 of FIFPOINTENSET/FIFPOINTENCLR/FIFPOINTSTAT register] \*/**

Definition at line 181 of file spi\_common\_5411x.h.

**6.41.2.41 #define SPI\_FIFOSTAT\_BITMASK (0x1F1FFB) /\*\* SPI-FIFO STAT Register BitMask \*/**

Definition at line 152 of file spi\_common\_5411x.h.

**6.41.2.42** `#define SPI_FIFOSTAT_PERINT (1 << 3) /** SPI-FIFO peripheral (SPI) interrupt */`

Definition at line 155 of file spi\_common\_5411x.h.

**6.41.2.43** `#define SPI_FIFOSTAT_RXERR (1 << 1) /** SPI-FIFO receive error */`

Definition at line 154 of file spi\_common\_5411x.h.

**6.41.2.44** `#define SPI_FIFOSTAT_RXFULL (1 << 7) /** SPI-FIFO receiver not full */`

Definition at line 159 of file spi\_common\_5411x.h.

**6.41.2.45** `#define SPI_FIFOSTAT_RXLVL( val ) (((val) >> 16) & 0x1F) /** SPI-FIFO extract receive level */`

Definition at line 161 of file spi\_common\_5411x.h.

**6.41.2.46** `#define SPI_FIFOSTAT_RXNOTEMPTY (1 << 6) /** SPI-FIFO receiver not empty */`

Definition at line 158 of file spi\_common\_5411x.h.

**6.41.2.47** `#define SPI_FIFOSTAT_TXEMPTY (1 << 4) /** SPI-FIFO transmitter empty */`

Definition at line 156 of file spi\_common\_5411x.h.

**6.41.2.48** `#define SPI_FIFOSTAT_TXERR (1 << 0) /** SPI-FIFO transmit error */`

Definition at line 153 of file spi\_common\_5411x.h.

**6.41.2.49** `#define SPI_FIFOSTAT_TXLVL( val ) (((val) >> 8) & 0x1F) /** SPI-FIFO extract transmit level */`

Definition at line 160 of file spi\_common\_5411x.h.

**6.41.2.50** `#define SPI_FIFOSTAT_TXNOTFULL (1 << 5) /** SPI-FIFO transmitter not full */`

Definition at line 157 of file spi\_common\_5411x.h.

**6.41.2.51** `#define SPI_FIFOTRIG_BITMASK (0x000f0f03) /** SPI FIFO trigger settings Register BitMask */`

UART FIFO trigger settings register defines.

Definition at line 166 of file spi\_common\_5411x.h.

**6.41.2.52** `#define SPI_FIFOTRIG_RXLVL( lvl ) ((lvl & 0x0f) << 16)`

Set RX Level trigger

Definition at line 170 of file spi\_common\_5411x.h.

**6.41.2.53 #define SPI\_FIFOTRIG\_RXLVL\_DEFAULT 0**

Set RX default trigger level to one item

Definition at line 173 of file spi\_common\_5411x.h.

**6.41.2.54 #define SPI\_FIFOTRIG\_RXLVLENA (1 << 1)**

RX level enable

Definition at line 168 of file spi\_common\_5411x.h.

**6.41.2.55 #define SPI\_FIFOTRIG\_TXLVL( lvl ) ((lvl & 0x0f) << 8)**

Set TX Level trigger

Definition at line 169 of file spi\_common\_5411x.h.

**6.41.2.56 #define SPI\_FIFOTRIG\_TXLVL\_DEFAULT 4**

Set TX default trigger level to half-full

Definition at line 172 of file spi\_common\_5411x.h.

**6.41.2.57 #define SPI\_FIFOTRIG\_TXLVLENA (1 << 0)**

TX level enable

Definition at line 167 of file spi\_common\_5411x.h.

**6.41.2.58 #define SPI\_INT\_BITMASK (0x0130) /\*\* SPI interrupt Enable/Disable bits \*/**

Macro defines for SPI interrupt register

Definition at line 129 of file spi\_common\_5411x.h.

**6.41.2.59 #define SPI\_INT\_MSTIDLE (1 << 8) /\*\* SPI master is Idle [BIT-8 of INTENSET/INTENCLR/INTSTAT register] \*/**

Definition at line 132 of file spi\_common\_5411x.h.

**6.41.2.60 #define SPI\_INT\_SSAEN (1 << 4) /\*\* Slave Select is asserted interrupt [BIT-4 of INTENSET/INTENCLR/INTSTAT register] \*/**

Definition at line 130 of file spi\_common\_5411x.h.

**6.41.2.61 #define SPI\_INT\_SSDEN (1 << 5) /\*\* Slave Select is deasserted interrupt [BIT-5 of INTENSET/INTENCLR/INTSTAT register] \*/**

Definition at line 131 of file spi\_common\_5411x.h.

**6.41.2.62 #define SPI\_RXDAT\_BITMASK (0x1FFFFFF) /\*\* SPI RXDAT Register BitMask \*/**

Macro defines for SPI FIFO Receiver Data register

Definition at line 188 of file spi\_common\_5411x.h.

**6.41.2.63** `#define SPI_RXDAT_DATA( n ) ((n) & 0xFFFF) /** Receiver Data */`

Definition at line 189 of file spi\_common\_5411x.h.

**6.41.2.64** `#define SPI_RXDAT_RXSSELN( n ) ((~((n >> 16) & 0x0f)) & 0x0f) /** Determine the active SSEL pin */`

Definition at line 190 of file spi\_common\_5411x.h.

**6.41.2.65** `#define SPI_RXDAT_RXSSELN_ACTIVE (0 << 16) /** The state of SSEL pin is active */`

Definition at line 191 of file spi\_common\_5411x.h.

**6.41.2.66** `#define SPI_RXDAT_SOT (1 << 20) /** Start of Transfer flag */`

Definition at line 192 of file spi\_common\_5411x.h.

**6.41.2.67** `#define SPI_STAT_BITMASK (0x01F0) /** SPI STAT Register BitMask */`

Macro defines for SPI Status register

Definition at line 119 of file spi\_common\_5411x.h.

**6.41.2.68** `#define SPI_STAT_EOT (1 << 7) /** End Transfer flag */`

Definition at line 123 of file spi\_common\_5411x.h.

**6.41.2.69** `#define SPI_STAT_MSTIDLE (1 << 8) /** Idle status flag */`

Definition at line 124 of file spi\_common\_5411x.h.

**6.41.2.70** `#define SPI_STAT_SSA (1 << 4) /** Slave Select Assert */`

Definition at line 120 of file spi\_common\_5411x.h.

**6.41.2.71** `#define SPI_STAT_SSD (1 << 5) /** Slave Select Deassert */`

Definition at line 121 of file spi\_common\_5411x.h.

**6.41.2.72** `#define SPI_STAT_STALLED (1 << 6) /** Stalled status flag */`

Definition at line 122 of file spi\_common\_5411x.h.

**6.41.2.73** `#define SPI_TXDAT_ASSERT_SSEL (0) /** Assert SSEL0 pin */`

Definition at line 200 of file spi\_common\_5411x.h.

**6.41.2.74** `#define SPI_TXDAT_ASSERTNUM_SSEL( n ) ((~(1 << (n))) & 0x0f) /** Assert SSELN pin */`

Definition at line 201 of file spi\_common\_5411x.h.

**6.41.2.75** `#define SPI_TXDAT_BITMASK (0xF7FFFF) /** SPI TXDATCTL Register BitMask */`

Macro defines for SPI FIFO Transmitter Data and Control register

Definition at line 197 of file spi\_common\_5411x.h.

**6.41.2.76** `#define SPI_TXDAT_CTRLMASK (0xF7F) /** SPI TXDATCTL Register BitMask for control bits only */`

Definition at line 199 of file spi\_common\_5411x.h.

**6.41.2.77** `#define SPI_TXDAT_DATA( n ) ((n) & 0xFFFF) /** SPI Transmit Data */`

Macro defines for SPI Transmitter Data Register

Definition at line 214 of file spi\_common\_5411x.h.

**6.41.2.78** `#define SPI_TXDAT_DATA( n ) ((n) & 0xFFFF) /** SPI Transmit Data */`

Macro defines for SPI Transmitter Data Register

Definition at line 214 of file spi\_common\_5411x.h.

**6.41.2.79** `#define SPI_TXDAT_DEASSERT_ALL (0xF) /** Deassert all SSEL pins */`

Definition at line 204 of file spi\_common\_5411x.h.

**6.41.2.80** `#define SPI_TXDAT_DEASSERT_SSEL (1) /** Deassert SSEL0 pin */`

Definition at line 202 of file spi\_common\_5411x.h.

**6.41.2.81** `#define SPI_TXDAT_DEASSERTNUM_SSEL( n ) (1 << (n)) /** Deassert SSELN pin */`

Definition at line 203 of file spi\_common\_5411x.h.

**6.41.2.82** `#define SPI_TXDAT_EOF (1 << 5) /** End of Frame flag (FRAME_DELAY is applied after sending the current part) */`

Definition at line 206 of file spi\_common\_5411x.h.

**6.41.2.83** `#define SPI_TXDAT_EOT (1 << 4) /** End of Transfer flag (TRANSFER_DELAY is applied after sending the current frame) */`

Definition at line 205 of file spi\_common\_5411x.h.

**6.41.2.84** `#define SPI_TXDAT_FLEN( n ) (((n) & 0x0F) << 8) /** Frame length - 1 */`

Definition at line 208 of file spi\_common\_5411x.h.

**6.41.2.85** `#define SPI_TXDAT_FLENMASK (0xF << 8) /** Frame length mask */`

Definition at line 209 of file spi\_common\_5411x.h.

6.41.2.86 `#define SPI_TXDAT_RXIGNORE (1 << 6) /** Receive Ignore Flag */`

Definition at line 207 of file `spi_common_5411x.h`.

### 6.41.3 Enumeration Type Documentation

6.41.3.1 `enum ROM_SPI_CLOCK_MODE_T`

SPI Clock Mode.

Enumerator

***ROM\_SPI\_CLOCK\_CPHA0\_CPOL0*** CPHA = 0, CPOL = 0  
***ROM\_SPI\_CLOCK\_MODE0*** Alias for CPHA = 0, CPOL = 0  
***ROM\_SPI\_CLOCK\_CPHA1\_CPOL0*** CPHA = 0, CPOL = 1  
***ROM\_SPI\_CLOCK\_MODE1*** Alias for CPHA = 0, CPOL = 1  
***ROM\_SPI\_CLOCK\_CPHA0\_CPOL1*** CPHA = 1, CPOL = 0  
***ROM\_SPI\_CLOCK\_MODE2*** Alias for CPHA = 1, CPOL = 0  
***ROM\_SPI\_CLOCK\_CPHA1\_CPOL1*** CPHA = 1, CPOL = 1  
***ROM\_SPI\_CLOCK\_MODE3*** Alias for CPHA = 1, CPOL = 1

Definition at line 222 of file `spi_common_5411x.h`.

6.41.3.2 `enum SPI_CLOCK_MODE_T`

SPI Clock Mode.

Enumerator

***SPI\_CLOCK\_CPHA0\_CPOL0*** CPHA = 0, CPOL = 0  
***SPI\_CLOCK\_MODE0*** Alias for CPHA = 0, CPOL = 0  
***SPI\_CLOCK\_CPHA1\_CPOL0*** CPHA = 0, CPOL = 1  
***SPI\_CLOCK\_MODE1*** Alias for CPHA = 0, CPOL = 1  
***SPI\_CLOCK\_CPHA0\_CPOL1*** CPHA = 1, CPOL = 0  
***SPI\_CLOCK\_MODE2*** Alias for CPHA = 1, CPOL = 0  
***SPI\_CLOCK\_CPHA1\_CPOL1*** CPHA = 1, CPOL = 1  
***SPI\_CLOCK\_MODE3*** Alias for CPHA = 1, CPOL = 1

Definition at line 329 of file `spi_common_5411x.h`.

### 6.41.4 Function Documentation

6.41.4.1 `__STATIC_INLINE void Chip_SPI_ClearCFGRegBits ( LPC_SPI_T * pSPI, uint32_t bits )`

Clear SPI CFG register values.

Parameters

---

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>bits</i>	: CFG register bits to clear, and OR'ed value of SPI_CFG_* definitions

**Returns**

Nothing

**Note**

This function safely clears only the selected bits in the SPI CFG register. It can be used to disable multiple bits at once.

Definition at line 272 of file spi\_common\_5411x.h.

#### 6.41.4.2 `__STATIC_INLINE void Chip_SPI_ClearFIFOCfg ( LPC_SPI_T * pSPI, uint32_t cfg )`

Clear FIFO Configuration register.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>cfg</i>	: Configuration value mask (OR'ed SPI_FIFOCFG_* values like <a href="#">SPI_FIFOCFG_ENABLETX</a> )

**Returns**

Nothing

Definition at line 493 of file spi\_common\_5411x.h.

#### 6.41.4.3 `__STATIC_INLINE void Chip_SPI_ClearFIFOStatus ( LPC_SPI_T * pSPI, uint32_t mask )`

Clear the FIFO status register.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>mask</i>	: Mask of the status bits that needs to be cleared

**Returns**

Nothing

Definition at line 519 of file spi\_common\_5411x.h.

#### 6.41.4.4 `__STATIC_INLINE void Chip_SPI_ClearStatus ( LPC_SPI_T * pSPI, uint32_t Flag )`

Clear SPI status.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>Flag</i>	: Clear Flag (Or-ed bit value of SPI_STAT_*)

**Returns**

Nothing

Definition at line 411 of file spi\_common\_5411x.h.

6.41.4.5 void Chip\_SPI\_ConfigureSPI ( LPC\_SPI\_T \* *pSPI*, SPI\_CFGSETUP\_T \* *pCFG* )

Setup SPI configuration.



## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>pCFG</i>	: Pointer to SPI configuration structure

## Returns

Nothing

Definition at line 63 of file spi\_common\_5411x.c.

6.41.4.6 `__STATIC_INLINE void Chip_SPI_DeInit ( LPC_SPI_T * pSPI )`

Disable SPI operation.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

## Returns

Nothing

## Note

The SPI controller is disabled.

Definition at line 246 of file spi\_common\_5411x.h.

6.41.4.7 `__STATIC_INLINE void Chip_SPI_Disable ( LPC_SPI_T * pSPI )`

Disable SPI peripheral.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

## Returns

Nothing

Definition at line 292 of file spi\_common\_5411x.h.

6.41.4.8 `__STATIC_INLINE void Chip_SPI_DisableFIFOInts ( LPC_SPI_T * pSPI, uint32_t intMask )`

Disable a SPI FIFO interrupt.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>intMask</i>	: Or'ed value of SPI_FIFOINT_* values to disable (See <a href="#">SPI_FIFOINT_BITMASK</a> )

## Returns

Nothing

Definition at line 574 of file spi\_common\_5411x.h.

6.41.4.9 `__STATIC_INLINE void Chip_SPI_DisableInts ( LPC_SPI_T * pSPI, uint32_t intMask )`

Disable a SPI interrupt.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>intMask</i>	: Or'ed value of SPI_INT_* values to disable (See <a href="#">SPI_INT_BITMASK</a> )

**Returns**

Nothing

Definition at line 433 of file spi\_common\_5411x.h.

#### 6.41.4.10 `__STATIC_INLINE void Chip_SPI_Enable ( LPC_SPI_T * pSPI )`

Enable SPI peripheral.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

**Returns**

Nothing

Definition at line 282 of file spi\_common\_5411x.h.

#### 6.41.4.11 `__STATIC_INLINE void Chip_SPI_EnableFIFOInts ( LPC_SPI_T * pSPI, uint32_t intMask )`

Enable a SPI FIFO interrupt.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>intMask</i>	: Or'ed value of SPI_FIFOINT_* values to enable

**Returns**

Nothing

Definition at line 563 of file spi\_common\_5411x.h.

#### 6.41.4.12 `__STATIC_INLINE void Chip_SPI_EnableInts ( LPC_SPI_T * pSPI, uint32_t intMask )`

Enable a SPI interrupt.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>intMask</i>	: Or'ed value of SPI_INT_* values to enable

**Returns**

Nothing

Definition at line 422 of file spi\_common\_5411x.h.

#### 6.41.4.13 `__STATIC_INLINE void Chip_SPI_EnableLSBFirst ( LPC_SPI_T * pSPI )`

Enable LSB First transfers.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

## Returns

Nothing

Definition at line 313 of file spi\_common\_5411x.h.

#### 6.41.4.14 \_\_STATIC\_INLINE void Chip\_SPI\_EnableMSBFirst ( LPC\_SPI\_T \* *pSPI* )

Enable MSB First transfers.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

## Returns

Nothing

Definition at line 323 of file spi\_common\_5411x.h.

#### 6.41.4.15 \_\_STATIC\_INLINE void Chip\_SPI\_EnableSlaveMode ( LPC\_SPI\_T \* *pSPI* )

Enable SPI slave mode.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

## Returns

Nothing

## Note

SPI master mode will be disabled with this call.

Definition at line 303 of file spi\_common\_5411x.h.

#### 6.41.4.16 \_\_STATIC\_INLINE void Chip\_SPI\_FlushFifos ( LPC\_SPI\_T \* *pSPI* )

Flush FIFOs.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

## Returns

Nothing

Definition at line 672 of file spi\_common\_5411x.h.

#### 6.41.4.17 \_\_STATIC\_INLINE uint32\_t Chip\_SPI\_GetEnabledInts ( LPC\_SPI\_T \* *pSPI* )

Return enabled SPI interrupts.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

**Returns**

An Or'ed value of SPI\_INT\_\* values

**Note**

Mask the return value with a SPI\_INT\_\* value to determine (See [SPI\\_INT\\_BITMASK](#)) if the interrupt is enabled.

INTSTAT contains enabled interrupts, not pending interrupts.

Definition at line 446 of file spi\_common\_5411x.h.

6.41.4.18 `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOEnabledInts ( LPC_SPI_T * pSPI )`

Return enabled SPI FIFO interrupts.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

**Returns**

An Or'ed value of SPI\_FIFoint\_\* values

Definition at line 584 of file spi\_common\_5411x.h.

6.41.4.19 `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOPendingInts ( LPC_SPI_T * pSPI )`

Return pending SPI FIFO interrupts.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

**Returns**

An Or'ed value of SPI\_FIFoint\_\* values

Definition at line 594 of file spi\_common\_5411x.h.

6.41.4.20 `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOStatus ( LPC_SPI_T * pSPI )`

Get the current status of SPI controller FIFO.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

**Returns**

SPI Status (Or-ed bit value of SPI\_FIFostat\_\*)

**Note**

Mask the return value with a value of type SPI\_FIFostat\_\* to determine if that status is active.

Definition at line 508 of file spi\_common\_5411x.h.

6.41.4.21 `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOTrigLevel ( LPC_SPI_T * pSPI )`

Get SPI FIFO trigger-level.

**Parameters**

<i>pSPI</i>	: Base of on-chip SPI peripheral
-------------	----------------------------------

**Returns**

Returns the complete raw trigger register.

Definition at line 548 of file spi\_common\_5411x.h.

**6.41.4.22** `__STATIC_INLINE uint32_t Chip_SPI_GetPendingInts ( LPC_SPI_T * pSPI )`

Return pending SPI interrupts.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

**Returns**

An Or'ed value of SPI\_INT\_\* values

Definition at line 456 of file spi\_common\_5411x.h.

**6.41.4.23** `__STATIC_INLINE uint32_t Chip_SPI_GetStatus ( LPC_SPI_T * pSPI )`

Get the current status of SPI controller.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

**Returns**

SPI Status (Or-ed bit value of SPI\_STAT\_\*)

**Note**

Mask the return value with a value of type SPI\_STAT\_\* to determine if that status is active.

Definition at line 400 of file spi\_common\_5411x.h.

**6.41.4.24** `int Chip_SPI_Init ( LPC_SPI_T * pSPI )`

Initialize the SPI.

**Parameters**

<i>pSPI</i>	: The base SPI peripheral on the chip
-------------	---------------------------------------

**Returns**

Nothing

Definition at line 50 of file spi\_common\_5411x.c.

**6.41.4.25** `__STATIC_INLINE uint32_t Chip_SPI_ReadRawRXFifo ( LPC_SPI_T * pSPI )`

Read raw data from receive FIFO with status bits.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

## Returns

Current value in receive data FIFO plus status bits

Definition at line 607 of file spi\_common\_5411x.h.

#### 6.41.4.26 `__STATIC_INLINE uint16_t Chip_SPI_ReadRXData ( LPC_SPI_T * pSPI )`

Read data from receive FIFO masking off status bits.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
-------------	---------------------------------------

## Returns

Current value in receive data FIFO

## Note

The entire register is read, but only the low 16-bits are returned.

Definition at line 620 of file spi\_common\_5411x.h.

#### 6.41.4.27 `__STATIC_INLINE void Chip_SPI_SetCFGRegBits ( LPC_SPI_T * pSPI, uint32_t bits )`

Set SPI CFG register values.

## Parameters

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>bits</i>	: CFG register bits to set, and OR'ed value of SPI_CFG_* definitions

## Returns

Nothing

## Note

This function safely sets only the selected bits in the SPI CFG register. It can be used to enable multiple bits at once.

Definition at line 259 of file spi\_common\_5411x.h.

#### 6.41.4.28 `__STATIC_INLINE void Chip_SPI_SetCSPolHigh ( LPC_SPI_T * pSPI, uint8_t csNum )`

Set polarity on the SPI chip select high.

## Parameters

---

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>csNum</i>	: Chip select number, 0 - 3

**Returns**

Nothing

**Note**

SPI chip select polarity is active high.

Definition at line 359 of file spi\_common\_5411x.h.

#### 6.41.4.29 `__STATIC_INLINE void Chip_SPI_SetCSPolLow ( LPC_SPI_T * pSPI, uint8_t csNum )`

Set polarity on the SPI chip select low.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>csNum</i>	: Chip select number, 0 - 3

**Returns**

Nothing

**Note**

SPI chip select polarity is active low.

Definition at line 371 of file spi\_common\_5411x.h.

#### 6.41.4.30 `__STATIC_INLINE void Chip_SPI_SetFIFOCfg ( LPC_SPI_T * pSPI, uint32_t cfg )`

Set FIFO Configuration register.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>cfg</i>	: Configuration value mask (OR'ed SPI_FIFOCFG_* values like <a href="#">SPI_FIFOCFG_ENABLETX</a> )

**Returns**

Nothing

Definition at line 482 of file spi\_common\_5411x.h.

#### 6.41.4.31 `__STATIC_INLINE void Chip_SPI_SetFIFOTrigLevel ( LPC_SPI_T * pSPI, uint8_t tx_lvl, uint8_t rx_lvl )`

Setup SPI FIFO trigger-level.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>tx_lvl</i>	: TX Trigger level [Valid values 0 to 7]
<i>rx_lvl</i>	: RX Trigger level [Valid values 0 to 7]



**Returns**

Nothing

**Note**

When *tx\_lvl* = 0; trigger will happen when TX FIFO is empty if *tx\_lvl* = 7; trigger will happen when TX FIFO has at least one free space

When *rx\_lvl* = 0; trigger will happen when RX FIFO has at least one data in it, if *rx\_lvl* = 7; trigger will happen when RX FIFO is full and cannot receive anymore data.

Definition at line 538 of file `spi_common_5411x.h`.

6.41.4.32 `__STATIC_INLINE void Chip_SPI_SetSPIMode ( LPC_SPI_T * pSPI, SPI_CLOCK_MODE_T mode )`

Set SPI mode.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>mode</i>	: SPI mode to set the SPI interface to

**Returns**

Nothing

Definition at line 346 of file `spi_common_5411x.h`.

6.41.4.33 `__STATIC_INLINE void Chip_SPI_SetTXCTRLData ( LPC_SPI_T * pSPI, uint16_t ctrl, uint16_t data )`

Write FIFOWR register: writes control options and data.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>ctrl</i>	: Control bits to be set
<i>data</i>	: Data to be transfered

**Returns**

Nothing

**Note**

This function safely sets only set bits in FIFOWR control register. It can be used to enable multiple options at once.

Definition at line 648 of file `spi_common_5411x.h`.

6.41.4.34 `__STATIC_INLINE void Chip_SPI_WriteFIFO ( LPC_SPI_T * pSPI, uint32_t data )`

Write FIFOWR register: writes 32-bit value to FIFO.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>data</i>	: Control and Data to be transfered

**Returns**

Nothing

Definition at line 634 of file spi\_common\_5411x.h.

6.41.4.35 `__STATIC_INLINE void Chip_SPI_WriteTXData ( LPC_SPI_T * pSPI, uint16_t data )`

Write data to transmit FIFO.

**Parameters**

<i>pSPI</i>	: Base on-chip SPI peripheral address
<i>data</i>	: Data to write

**Returns**

Nothing

Definition at line 660 of file spi\_common\_5411x.h.

## 6.42 CHIP: LPC5411X SPI master driver

### 6.42.1 Detailed Description

#### Data Structures

- struct [SPIM\\_DELAY\\_CONFIG\\_T](#)  
*SPI Delay Configure Struct.*
- struct [SPIM\\_XFER\\_T](#)  
*SPI Master transfer data context.*

#### Macros

- #define [SPIM\\_XFER\\_OPT\\_FRAME\\_DLY](#) (1 << 0) /\* frame delay between frames \*/
- #define [SPIM\\_XFER\\_OPT\\_FRAME\\_ASSERT](#) (1 << 1) /\* assert/de-assert ssel for each frame \*/
- #define [SPIM\\_XFER\\_OPT\\_DMA](#) (1 << 2) /\* use DMA \*/

#### Enumerations

- enum [SPIM\\_EVENT\\_T](#) {  
  [SPIM\\_EVENT\\_WAIT](#), [SPIM\\_EVENT\\_ERRORTX](#), [SPIM\\_EVENT\\_ERRORRX](#), [SPIM\\_EVENT\\_ERROR](#),  
  [SPIM\\_EVENT\\_DONE](#) }  
*SPI master Xfer events.*
- enum [SPIM\\_XFER\\_STATE\\_T](#) {  
  [SPI\\_XFER\\_STATE\\_IDLE](#), [SPI\\_XFER\\_STATE\\_BUSY](#), [SPI\\_XFER\\_STATE\\_DONE](#), [SPI\\_XFER\\_STATE\\_S←TALL](#),  
  [SPI\\_XFER\\_STATE\\_ERROR](#) }  
*States of SPI Master Xfer.*

#### Functions

- `__STATIC_INLINE uint32_t Chip_SPIM_GetClockRate (LPC_SPI_T *pSPI)`  
*Get SPI master bit rate.*
- `uint32_t Chip_SPIM_SetClockRate (LPC_SPI_T *pSPI, uint32_t rate)`  
*Set SPI master bit rate.*
- `void Chip_SPIM_DelayConfig (LPC_SPI_T *pSPI, SPIM_DELAY_CONFIG_T *pConfig)`  
*Config SPI Delay parameters.*
- `__STATIC_INLINE void Chip_SPIM_ForceEndOfTransfer (LPC_SPI_T *pSPI)`  
*Forces an end of transfer for the current master transfer.*
- `__STATIC_INLINE void Chip_SPIM_EnableLoopBack (LPC_SPI_T *pSPI)`  
*Enable loopback mode.*
- `__STATIC_INLINE void Chip_SPIM_DisableLoopBack (LPC_SPI_T *pSPI)`  
*Disable loopback mode.*
- `void Chip_SPIM_XferHandler (LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)`  
*SPI master transfer state change handler.*
- `void Chip_SPIM_Xfer (LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)`  
*Start non-blocking SPI master transfer.*
- `void Chip_SPIM_XferFIFO (LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)`  
*Start polled SPI master transfer.*
- `void Chip_SPIM_XferBlocking (LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)`  
*Perform blocking SPI master transfer.*

## 6.42.2 Macro Definition Documentation

### 6.42.2.1 `#define SPIM_XFER_OPT_DMA (1 << 2) /* use DMA */`

Definition at line 147 of file `spim_5411x.h`.

### 6.42.2.2 `#define SPIM_XFER_OPT_FRAME_ASSERT (1 << 1) /* assert/de-assert ssel for each frame */`

Definition at line 146 of file `spim_5411x.h`.

### 6.42.2.3 `#define SPIM_XFER_OPT_FRAME_DLY (1 << 0) /* frame delay between frames */`

Definition at line 145 of file `spim_5411x.h`.

## 6.42.3 Enumeration Type Documentation

### 6.42.3.1 `enum SPIM_EVENT_T`

SPI master Xfer events.

Enumerator

***SPIM\_EVENT\_WAIT*** SPI Xfer is waiting to be complete  
***SPIM\_EVENT\_ERRORTX*** SPI TX underflow error  
***SPIM\_EVENT\_ERRORRX*** SPI RX overflow error  
***SPIM\_EVENT\_ERROR*** SPI Xfer ended with an error!  
***SPIM\_EVENT\_DONE*** SPI Xfer completed without errors

Definition at line 126 of file `spim_5411x.h`.

### 6.42.3.2 `enum SPIM_XFER_STATE_T`

States of SPI Master Xfer.

Enumerator

***SPI\_XFER\_STATE\_IDLE*** Transfer is idle and not started  
***SPI\_XFER\_STATE\_BUSY*** Transfer has started and in progress  
***SPI\_XFER\_STATE\_DONE*** Transfer is complete without errors  
***SPI\_XFER\_STATE\_STALL*** SPI Bus stalled  
***SPI\_XFER\_STATE\_ERROR*** SPI transfer terminated with errors

Definition at line 137 of file `spim_5411x.h`.

## 6.42.4 Function Documentation

### 6.42.4.1 `void Chip_SPIM_DelayConfig ( LPC_SPI_T * pSPI, SPIM_DELAY_CONFIG_T * pConfig )`

Config SPI Delay parameters.

## Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
<i>pConfig</i>	: SPI Delay Configure Struct (See <a href="#">SPIM_DELAY_CONFIG_T</a> )

## Returns

Nothing

Definition at line 152 of file spim\_5411x.c.

#### 6.42.4.2 `__STATIC_INLINE void Chip_SPIM_DisableLoopBack ( LPC_SPI_T * pSPI )`

Disable loopback mode.

## Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
-------------	--

## Returns

Nothing

Definition at line 116 of file spim\_5411x.h.

#### 6.42.4.3 `__STATIC_INLINE void Chip_SPIM_EnableLoopBack ( LPC_SPI_T * pSPI )`

Enable loopback mode.

## Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
-------------	--

## Returns

Nothing

## Note

Serial input is taken from the serial output (MOSI or MISO) rather than the serial input pin.

Definition at line 106 of file spim\_5411x.h.

#### 6.42.4.4 `__STATIC_INLINE void Chip_SPIM_ForceEndOfTransfer ( LPC_SPI_T * pSPI )`

Forces an end of transfer for the current master transfer.

## Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
-------------	--

## Returns

Nothing

## Note

Use this function to perform an immediate end of transfer for the current master operation. If the master is currently transferring data started with the `Chip_SPIM_Xfer` function, this terminates the transfer after the current byte completes and completes the transfer.

Definition at line 94 of file spim\_5411x.h.

#### 6.42.4.5 `__STATIC_INLINE uint32_t Chip_SPIM_GetClockRate ( LPC_SPI_T * pSPI )`

Get SPI master bit rate.

##### Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
-------------	--

##### Returns

The actual SPI clock bit rate

Definition at line 51 of file spim\_5411x.h.

#### 6.42.4.6 `uint32_t Chip_SPIM_SetClockRate ( LPC_SPI_T * pSPI, uint32_t rate )`

Set SPI master bit rate.

##### Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
<i>rate</i>	: Desired clock bit rate for the SPI interface

##### Returns

The actual SPI clock bit rate

##### Note

This function will set the SPI clock divider to get closest to the desired rate as possible.

Definition at line 125 of file spim\_5411x.c.

#### 6.42.4.7 `void Chip_SPIM_Xfer ( LPC_SPI_T * pSPI, SPIM_XFER_T * xfer )`

Start non-blocking SPI master transfer.

##### Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
<i>xfer</i>	: Pointer to a <a href="#">SPIM_XFER_T</a> structure see notes below

##### Returns

Nothing

##### Note

This function starts a non-blocking SPI master transfer with the parameters setup in the passed [SPIM\\_XFER\\_T](#) structure. Once the transfer is started, the interrupt handler must call `Chip_SPIM_XferHandler` to keep the transfer going and fed with data. This function should only be called when the master is idle.

This function must be called with the options and sselNum fields correctly setup. Initial data buffers and the callback pointer must also be setup. No sanity checks are performed on the passed data.

Example call:

```
SPIM\_XFER\_T mxfer; mxfer.pCB = (&)masterCallbacks; mxfer.sselNum = 2; // Use chip select 2 mxfer.options =
```

```

SPI_TXDAT_FLEN(8); // 8 data bits, supports 1 - 16 bits
mxfer.options |= SPI_TXDAT_EOT | SPI_TXDAT_EOF;
// Apply frame and transfer delays to master transfer
mxfer.options |= SPI_TXDAT_RXIGNORE; // Ignore RX data,
// will toss receive data regardless of pRXData8 or pRXData16 buffer
mxfer.pTXData8 = SendBuffer; mxfer.txCount = 16;
// Number of bytes to send before SPIMasterXferSend callback is called
mxfer.pRXData8 = RecvBuffer; // Will not receive data if pRXData8/pRXData16 is NULL or SPI_TXDAT_RXIGNORE option is set
mxfer.rxCount = 16; // Number of bytes to receive before SPIMasterXferRecv callback is called
Chip_SPIM_Xfer(LPC_SPI0, &mxfer); // Start transfer

```

Note that the transfer, once started, needs to be constantly fed by the callbacks. The txCount and rxCount field only indicate the buffer size before the callbacks are called. To terminate the transfer, the SPIMasterXferSend callback must set the terminate field.

Definition at line 209 of file spim\_5411x.c.

#### 6.42.4.8 void Chip\_SPIM\_XferBlocking ( LPC\_SPI\_T \* pSPI, SPIM\_XFER\_T \* xfer )

Perform blocking SPI master transfer.

##### Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
<i>xfer</i>	: Pointer to a <a href="#">SPIM_XFER_T</a> structure see notes below

##### Returns

Nothing

##### Note

This function starts a blocking SPI master transfer with the parameters setup in the passed [SPIM\\_XFER\\_T](#) structure. Once the transfer is started, the callbacks in Chip\_SPIM\_XferHandler may be called to keep the transfer going and fed with data. SPI interrupts must be disabled prior to calling this function. It is not recommended to use this function.

Definition at line 275 of file spim\_5411x.c.

#### 6.42.4.9 void Chip\_SPIM\_XferFIFO ( LPC\_SPI\_T \* pSPI, SPIM\_XFER\_T \* xfer )

Start polled SPI master transfer.

##### Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
<i>xfer</i>	: Pointer to a <a href="#">SPIM_XFER_T</a> structure see notes below

##### Returns

Nothing

##### Note

This function starts a polled SPI master transfer with the parameters setup in the passed [SPIM\\_XFER\\_T](#) structure. This function should only be called when the master is idle.

This function forces the xfer counts to be smaller than the FIFO size. For small transfers, this is very efficient as there are no interrupts or DMA.

This function must be called with the options and sselNum fields set correctly Initial data buffers. The xfer transmit buffer is limited to the size of the FIFO.

Definition at line 240 of file spim\_5411x.c.

6.42.4.10 void Chip\_SPIM\_XferHandler ( LPC\_SPI\_T \* *pSPI*, SPIM\_XFER\_T \* *xfer* )

SPI master transfer state change handler.

#### Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
<i>xfer</i>	: Pointer to a <a href="#">SPIM_XFER_T</a> structure see notes below

#### Returns

Nothing

#### Note

See [SPIM\\_XFER\\_T](#) for more information on this function. When using this function, the SPI master interrupts should be enabled and setup in the SPI interrupt handler to call this function when they fire. This function is meant to be called from the interrupt handler.

Definition at line 161 of file spim\_5411x.c.



## 6.43 CHIP: LPC5411X SPI slave driver

### 6.43.1 Detailed Description

#### Data Structures

- struct [SPIS\\_XFER\\_T](#)

#### Enumerations

- enum [SPIS\\_EVENT\\_T](#) {  
[SPIS\\_EVENT\\_SASSERT](#), [SPIS\\_EVENT\\_SDEASSERT](#), [SPIS\\_EVENT\\_DONE](#), [SPIS\\_EVENT\\_ERRORTX](#),  
[SPIS\\_EVENT\\_ERRORRX](#), [SPIS\\_EVENT\\_THRESHOLD](#) }  
*Slave callback events.*

#### Functions

- void [Chip\\_SPIS\\_Init](#) ([LPC\\_SPI\\_T](#) \*pSPI)  
*SPI slave initialization.*
- void [Chip\\_SPIS\\_EnableInts](#) ([LPC\\_SPI\\_T](#) \*pSPI)  
*SPI slave interrupt enable.*
- void [Chip\\_SPIS\\_DisableInts](#) ([LPC\\_SPI\\_T](#) \*pSPI)  
*SPI slave interrupt disable.*
- void [Chip\\_SPIS\\_LoadFIFO](#) ([LPC\\_SPI\\_T](#) \*pSPI, [SPIS\\_XFER\\_T](#) \*xfer)  
*Load slave transmit FIFO.*
- void [Chip\\_SPIS\\_ReadFIFO](#) ([LPC\\_SPI\\_T](#) \*pSPI, [SPIS\\_XFER\\_T](#) \*xfer)  
*SPI slave FIFO read.*
- void [Chip\\_SPIS\\_XferHandler](#) ([LPC\\_SPI\\_T](#) \*pSPI, [SPIS\\_XFER\\_T](#) \*xfer)  
*SPI slave transfer state change handler.*

### 6.43.2 Enumeration Type Documentation

#### 6.43.2.1 enum [SPIS\\_EVENT\\_T](#)

Slave callback events.

##### Enumerator

- [SPIS\\_EVENT\\_SASSERT](#)** Slave select line asserted  
**[SPIS\\_EVENT\\_SDEASSERT](#)** Slave select line de-asserted  
**[SPIS\\_EVENT\\_DONE](#)** Slave done with this transfer [More data can be provided]  
**[SPIS\\_EVENT\\_ERRORTX](#)** TX Underflow error  
**[SPIS\\_EVENT\\_ERRORRX](#)** RX Overflow error  
**[SPIS\\_EVENT\\_THRESHOLD](#)** Slave transfer has reached its threshold

Definition at line 47 of file spis\_5411x.h.

### 6.43.3 Function Documentation

#### 6.43.3.1 void [Chip\\_SPIS\\_DisableInts](#) ( [LPC\\_SPI\\_T](#) \* pSPI )

SPI slave interrupt disable.

**Parameters**

<i>pSPI</i>	: The base of SPI peripheral on the chip
-------------	--

**Returns**

Nothing

**Note**

Clears / disables the following SPI interrupts: SPI\_INT\_SSAEN, SPI\_INT\_SSDEN. Clears / disables the following FIFO interrupts: SPI\_FIFOINT\_RXERR, SPI\_FIFOINT\_TXERR, SPI\_FIFOINT\_RXLVL, and SPI\_FIFOINT\_TXLVL.

Definition at line 132 of file spis\_5411x.c.

**6.43.3.2 void Chip\_SPIS\_EnableInts ( LPC\_SPI\_T \* *pSPI* )**

SPI slave interrupt enable.

**Parameters**

<i>pSPI</i>	: The base of SPI peripheral on the chip
-------------	--

**Returns**

Nothing

**Note**

Clears / enables the following SPI interrupts: SPI\_INT\_SSAEN, SPI\_INT\_SSDEN. Clears / enables the following FIFO interrupts: SPI\_FIFOINT\_RXERR, SPI\_FIFOINT\_TXERR, SPI\_FIFOINT\_RXLVL, and SPI\_FIFOINT\_TXLVL.

Definition at line 122 of file spis\_5411x.c.

**6.43.3.3 void Chip\_SPIS\_Init ( LPC\_SPI\_T \* *pSPI* )**

SPI slave initialization.

**Parameters**

<i>pSPI</i>	: The base of SPI peripheral on the chip
-------------	--

**Returns**

Nothing

**Note**

Initializes a slave SPI port.

Definition at line 110 of file spis\_5411x.c.

**6.43.3.4 void Chip\_SPIS\_LoadFIFO ( LPC\_SPI\_T \* *pSPI*, SPIS\_XFER\_T \* *xfer* )**

Load slave transmit FIFO.

## Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
<i>xfer</i>	: Pointer to a <a href="#">SPIS_XFER_T</a> structure see notes below

## Returns

Nothing

## Note

Prime the transmit FIFO with data prior to the master xfer start. If data is not pre-buffered, the initial slave transmit data will always be 0x0 with a slave transmit underflow status.

Definition at line 142 of file spis\_5411x.c.

6.43.3.5 void Chip\_SPIS\_ReadFIFO ( LPC\_SPI\_T \* *pSPI*, SPIS\_XFER\_T \* *xfer* )

SPI slave FIFO read.

## Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
<i>xfer</i>	: Pointer to a <a href="#">SPIS_XFER_T</a> structure

## Returns

Nothing

## Note

This function reads all RX FIFO data into the xfer RX buffer. Interrupts are not required and should be disabled prior to calling this function.

Definition at line 163 of file spis\_5411x.c.

6.43.3.6 void Chip\_SPIS\_XferHandler ( LPC\_SPI\_T \* *pSPI*, SPIS\_XFER\_T \* *xfer* )

SPI slave transfer state change handler.

## Parameters

<i>pSPI</i>	: The base of SPI peripheral on the chip
<i>xfer</i>	: Pointer to a <a href="#">SPIS_XFER_T</a> structure see notes below

## Returns

Nothing

## Note

See [SPIS\\_XFER\\_T](#) for more information on this function. When using this function, the SPI slave interrupts should be enabled and setup in the SPI interrupt handler to call this function when they fire. This function is meant to be called from the interrupt handler. The [SPIS\\_XFER\\_T](#) data does not need to be setup prior to the call and should be setup by the callbacks instead.

The callbacks are handled in the interrupt handler. If you are getting overflow or underflow errors, you might need to lower the speed of the master clock or extend the master's select assertion time.

Definition at line 177 of file spis\_5411x.c.

## 6.44 CHIP: LPC5411X State Configurable Timer PWM driver

### 6.44.1 Detailed Description

For more information on how to use the driver please visit the FAQ page at [www.lpcware.com](http://www.lpcware.com)

#### Functions

- `__STATIC_INLINE uint32_t Chip_SCTPWM_GetTicksPerCycle (LPC_SCT_T *pSCT)`  
*Get number of ticks per PWM cycle.*
- `__STATIC_INLINE uint32_t Chip_SCTPWM_PercentageToTicks (LPC_SCT_T *pSCT, uint8_t percent)`  
*Converts a percentage to ticks.*
- `__STATIC_INLINE uint32_t Chip_SCTPWM_GetDutyCycle (LPC_SCT_T *pSCT, uint8_t index)`  
*Get number of ticks on per PWM cycle.*
- `__STATIC_INLINE void Chip_SCTPWM_SetDutyCycle (LPC_SCT_T *pSCT, uint8_t index, uint32_t ticks)`  
*Get number of ticks on per PWM cycle.*
- `__STATIC_INLINE void Chip_SCTPWM_Init (LPC_SCT_T *pSCT)`  
*Initialize the SCT/PWM clock and reset.*
- `__STATIC_INLINE void Chip_SCTPWM_Start (LPC_SCT_T *pSCT)`  
*Start the SCT PWM.*
- `__STATIC_INLINE void Chip_SCTPWM_Stop (LPC_SCT_T *pSCT)`  
*Stop the SCT PWM.*
- `void Chip_SCTPWM_SetRate (LPC_SCT_T *pSCT, uint32_t freq)`  
*Sets the frequency of the generated PWM wave.*
- `void Chip_SCTPWM_SetOutPin (LPC_SCT_T *pSCT, uint8_t index, uint8_t pin)`  
*Setup the OUTPUT pin and associate it with an index.*

### 6.44.2 Function Documentation

#### 6.44.2.1 `__STATIC_INLINE uint32_t Chip_SCTPWM_GetDutyCycle ( LPC_SCT_T * pSCT, uint8_t index )`

Get number of ticks on per PWM cycle.

##### Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>index</i>	: Index of the PWM 1 to N (see notes)

##### Returns

Number of ticks for which the output will be ON per cycle

##### Note

*index* will be 1 to N where N is the "Number of match registers available in the SCT - 1" or "Number of OUTPUT pins available in the SCT" whichever is minimum.

Definition at line 86 of file `sct_pwm_5411x.h`.

#### 6.44.2.2 `__STATIC_INLINE uint32_t Chip_SCTPWM_GetTicksPerCycle ( LPC_SCT_T * pSCT )`

Get number of ticks per PWM cycle.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
-------------	--

## Returns

Number of ticks that will be counted per cycle

## Note

Return value of this function will be valid only after calling [Chip\\_SCTPWM\\_SetRate\(\)](#)

Definition at line 56 of file sct\_pwm\_5411x.h.

6.44.2.3 `__STATIC_INLINE void Chip_SCTPWM_Init ( LPC_SCT_T * pSCT )`

Initialize the SCT/PWM clock and reset.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
-------------	--

## Returns

None

Definition at line 113 of file sct\_pwm\_5411x.h.

6.44.2.4 `__STATIC_INLINE uint32_t Chip_SCTPWM_PercentageToTicks ( LPC_SCT_T * pSCT, uint8_t percent )`

Converts a percentage to ticks.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>percent</i>	: Percentage to convert (0 - 100)

## Returns

Number of ticks corresponding to given percentage

## Note

Do not use this function when using very low pwm rate (like 100Hz or less), on a chip that has very high frequency as the calculation might cause integer overflow

Definition at line 71 of file sct\_pwm\_5411x.h.

6.44.2.5 `__STATIC_INLINE void Chip_SCTPWM_SetDutyCycle ( LPC_SCT_T * pSCT, uint8_t index, uint32_t ticks )`

Get number of ticks on per PWM cycle.

## Parameters

---

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>index</i>	: Index of the PWM 1 to N (see notes)
<i>ticks</i>	: Number of ticks the output should stay ON

**Returns**

None

**Note**

*index* will be 1 to N where N is the "Number of match registers available in the SCT - 1" or "Number of OUTPUT pins available in the SCT" whichever is minimum. The new duty cycle will be effective only after completion of current PWM cycle.

Definition at line 103 of file sct\_pwm\_5411x.h.

6.44.2.6 void Chip\_SCTPWM\_SetOutPin ( LPC\_SCT\_T \* *pSCT*, uint8\_t *index*, uint8\_t *pin* )

Setup the OUTPUT pin and associate it with an index.

**Parameters**

<i>pSCT</i>	: The base of the SCT peripheral on the chip
<i>index</i>	: Index of PWM 1 to N (see notes)
<i>pin</i>	: COUT pin to be associated with the index

**Returns**

None

**Note**

*index* will be 1 to N where N is the "Number of match registers available in the SCT - 1" or "Number of OUTPUT pins available in the SCT" whichever is minimum.

Definition at line 51 of file sct\_pwm\_5411x.c.

6.44.2.7 void Chip\_SCTPWM\_SetRate ( LPC\_SCT\_T \* *pSCT*, uint32\_t *freq* )

Sets the frequency of the generated PWM wave.

**Parameters**

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>freq</i>	: Frequency in Hz

**Returns**

None

Definition at line 67 of file sct\_pwm\_5411x.c.

6.44.2.8 \_\_STATIC\_INLINE void Chip\_SCTPWM\_Start ( LPC\_SCT\_T \* *pSCT* )

Start the SCT PWM.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
-------------	--

## Returns

None

## Note

This function must be called after all the configuration is completed. Do not call [Chip\\_SCTPWM\\_SetRate\(\)](#) or [Chip\\_SCTPWM\\_SetOutPin\(\)](#) after the SCT/PWM is started. Use [Chip\\_SCTPWM\\_Stop\(\)](#) to stop the SCT/PWM before reconfiguring, [Chip\\_SCTPWM\\_SetDutyCycle\(\)](#) can be called when the SCT/PWM is running to change the DutyCycle.

Definition at line 129 of file sct\_pwm\_5411x.h.

#### 6.44.2.9 `__STATIC_INLINE void Chip_SCTPWM_Stop ( LPC_SCT_T * pSCT )`

Stop the SCT PWM.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
-------------	--

## Returns

None

Definition at line 139 of file sct\_pwm\_5411x.h.

## 6.45 CHIP: LPC5411X State Configurable Timer driver

### 6.45.1 Detailed Description

#### Data Structures

- struct [LPC\\_SCT\\_T](#)  
*State Configurable Timer register block structure.*

#### Macros

- #define [CONFIG\\_SCT\\_nEV](#) (13)
- #define [CONFIG\\_SCT\\_nRG](#) (13)
- #define [CONFIG\\_SCT\\_nOU](#) (8)
- #define [CONFIG\\_SCT\\_nIN](#) (8)
- #define [SCT\\_CONFIG\\_16BIT\\_COUNTER](#) 0x00000000  
*Macro defines for SCT configuration register.*
- #define [SCT\\_CONFIG\\_32BIT\\_COUNTER](#) 0x00000001
- #define [SCT\\_CONFIG\\_CLKMODE\\_BUSCLK](#) (0x0 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_SCTCLK](#) (0x1 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_INCLK](#) (0x2 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_INEDGECLK](#) (0x3 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_SYSCCLK](#) (0x0 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_PRESCALED\\_SYSCCLK](#) (0x1 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_SCT\\_INPUT](#) (0x2 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_PRESCALED\\_SCT\\_INPUT](#) (0x3 << 1)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_0](#) (0x0UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_0](#) (0x1UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_1](#) (0x2UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_1](#) (0x3UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_2](#) (0x4UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_2](#) (0x5UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_3](#) (0x6UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_3](#) (0x7UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_4](#) (0x8UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_4](#) (0x9UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_5](#) (0xAUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_5](#) (0xBUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_6](#) (0xCUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_6](#) (0xDUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_7](#) (0xEUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_7](#) (0xFUL << 3)
- #define [SCT\\_CONFIG\\_NORELOAD\\_U](#) (0x1 << 7)
- #define [SCT\\_CONFIG\\_NORELOADH](#) (0x1 << 8)
- #define [SCT\\_CONFIG\\_AUTOLIMIT\\_U](#) (0x1UL << 17)
- #define [SCT\\_CONFIG\\_AUTOLIMIT\\_L](#) (0x1UL << 17)
- #define [SCT\\_CONFIG\\_AUTOLIMIT\\_H](#) (0x1UL << 18)
- #define [COUNTUP\\_TO\\_LIMIT\\_THEN\\_CLEAR\\_TO\\_ZERO](#) 0  
*Macro defines for SCT control register.*
- #define [COUNTUP\\_TO\\_LIMIT\\_THEN\\_CLEAR\\_TO\\_ZERO](#) 0  
*Macro defines for SCT control register.*
- #define [COUNTUP\\_TO\\_LIMIT\\_THEN\\_COUNTDOWN\\_TO\\_ZERO](#) 1
- #define [COUNTUP\\_TO\\_LIMIT\\_THEN\\_COUNTDOWN\\_TO\\_ZERO](#) 1



- #define `SCT_CTRL_STOP_L` (1 << 1)
- #define `SCT_CTRL_HALT_L` (1 << 2)
- #define `SCT_CTRL_CLRCTR_L` (1 << 3)
- #define `SCT_CTRL_BIDIR_L(x)` (((x) & 0x01) << 4)
- #define `SCT_CTRL_PRE_L(x)` (((x) & 0xFF) << 5)
- #define `SCT_CTRL_STOP_H` (1 << 17)
- #define `SCT_CTRL_HALT_H` (1 << 18)
- #define `SCT_CTRL_CLRCTR_H` (1 << 19)
- #define `SCT_CTRL_BIDIR_H(x)` (((x) & 0x01) << 20)
- #define `SCT_CTRL_PRE_H(x)` (((x) & 0xFF) << 21)
- #define `SCT_EV_CTRL_MATCHSEL(reg)` (reg << 0)
- #define `SCT_EV_CTRL_HEVENT_L` (0UL << 4)
- #define `SCT_EV_CTRL_HEVENT_H` (1UL << 4)
- #define `SCT_EV_CTRL_OUTSEL_INPUT` (0UL << 5)
- #define `SCT_EV_CTRL_OUTSEL_OUTPUT` (0UL << 5)
- #define `SCT_EV_CTRL_IOSEL(signal)` (signal << 6)
- #define `SCT_EV_CTRL_IOCOND_LOW` (0UL << 10)
- #define `SCT_EV_CTRL_IOCOND_RISE` (0x1UL << 10)
- #define `SCT_EV_CTRL_IOCOND_FALL` (0x2UL << 10)
- #define `SCT_EV_CTRL_IOCOND_HIGH` (0x3UL << 10)
- #define `SCT_EV_CTRL_COMBMODE_OR` (0x0UL << 12)
- #define `SCT_EV_CTRL_COMBMODE_MATCH` (0x1UL << 12)
- #define `SCT_EV_CTRL_COMBMODE_IO` (0x2UL << 12)
- #define `SCT_EV_CTRL_COMBMODE_AND` (0x3UL << 12)
- #define `SCT_EV_CTRL_STATELD` (0x1UL << 14)
- #define `SCT_EV_CTRL_STATEV(x)` (x << 15)
- #define `SCT_EV_CTRL_MATCHMEM` (0x1UL << 20)
- #define `SCT_EV_CTRL_DIRECTION_INDEPENDENT` (0x0UL << 21)
- #define `SCT_EV_CTRL_DIRECTION_UP` (0x1UL << 21)
- #define `SCT_EV_CTRL_DIRECTION_DOWN` (0x2UL << 21)
- #define `SCT_RES_NOCHANGE` (0)

*Macro defines for SCT Conflict resolution register.*

- #define `SCT_RES_SET_OUTPUT` (1)
- #define `SCT_RES_CLEAR_OUTPUT` (2)
- #define `SCT_RES_TOGGLE_OUTPUT` (3)

## Enumerations

- enum `CHIP_SCT_MATCH_REG_T` {  
`SCT_MATCH_0` = 0, `SCT_MATCH_1`, `SCT_MATCH_2`, `SCT_MATCH_3`,  
`SCT_MATCH_4`, `SCT_MATCH_5`, `SCT_MATCH_6`, `SCT_MATCH_7`,  
`SCT_MATCH_8`, `SCT_MATCH_9`, `SCT_MATCH_10`, `SCT_MATCH_11`,  
`SCT_MATCH_12`, `SCT_MATCH_13`, `SCT_MATCH_14`, `SCT_MATCH_15` }
- enum `CHIP_SCT_EVENT_T` {  
`SCT_EVT_0` = (1 << 0), `SCT_EVT_1` = (1 << 1), `SCT_EVT_2` = (1 << 2), `SCT_EVT_3` = (1 << 3),  
`SCT_EVT_4` = (1 << 4), `SCT_EVT_5` = (1 << 5), `SCT_EVT_6` = (1 << 6), `SCT_EVT_7` = (1 << 7),  
`SCT_EVT_8` = (1 << 8), `SCT_EVT_9` = (1 << 9), `SCT_EVT_10` = (1 << 10), `SCT_EVT_11` = (1 << 11),  
`SCT_EVT_12` = (1 << 12), `SCT_EVT_13` = (1 << 13), `SCT_EVT_14` = (1 << 14), `SCT_EVT_15` = (1 << 15) }

## Functions

- `__STATIC_INLINE void Chip_SCT_EventControl (LPC_SCT_T *pSCT, uint32_t event_number, uint32_t value)`  
*Set event control register.*
- `__STATIC_INLINE void Chip_SCT_EventStateMask (LPC_SCT_T *pSCT, uint32_t event_number, uint32_t event_state_mask)`  
*Set event state mask register.*
- `__STATIC_INLINE void Chip_SCT_Config (LPC_SCT_T *pSCT, uint32_t cfg)`  
*Set configuration register.*
- `__STATIC_INLINE void Chip_SCT_Limit (LPC_SCT_T *pSCT, uint32_t value)`  
*Configures the Limit register.*
- `void Chip_SCT_SetClrControl (LPC_SCT_T *pSCT, uint32_t value, FunctionalState ena)`  
*Set or Clear the Control register.*
- `void Chip_SCT_SetConflictResolution (LPC_SCT_T *pSCT, uint8_t outnum, uint8_t value)`  
*Set the conflict resolution.*
- `__STATIC_INLINE void Chip_SCT_SetCount (LPC_SCT_T *pSCT, uint32_t count)`  
*Set unified count value in State Configurable Timer.*
- `__STATIC_INLINE void Chip_SCT_SetCountL (LPC_SCT_T *pSCT, uint16_t count)`  
*Set lower count value in State Configurable Timer.*
- `__STATIC_INLINE void Chip_SCT_SetCountH (LPC_SCT_T *pSCT, uint16_t count)`  
*Set higher count value in State Configurable Timer.*
- `__STATIC_INLINE void Chip_SCT_SetMatchCount (LPC_SCT_T *pSCT, CHIP_SCT_MATCH_REG_T n, uint32_t value)`  
*Set unified match count value in State Configurable Timer.*
- `__STATIC_INLINE void Chip_SCT_SetMatchReload (LPC_SCT_T *pSCT, CHIP_SCT_MATCH_REG_T n, uint32_t value)`  
*Set unified match reload count value in State Configurable Timer.*
- `__STATIC_INLINE void Chip_SCT_EnableEventInt (LPC_SCT_T *pSCT, CHIP_SCT_EVENT_T evt)`  
*Enable the interrupt for the specified event in State Configurable Timer.*
- `__STATIC_INLINE void Chip_SCT_DisableEventInt (LPC_SCT_T *pSCT, CHIP_SCT_EVENT_T evt)`  
*Disable the interrupt for the specified event in State Configurable Timer.*
- `__STATIC_INLINE void Chip_SCT_ClearEventFlag (LPC_SCT_T *pSCT, CHIP_SCT_EVENT_T evt)`  
*Clear the specified event flag in State Configurable Timer.*
- `__STATIC_INLINE void Chip_SCT_SetControl (LPC_SCT_T *pSCT, uint32_t value)`  
*Set control register in State Configurable Timer.*
- `__STATIC_INLINE void Chip_SCT_ClearControl (LPC_SCT_T *pSCT, uint32_t value)`  
*Clear control register in State Configurable Timer.*
- `void Chip_SCT_Init (LPC_SCT_T *pSCT)`  
*Initializes the State Configurable Timer.*
- `void Chip_SCT_DeInit (LPC_SCT_T *pSCT)`  
*Deinitializes the State Configurable Timer.*

## 6.45.2 Macro Definition Documentation

### 6.45.2.1 #define CONFIG\_SCT\_nEV (13)

Number of events

Definition at line 48 of file sct\_5411x.h.

**6.45.2.2 #define CONFIG\_SCT\_nIN (8)**

Number of outputs

Definition at line 51 of file sct\_5411x.h.

**6.45.2.3 #define CONFIG\_SCT\_nOU (8)**

Number of outputs

Definition at line 50 of file sct\_5411x.h.

**6.45.2.4 #define CONFIG\_SCT\_nRG (13)**

Number of match/compare registers

Definition at line 49 of file sct\_5411x.h.

**6.45.2.5 #define COUNTUP\_TO\_LIMIT\_THEN\_COUNTDOWN\_TO\_ZERO 1**

Definition at line 224 of file sct\_5411x.h.

**6.45.2.6 #define COUNTUP\_TO\_LIMIT\_THEN\_COUNTDOWN\_TO\_ZERO 1**

Definition at line 224 of file sct\_5411x.h.

**6.45.2.7 #define COUNTUP\_TO\_LIMIT\_THEN\_CLEAR\_TO\_ZERO 0**

Macro defines for SCT control register.

Direction for low or unified counter

Direction for high counter

Definition at line 223 of file sct\_5411x.h.

**6.45.2.8 #define COUNTUP\_TO\_LIMIT\_THEN\_CLEAR\_TO\_ZERO 0**

Macro defines for SCT control register.

Direction for low or unified counter

Direction for high counter

Definition at line 223 of file sct\_5411x.h.

**6.45.2.9 #define SCT\_CONFIG\_16BIT\_COUNTER 0x00000000**

Macro defines for SCT configuration register.

Operate as 2 16-bit counters

Definition at line 176 of file sct\_5411x.h.

**6.45.2.10 #define SCT\_CONFIG\_32BIT\_COUNTER 0x00000001**

Operate as 1 32-bit counter

Definition at line 177 of file sct\_5411x.h.

6.45.2.11 `#define SCT_CONFIG_AUTOLIMIT_H (0x1UL << 18)`

Definition at line 209 of file sct\_5411x.h.

6.45.2.12 `#define SCT_CONFIG_AUTOLIMIT_L (0x1UL << 17)`

Definition at line 208 of file sct\_5411x.h.

6.45.2.13 `#define SCT_CONFIG_AUTOLIMIT_U (0x1UL << 17)`

Definition at line 207 of file sct\_5411x.h.

6.45.2.14 `#define SCT_CONFIG_CKSEL_FALLING_IN_0 (0x1UL << 3)`

Definition at line 190 of file sct\_5411x.h.

6.45.2.15 `#define SCT_CONFIG_CKSEL_FALLING_IN_1 (0x3UL << 3)`

Definition at line 192 of file sct\_5411x.h.

6.45.2.16 `#define SCT_CONFIG_CKSEL_FALLING_IN_2 (0x5UL << 3)`

Definition at line 194 of file sct\_5411x.h.

6.45.2.17 `#define SCT_CONFIG_CKSEL_FALLING_IN_3 (0x7UL << 3)`

Definition at line 196 of file sct\_5411x.h.

6.45.2.18 `#define SCT_CONFIG_CKSEL_FALLING_IN_4 (0x9UL << 3)`

Definition at line 198 of file sct\_5411x.h.

6.45.2.19 `#define SCT_CONFIG_CKSEL_FALLING_IN_5 (0xBUL << 3)`

Definition at line 200 of file sct\_5411x.h.

6.45.2.20 `#define SCT_CONFIG_CKSEL_FALLING_IN_6 (0xDUL << 3)`

Definition at line 202 of file sct\_5411x.h.

6.45.2.21 `#define SCT_CONFIG_CKSEL_FALLING_IN_7 (0xFUL << 3)`

Definition at line 204 of file sct\_5411x.h.

6.45.2.22 `#define SCT_CONFIG_CKSEL_RISING_IN_0 (0x0UL << 3)`

Definition at line 189 of file sct\_5411x.h.

6.45.2.23 **#define SCT\_CONFIG\_CKSEL\_RISING\_IN\_1 (0x2UL << 3)**

Definition at line 191 of file sct\_5411x.h.

6.45.2.24 **#define SCT\_CONFIG\_CKSEL\_RISING\_IN\_2 (0x4UL << 3)**

Definition at line 193 of file sct\_5411x.h.

6.45.2.25 **#define SCT\_CONFIG\_CKSEL\_RISING\_IN\_3 (0x6UL << 3)**

Definition at line 195 of file sct\_5411x.h.

6.45.2.26 **#define SCT\_CONFIG\_CKSEL\_RISING\_IN\_4 (0x8UL << 3)**

Definition at line 197 of file sct\_5411x.h.

6.45.2.27 **#define SCT\_CONFIG\_CKSEL\_RISING\_IN\_5 (0xAUL << 3)**

Definition at line 199 of file sct\_5411x.h.

6.45.2.28 **#define SCT\_CONFIG\_CKSEL\_RISING\_IN\_6 (0xCUL << 3)**

Definition at line 201 of file sct\_5411x.h.

6.45.2.29 **#define SCT\_CONFIG\_CKSEL\_RISING\_IN\_7 (0xEUL << 3)**

Definition at line 203 of file sct\_5411x.h.

6.45.2.30 **#define SCT\_CONFIG\_CLKMODE\_BUSCLK (0x0 << 1)**

Bus clock

Definition at line 179 of file sct\_5411x.h.

6.45.2.31 **#define SCT\_CONFIG\_CLKMODE\_INCLK (0x2 << 1)**

Input clock selected in CLKSEL field

Definition at line 181 of file sct\_5411x.h.

6.45.2.32 **#define SCT\_CONFIG\_CLKMODE\_INEDGECLK (0x3 << 1)**

Input clock edge selected in CLKSEL field

Definition at line 182 of file sct\_5411x.h.

6.45.2.33 **#define SCT\_CONFIG\_CLKMODE\_PRESCALED\_SCT\_INPUT (0x3 << 1)**

Prescaled input clock/edge selected in CKSEL field

Definition at line 187 of file sct\_5411x.h.

**6.45.2.34 #define SCT\_CONFIG\_CLKMODE\_PRESCALED\_SYSCLK (0x1 << 1)**

Prescaled system clock

Definition at line 185 of file sct\_5411x.h.

**6.45.2.35 #define SCT\_CONFIG\_CLKMODE\_SCT\_INPUT (0x2 << 1)**

Input clock/edge selected in CKSEL field

Definition at line 186 of file sct\_5411x.h.

**6.45.2.36 #define SCT\_CONFIG\_CLKMODE\_SCTCLK (0x1 << 1)**

SCT clock

Definition at line 180 of file sct\_5411x.h.

**6.45.2.37 #define SCT\_CONFIG\_CLKMODE\_SYSCLK (0x0 << 1)**

System clock

Definition at line 184 of file sct\_5411x.h.

**6.45.2.38 #define SCT\_CONFIG\_NORELOADH (0x1 << 8)**

Operate as 1 32-bit counter

Definition at line 206 of file sct\_5411x.h.

**6.45.2.39 #define SCT\_CONFIG\_NORELOADL\_U (0x1 << 7)**

Operate as 1 32-bit counter

Definition at line 205 of file sct\_5411x.h.

**6.45.2.40 #define SCT\_CTRL\_BIDIR\_H( x ) (((x) & 0x01) << 20)**

Definition at line 228 of file sct\_5411x.h.

**6.45.2.41 #define SCT\_CTRL\_BIDIR\_L( x ) (((x) & 0x01) << 4)**

Bidirectional bit

Definition at line 220 of file sct\_5411x.h.

**6.45.2.42 #define SCT\_CTRL\_CLRCTR\_H (1 << 19)**

Clear high counter

Definition at line 227 of file sct\_5411x.h.

**6.45.2.43 #define SCT\_CTRL\_CLRCTR\_L (1 << 3)**

Clear low or unified counter

Definition at line 219 of file sct\_5411x.h.

**6.45.2.44 #define SCT\_CTRL\_HALT\_H (1 << 18)**

Halt high counter

Definition at line 226 of file sct\_5411x.h.

**6.45.2.45 #define SCT\_CTRL\_HALT\_L (1 << 2)**

Halt low counter

Definition at line 218 of file sct\_5411x.h.

**6.45.2.46 #define SCT\_CTRL\_PRE\_H( x ) (((x) & 0xFF) << 21)**

Prescale clock for high counter

Definition at line 229 of file sct\_5411x.h.

**6.45.2.47 #define SCT\_CTRL\_PRE\_L( x ) (((x) & 0xFF) << 5)**

Prescale clock for low or unified counter

Definition at line 221 of file sct\_5411x.h.

**6.45.2.48 #define SCT\_CTRL\_STOP\_H (1 << 17)**

Stop high counter

Definition at line 225 of file sct\_5411x.h.

**6.45.2.49 #define SCT\_CTRL\_STOP\_L (1 << 1)**

Stop low counter

Definition at line 217 of file sct\_5411x.h.

**6.45.2.50 #define SCT\_EV\_CTRL\_COMBMODE\_AND (0x3UL << 12)**

Definition at line 245 of file sct\_5411x.h.

**6.45.2.51 #define SCT\_EV\_CTRL\_COMBMODE\_IO (0x2UL << 12)**

Definition at line 244 of file sct\_5411x.h.

**6.45.2.52 #define SCT\_EV\_CTRL\_COMBMODE\_MATCH (0x1UL << 12)**

Definition at line 243 of file sct\_5411x.h.

6.45.2.53 **#define SCT\_EV\_CTRL\_COMBMODE\_OR** (0x0UL << 12)

Definition at line 242 of file sct\_5411x.h.

6.45.2.54 **#define SCT\_EV\_CTRL\_DIRECTION\_DOWN** (0x2UL << 21)

Definition at line 251 of file sct\_5411x.h.

6.45.2.55 **#define SCT\_EV\_CTRL\_DIRECTION\_INDEPENDENT** (0x0UL << 21)

Definition at line 249 of file sct\_5411x.h.

6.45.2.56 **#define SCT\_EV\_CTRL\_DIRECTION\_UP** (0x1UL << 21)

Definition at line 250 of file sct\_5411x.h.

6.45.2.57 **#define SCT\_EV\_CTRL\_HEVENT\_H** (1UL << 4)

Definition at line 233 of file sct\_5411x.h.

6.45.2.58 **#define SCT\_EV\_CTRL\_HEVENT\_L** (0UL << 4)

Definition at line 232 of file sct\_5411x.h.

6.45.2.59 **#define SCT\_EV\_CTRL\_IOCOND\_FALL** (0x2UL << 10)

Definition at line 240 of file sct\_5411x.h.

6.45.2.60 **#define SCT\_EV\_CTRL\_IOCOND\_HIGH** (0x3UL << 10)

Definition at line 241 of file sct\_5411x.h.

6.45.2.61 **#define SCT\_EV\_CTRL\_IOCOND\_LOW** (0UL << 10)

Definition at line 238 of file sct\_5411x.h.

6.45.2.62 **#define SCT\_EV\_CTRL\_IOCOND\_RISE** (0x1UL << 10)

Definition at line 239 of file sct\_5411x.h.

6.45.2.63 **#define SCT\_EV\_CTRL\_IOSEL( *signal* )** (*signal* << 6)

Definition at line 236 of file sct\_5411x.h.

6.45.2.64 **#define SCT\_EV\_CTRL\_MATCHMEM** (0x1UL << 20)

Definition at line 248 of file sct\_5411x.h.



6.45.2.65 `#define SCT_EV_CTRL_MATCHSEL( reg ) (reg << 0)`

Definition at line 231 of file sct\_5411x.h.

6.45.2.66 `#define SCT_EV_CTRL_OUTSEL_INPUT (0UL << 5)`

Definition at line 234 of file sct\_5411x.h.

6.45.2.67 `#define SCT_EV_CTRL_OUTSEL_OUTPUT (0UL << 5)`

Definition at line 235 of file sct\_5411x.h.

6.45.2.68 `#define SCT_EV_CTRL_STATELD (0x1UL << 14)`

Definition at line 246 of file sct\_5411x.h.

6.45.2.69 `#define SCT_EV_CTRL_STATEV( x ) (x << 15)`

Definition at line 247 of file sct\_5411x.h.

6.45.2.70 `#define SCT_RES_CLEAR_OUTPUT (2)`

Definition at line 258 of file sct\_5411x.h.

6.45.2.71 `#define SCT_RES_NOCHANGE (0)`

Macro defines for SCT Conflict resolution register.

Definition at line 256 of file sct\_5411x.h.

6.45.2.72 `#define SCT_RES_SET_OUTPUT (1)`

Definition at line 257 of file sct\_5411x.h.

6.45.2.73 `#define SCT_RES_TOGGLE_OUTPUT (3)`

Definition at line 259 of file sct\_5411x.h.

### 6.45.3 Enumeration Type Documentation

6.45.3.1 `enum CHIP_SCT_EVENT_T`

SCT Event values enum

Enumerator

**`SCT_EVT_0`** Event 0

**`SCT_EVT_1`** Event 1

**`SCT_EVT_2`** Event 2

**`SCT_EVT_3`** Event 3

**`SCT_EVT_4`** Event 4

***SCT\_EVT\_5*** Event 5  
***SCT\_EVT\_6*** Event 6  
***SCT\_EVT\_7*** Event 7  
***SCT\_EVT\_8*** Event 8  
***SCT\_EVT\_9*** Event 9  
***SCT\_EVT\_10*** Event 10  
***SCT\_EVT\_11*** Event 11  
***SCT\_EVT\_12*** Event 12  
***SCT\_EVT\_13*** Event 13  
***SCT\_EVT\_14*** Event 14  
***SCT\_EVT\_15*** Event 15

Definition at line 286 of file sct\_5411x.h.

#### 6.45.3.2 enum CHIP\_SCT\_MATCH\_REG\_T

SCT Match register values enum

Enumerator

***SCT\_MATCH\_0*** SCT Match register 0  
***SCT\_MATCH\_1***  
***SCT\_MATCH\_2***  
***SCT\_MATCH\_3***  
***SCT\_MATCH\_4***  
***SCT\_MATCH\_5***  
***SCT\_MATCH\_6***  
***SCT\_MATCH\_7***  
***SCT\_MATCH\_8***  
***SCT\_MATCH\_9***  
***SCT\_MATCH\_10***  
***SCT\_MATCH\_11***  
***SCT\_MATCH\_12***  
***SCT\_MATCH\_13***  
***SCT\_MATCH\_14***  
***SCT\_MATCH\_15***

Definition at line 264 of file sct\_5411x.h.

#### 6.45.4 Function Documentation

##### 6.45.4.1 `__STATIC_INLINE void Chip_SCT_ClearControl ( LPC_SCT_T * pSCT, uint32_t value )`

Clear control register in State Configurable Timer.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>value</i>	: Value (ORed value of SCT_CTRL_* bits)

## Returns

Nothing

Definition at line 477 of file sct\_5411x.h.

#### 6.45.4.2 `__STATIC_INLINE void Chip_SCT_ClearEventFlag ( LPC_SCT_T * pSCT, CHIP_SCT_EVENT_T evt )`

Clear the specified event flag in State Configurable Timer.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>evt</i>	: Event value

## Returns

Nothing

Definition at line 457 of file sct\_5411x.h.

#### 6.45.4.3 `__STATIC_INLINE void Chip_SCT_Config ( LPC_SCT_T * pSCT, uint32_t cfg )`

Set configuration register.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>cfg</i>	: The 32-bit configuration setting

## Returns

Nothing

Definition at line 335 of file sct\_5411x.h.

#### 6.45.4.4 `void Chip_SCT_DeInit ( LPC_SCT_T * pSCT )`

Deinitializes the State Configurable Timer.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
-------------	--

## Returns

Nothing

Definition at line 58 of file sct\_5411x.c.

#### 6.45.4.5 `__STATIC_INLINE void Chip_SCT_DisableEventInt ( LPC_SCT_T * pSCT, CHIP_SCT_EVENT_T evt )`

Disable the interrupt for the specified event in State Configurable Timer.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>evt</i>	: Event value

## Returns

Nothing

Definition at line 447 of file sct\_5411x.h.

6.45.4.6 `__STATIC_INLINE void Chip_SCT_EnableEventInt ( LPC_SCT_T * pSCT, CHIP_SCT_EVENT_T evt )`

Enable the interrupt for the specified event in State Configurable Timer.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>evt</i>	: Event value

## Returns

Nothing

Definition at line 437 of file sct\_5411x.h.

6.45.4.7 `__STATIC_INLINE void Chip_SCT_EventControl ( LPC_SCT_T * pSCT, uint32_t event_number, uint32_t value )`

Set event control register.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>event_number</i>	
<i>value</i>	: The 32-bit event control setting

## Returns

Nothing

Definition at line 312 of file sct\_5411x.h.

6.45.4.8 `__STATIC_INLINE void Chip_SCT_EventStateMask ( LPC_SCT_T * pSCT, uint32_t event_number, uint32_t event_state_mask )`

Set event state mask register.

## Parameters

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>event_number</i>	
<i>event_state_mask</i>	: The 32-bit event state mask setting

## Returns

Nothing

Definition at line 324 of file sct\_5411x.h.

6.45.4.9 void Chip\_SCT\_Init ( LPC\_SCT\_T \* *pSCT* )

Initializes the State Configurable Timer.

**Parameters**

<i>pSCT</i>	: The base of SCT peripheral on the chip
-------------	--

**Returns**

Nothing

Definition at line 51 of file sct\_5411x.c.

#### 6.45.4.10 `__STATIC_INLINE void Chip_SCT_Limit ( LPC_SCT_T * pSCT, uint32_t value )`

Configures the Limit register.

**Parameters**

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>value</i>	: The 32-bit Limit register value

**Returns**

Nothing

Definition at line 345 of file sct\_5411x.h.

#### 6.45.4.11 `void Chip_SCT_SetClrControl ( LPC_SCT_T * pSCT, uint32_t value, FunctionalState ena )`

Set or Clear the Control register.

**Parameters**

<i>pSCT</i>	: Pointer to SCT register block
<i>value</i>	: SCT Control register value
<i>ena</i>	: ENABLE - To set the fields specified by value : DISABLE - To clear the field specified by value

**Returns**

Nothing Set or clear the control register bits as specified by the *value* parameter. If *ena* is set to ENABLE, the mentioned register fields will be set. If *ena* is set to DISABLE, the mentioned register fields will be cleared

Definition at line 64 of file sct\_5411x.c.

#### 6.45.4.12 `void Chip_SCT_SetConflictResolution ( LPC_SCT_T * pSCT, uint8_t outnum, uint8_t value )`

Set the conflict resolution.

**Parameters**

<i>pSCT</i>	: Pointer to SCT register block
<i>outnum</i>	: Output number
<i>value</i>	: Output value <ul style="list-style-type: none"> <li>SCT_RES_NOCHANGE :No change</li> </ul>

- SCT\_RES\_SET\_OUTPUT :Set output
- SCT\_RES\_CLEAR\_OUTPUT :Clear output

- SCT\_RES\_TOGGLE\_OUTPUT :Toggle output : SCT\_RES\_NOCHANGE : DISABLE - To clear the field specified by value

**Returns**

Nothing Set conflict resolution for the output *outnum*

Definition at line 75 of file sct\_5411x.c.

#### 6.45.4.13 `__STATIC_INLINE void Chip_SCT_SetControl ( LPC_SCT_T * pSCT, uint32_t value )`

Set control register in State Configurable Timer.

**Parameters**

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>value</i>	: Value (ORed value of SCT_CTRL_* bits)

**Returns**

Nothing

Definition at line 467 of file sct\_5411x.h.

#### 6.45.4.14 `__STATIC_INLINE void Chip_SCT_SetCount ( LPC_SCT_T * pSCT, uint32_t count )`

Set unified count value in State Configurable Timer.

**Parameters**

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>count</i>	: The 32-bit count value

**Returns**

Nothing

Definition at line 385 of file sct\_5411x.h.

#### 6.45.4.15 `__STATIC_INLINE void Chip_SCT_SetCountH ( LPC_SCT_T * pSCT, uint16_t count )`

Set higher count value in State Configurable Timer.

**Parameters**

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>count</i>	: The 16-bit count value

**Returns**

Nothing

Definition at line 405 of file sct\_5411x.h.

#### 6.45.4.16 `__STATIC_INLINE void Chip_SCT_SetCountL ( LPC_SCT_T * pSCT, uint16_t count )`

Set lower count value in State Configurable Timer.

**Parameters**

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>count</i>	: The 16-bit count value

**Returns**

Nothing

Definition at line 395 of file sct\_5411x.h.

6.45.4.17 `__STATIC_INLINE void Chip_SCT_SetMatchCount ( LPC_SCT_T * pSCT, CHIP_SCT_MATCH_REG_T n, uint32_t value )`

Set unified match count value in State Configurable Timer.

**Parameters**

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>n</i>	: Match register value
<i>value</i>	: The 32-bit match count value

**Returns**

Nothing

Definition at line 416 of file sct\_5411x.h.

6.45.4.18 `__STATIC_INLINE void Chip_SCT_SetMatchReload ( LPC_SCT_T * pSCT, CHIP_SCT_MATCH_REG_T n, uint32_t value )`

Set unified match reload count value in State Configurable Timer.

**Parameters**

<i>pSCT</i>	: The base of SCT peripheral on the chip
<i>n</i>	: Match register value
<i>value</i>	: The 32-bit match count reload value

**Returns**

Nothing

Definition at line 427 of file sct\_5411x.h.



## 6.46 CHIP: LPC5411X System and Control Driver

### 6.46.1 Detailed Description

#### Data Structures

- struct [LPC\\_SYSCON\\_T](#)  
*LPC5411X Main system configuration register block structure.*
- struct [LPC\\_ASYNC\\_SYSCON\\_T](#)  
*LPC5411X Asynchronous system configuration register block structure.*

#### Macros

- #define [SYSCON\\_FROCTRL\\_MASK](#) ((1 << 15) | (0xF << 26))  
*FROCTRL register bits.*
- #define [SYSCON\\_FROCTRL\\_WRTRIM](#) (1UL << 31)
- #define [SYSCON\\_FROCTRL\\_HSPDCLK](#) (1UL << 30)
- #define [SYSCON\\_FROCTRL\\_USBMODCHG](#) (1UL << 25)
- #define [SYSCON\\_FROCTRL\\_USBCLKADJ](#) (1UL << 24)
- #define [SYSCON\\_FROCTRL\\_SEL96MHZ](#) (1UL << 14)
- #define [SYSCON\\_NMISRC\\_M0\\_ENABLE](#) ((uint32\_t) 1 << 30)
- #define [SYSCON\\_NMISRC\\_M4\\_ENABLE](#) ((uint32\_t) 1 << 31)
- #define [SYSCON\\_RST\\_POR](#) (1 << 0)
- #define [SYSCON\\_RST\\_EXTRST](#) (1 << 1)
- #define [SYSCON\\_RST\\_WDT](#) (1 << 2)
- #define [SYSCON\\_RST\\_BOD](#) (1 << 3)
- #define [SYSCON\\_RST\\_SYSRST](#) (1 << 4)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_FRO](#) (1 << 4)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_FLASH](#) (1 << 5)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_TS](#) (1 << 6)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_BOD\\_RST](#) (1 << 7)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_BOD\\_INTR](#) (1 << 8)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_ADC0](#) (1 << 10)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_VDDFLASH](#) (1 << 11)
- #define [SYSCON\\_PDRUNCFG\\_LP\\_VDDFLASH](#) (1 << 12)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_SRAM0](#) (1 << 13)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_SRAM1](#) (1 << 14)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_SRAM2](#) (1 << 15)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_SRAMX](#) (1 << 16)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_ROM](#) (1 << 17)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_VDDHV\\_ENA](#) (1 << 18)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_VDDA\\_ENA](#) (1 << 19)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_WDT\\_OSC](#) (1 << 20)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_USB\\_PHY](#) (1 << 21)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_SYS\\_PLL](#) (1 << 22)
- #define [SYSCON\\_PDRUNCFG\\_PD\\_VREFP](#) (1 << 23)
- #define [SYSCON\\_AUTOCGOR\\_RAM0X](#) (1 << 1)
- #define [SYSCON\\_AUTOCGOR\\_RAM1](#) (1 << 2)
- #define [SYSCON\\_AUTOCGOR\\_RAM2](#) (1 << 3)
- #define [SYSCON\\_AUTOCGOR\\_MASK](#) (SYSCON\_AUTOCGOR\_RAM0X | SYSCON\_AUTOCGOR\_RAM1 | SYSCON\_AUTOCGOR\_RAM2)

## Enumerations

- enum `CHIP_SYSCON_BOOT_MODE_REMAP_T` { `REMAP_BOOT_LOADER_MODE`, `REMAP_USER_RAM_MODE`, `REMAP_USER_FLASH_MODE` }
  - enum `CHIP_SYSCON_PERIPH_RESET_T` {  
`RESET_FLASH = 7`, `RESET_FMC`, `RESET_SPIFI = 10`, `RESET_MUX`,  
`RESET_IOCON = 13`, `RESET_GPIO0`, `RESET_GPIO1`, `RESET_PINT = 18`,  
`RESET_GINT`, `RESET_DMA`, `RESET_CRC`, `RESET_WWDT`,  
`RESET_ADC = 27`, `RESET_ADC0 = 27`, `RESET_MRT = 32`, `RESET_SCT0 = 32 + 2`,  
`RESET_SCT = 32 + 2`, `RESET_UTICK = 32 + 10`, `RESET_FLEXCOMM0`, `RESET_FLEXCOMM1`,  
`RESET_FLEXCOMM2`, `RESET_FLEXCOMM3`, `RESET_FLEXCOMM4`, `RESET_FLEXCOMM5`,  
`RESET_FLEXCOMM6`, `RESET_FLEXCOMM7`, `RESET_DMIC`, `RESET_TIMER2 = 32 + 22`,  
`RESET_USB`, `RESET_TIMER0`, `RESET_TIMER1`, `RESET_TIMER3 = 128 + 13`,  
`RESET_TIMER4` }
  - enum `SYSCON_FLASHTIM_T` {  
`SYSCON_FLASH_1CYCLE = 0`, `FLASHTIM_20MHZ_CPU = SYSCON_FLASH_1CYCLE`, `SYSCON_FLASH_2CYCLE`,  
`SYSCON_FLASH_3CYCLE`,  
`SYSCON_FLASH_4CYCLE`, `SYSCON_FLASH_5CYCLE`, `SYSCON_FLASH_6CYCLE`, `SYSCON_FLASH_7CYCLE`,  
`SYSCON_FLASH_8CYCLE` }
- FLASH Access time definitions.*
- enum `CHIP_SYSCON_WAKEUP_T` {  
`SYSCON_STARTER_WWDT_BOD = 0`, `SYSCON_STARTER_DMA`, `SYSCON_STARTER_GINT0`, `SYSCON_STARTER_GINT1`,  
`SYSCON_STARTER_PINT0`, `SYSCON_STARTER_PINT1`, `SYSCON_STARTER_PINT2`, `SYSCON_STARTER_PINT3`,  
`SYSCON_STARTER_UTICK`, `SYSCON_STARTER_MRT`, `SYSCON_STARTER_TIMER0`, `SYSCON_STARTER_TIMER1`,  
`SYSCON_STARTER_SCT0`, `SYSCON_STARTER_TIMER3`, `SYSCON_STARTER_FLEXCOMM0`, `SYSCON_STARTER_FLEXCOMM1`,  
`SYSCON_STARTER_FLEXCOMM2`, `SYSCON_STARTER_FLEXCOMM3`, `SYSCON_STARTER_FLEXCOMM4`, `SYSCON_STARTER_FLEXCOMM5`,  
`SYSCON_STARTER_FLEXCOMM6`, `SYSCON_STARTER_FLEXCOMM7`, `SYSCON_STARTER_ADC0_SEQB`,  
`SYSCON_STARTER_ADC0_THCMP`, `SYSCON_STARTER_DMIC`, `SYSCON_STARTER_HWVAD`, `SYSCON_STARTER_USBNEEDCLK`,  
`SYSCON_STARTER_USB`, `SYSCON_STARTER_RTC`, `SYSCON_STARTER_RESERVED0`, `SYSCON_STARTER_MAILBOX`,  
`SYSCON_STARTER_PINT4`, `SYSCON_STARTER_PINT5`, `SYSCON_STARTER_PINT6`, `SYSCON_STARTER_PINT7`,  
`SYSCON_STARTER_TIMER2`, `SYSCON_STARTER_TIMER4` }

## Functions

- `__STATIC_INLINE void Chip_SYSCON_Map (CHIP_SYSCON_BOOT_MODE_REMAP_T remap)`  
*Re-map interrupt vectors.*
- `__STATIC_INLINE CHIP_SYSCON_BOOT_MODE_REMAP_T Chip_SYSCON_GetMemoryMap (void)`  
*Get system remap setting.*
- `__STATIC_INLINE void Chip_SYSCON_SetSYSTCKCAL (uint32_t sysCalVal)`  
*Set System tick timer calibration value.*
- `void Chip_SYSCON_SetNMISource (uint32_t intsrc)`  
*Set source for non-maskable interrupt (NMI)*
- `void Chip_SYSCON_EnableNMISource (void)`  
*Enable interrupt used for NMI source.*
- `void Chip_SYSCON_DisableNMISource (void)`

- Disable interrupt used for NMI source.*

  - void [Chip\\_SYSCON\\_Enable\\_ASYNC\\_Syscon](#) (bool enable)

*Enable or disable asynchronous APB bridge and subsystem.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_SetUSARTFRGCtrl](#) (uint8\_t fmul, uint8\_t fddiv)

*Set UART Fractional divider value.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SYSCON\\_GetSystemRSTStatus](#) (void)

*Get system reset status.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_ClearSystemRSTStatus](#) (uint32\_t reset)

*Clear system reset status.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_PeriphReset](#) (CHIP\_SYSCON\_PERIPH\_RESET\_T periph)

*Resets a peripheral.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SYSCON\\_GetPORPIOStatus](#) (uint8\_t port)

*Read POR captured PIO status.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SYSCON\\_GetResetPIOStatus](#) (uint8\_t port)

*Read reset captured PIO status.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_StartFreqMeas](#) (void)

*Starts a frequency measurement cycle.*
- `__STATIC_INLINE` bool [Chip\\_SYSCON\\_IsFreqMeasComplete](#) (void)

*Indicates when a frequency measurement cycle is complete.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SYSCON\\_GetRawFreqMeasCapval](#) (void)

*Returns the raw capture value for a frequency measurement cycle.*
- uint32\_t [Chip\\_SYSCON\\_GetCompFreqMeas](#) (uint32\_t refClockRate)

*Returns the computed value for a frequency measurement cycle.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_SetFLASHAccess](#) (SYSCON\_FLASHTIM\_T clks)

*Set FLASH memory access time in clocks.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SYSCON\\_GetPowerStates](#) (void)

*Power up one or more blocks or peripherals.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_PowerDown](#) (uint32\_t powerdownmask)

*Power down one or more blocks or peripherals.*
- void [Chip\\_SYSCON\\_PowerUp](#) (uint32\_t powerupmask)

*Power up one or more blocks or peripherals.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_EnableWakeup](#) (CHIP\_SYSCON\_WAKEUP\_T periphId)

*Enables a pin's (PINT) wakeup logic.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_DisableWakeup](#) (CHIP\_SYSCON\_WAKEUP\_T periphId)

*Disables peripheral's wakeup logic.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SYSCON\\_GetDeviceID](#) (void)

*Return the pointer to device ID registers.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_DisableAutoClocking](#) (uint32\_t mask)

*Disables Auto clock gating for SRAM's.*
- `__STATIC_INLINE` void [Chip\\_SYSCON\\_EnableAutoClocking](#) (uint32\_t mask)

*Re-enables Auto clock gating for SRAM's.*

## 6.46.2 Macro Definition Documentation

6.46.2.1 `#define SYSCON_AUTOCGOR_MASK (SYSCON_AUTOCGOR_RAM0X | SYSCON_AUTOCGOR_RAM1 | SYSCON_AUTOCGOR_RAM2)`

Definition at line 579 of file `syscon_5411x.h`.

#### 6.46.2.2 `#define SYSCON_AUTOCGOR_RAM0 (1 << 1)`

Auto Clock Gating Override definition bitsRAM0 and RAMX override

Definition at line 576 of file syscon\_5411x.h.

#### 6.46.2.3 `#define SYSCON_AUTOCGOR_RAM1 (1 << 2)`

RAM1 override

Definition at line 577 of file syscon\_5411x.h.

#### 6.46.2.4 `#define SYSCON_AUTOCGOR_RAM2 (1 << 3)`

RAM2 override

Definition at line 578 of file syscon\_5411x.h.

#### 6.46.2.5 `#define SYSCON_FROCTRL_HSPDCLK (1UL << 30)`

High speed clock (FROHF) enable bit

Definition at line 170 of file syscon\_5411x.h.

#### 6.46.2.6 `#define SYSCON_FROCTRL_MASK ((1 << 15) | (0xF << 26))`

FROCTRL register bits.

MASK for reserved bits in FROCTRL register

Definition at line 168 of file syscon\_5411x.h.

#### 6.46.2.7 `#define SYSCON_FROCTRL_SEL96MHZ (1UL << 14)`

When set FROHF will be 96MHz; else FROHF will be 48MHz

Definition at line 173 of file syscon\_5411x.h.

#### 6.46.2.8 `#define SYSCON_FROCTRL_USBCLKADJ (1UL << 24)`

Automatically adjust FRO trim value based on SOF from USB

Definition at line 172 of file syscon\_5411x.h.

#### 6.46.2.9 `#define SYSCON_FROCTRL_USBMODCHG (1UL << 25)`

When set Trim value is pending to be set by SOF from USB

Definition at line 171 of file syscon\_5411x.h.

#### 6.46.2.10 `#define SYSCON_FROCTRL_WRTRIM (1UL << 31)`

Enable Writes to FROCTRL register

Definition at line 169 of file syscon\_5411x.h.

6.46.2.11 `#define SYSCON_NMISRC_M0_ENABLE ((uint32_t) 1 << 30)`

Non-Maskable Interrupt Enable/Disable value Enable the Non-Maskable Interrupt M0 (NMI) source

Definition at line 215 of file `syscon_5411x.h`.

6.46.2.12 `#define SYSCON_NMISRC_M4_ENABLE ((uint32_t) 1 << 31)`

Enable the Non-Maskable Interrupt M4 (NMI) source

Definition at line 216 of file `syscon_5411x.h`.

6.46.2.13 `#define SYSCON_PDRUNCFG_LP_VDDFLASH (1 << 12)`

Flash LP Vdd

Definition at line 458 of file `syscon_5411x.h`.

6.46.2.14 `#define SYSCON_PDRUNCFG_PD_ADC0 (1 << 10)`

ADC0

Definition at line 456 of file `syscon_5411x.h`.

6.46.2.15 `#define SYSCON_PDRUNCFG_PD_BOD_INTR (1 << 8)`

Brown-out Detect interrupt

Definition at line 455 of file `syscon_5411x.h`.

6.46.2.16 `#define SYSCON_PDRUNCFG_PD_BOD_RST (1 << 7)`

Brown-out Detect reset

Definition at line 454 of file `syscon_5411x.h`.

6.46.2.17 `#define SYSCON_PDRUNCFG_PD_FLASH (1 << 5)`

Flash memory

Definition at line 452 of file `syscon_5411x.h`.

6.46.2.18 `#define SYSCON_PDRUNCFG_PD_FRO (1 << 4)`

Power control definition bits (0 = powered, 1 = powered down) FRO oscillator

Definition at line 451 of file `syscon_5411x.h`.

6.46.2.19 `#define SYSCON_PDRUNCFG_PD_ROM (1 << 17)`

ROM

Definition at line 463 of file `syscon_5411x.h`.

6.46.2.20 `#define SYSCON_PDRUNCFG_PD_SRAM0 (1 << 13)`

SRAM0

Definition at line 459 of file syscon\_5411x.h.

6.46.2.21 `#define SYSCON_PDRUNCFG_PD_SRAM1 (1 << 14)`

SRAM1

Definition at line 460 of file syscon\_5411x.h.

6.46.2.22 `#define SYSCON_PDRUNCFG_PD_SRAM2 (1 << 15)`

SRAM2

Definition at line 461 of file syscon\_5411x.h.

6.46.2.23 `#define SYSCON_PDRUNCFG_PD_SRAMX (1 << 16)`

SRAMX

Definition at line 462 of file syscon\_5411x.h.

6.46.2.24 `#define SYSCON_PDRUNCFG_PD_SYS_PLL (1 << 22)`

PLL0

Definition at line 468 of file syscon\_5411x.h.

6.46.2.25 `#define SYSCON_PDRUNCFG_PD_TS (1 << 6)`

Temperature Sensor

Definition at line 453 of file syscon\_5411x.h.

6.46.2.26 `#define SYSCON_PDRUNCFG_PD_USB_PHY (1 << 21)`

USB Phy

Definition at line 467 of file syscon\_5411x.h.

6.46.2.27 `#define SYSCON_PDRUNCFG_PD_VDDA_ENA (1 << 19)`

Vdda to the ADC, must be enabled for the ADC to work

Definition at line 465 of file syscon\_5411x.h.

6.46.2.28 `#define SYSCON_PDRUNCFG_PD_VDDFLASH (1 << 11)`

Flash Vdd

Definition at line 457 of file syscon\_5411x.h.

**6.46.2.29 #define SYSCON\_PDRUNCFG\_PD\_VDDHV\_ENA (1 << 18)**

Vdd HV

Definition at line 464 of file syscon\_5411x.h.

**6.46.2.30 #define SYSCON\_PDRUNCFG\_PD\_VREFP (1 << 23)**

Vrefp to the ADC, must be enabled for the ADC to work

Definition at line 469 of file syscon\_5411x.h.

**6.46.2.31 #define SYSCON\_PDRUNCFG\_PD\_WDT\_OSC (1 << 20)**

Watchdog oscillator

Definition at line 466 of file syscon\_5411x.h.

**6.46.2.32 #define SYSCON\_RST\_BOD (1 << 3)**

Brown-out detect reset status

Definition at line 265 of file syscon\_5411x.h.

**6.46.2.33 #define SYSCON\_RST\_EXTRST (1 << 1)**

External reset status

Definition at line 263 of file syscon\_5411x.h.

**6.46.2.34 #define SYSCON\_RST\_POR (1 << 0)**

System reset status valuesPOR reset status

Definition at line 262 of file syscon\_5411x.h.

**6.46.2.35 #define SYSCON\_RST\_SYSRST (1 << 4)**

software system reset status

Definition at line 266 of file syscon\_5411x.h.

**6.46.2.36 #define SYSCON\_RST\_WDT (1 << 2)**

Watchdog reset status

Definition at line 264 of file syscon\_5411x.h.

**6.46.3 Enumeration Type Documentation****6.46.3.1 enum CHIP\_SYSCON\_BOOT\_MODE\_REMAP\_T**

System memory remap modes used to remap interrupt vectors

Enumerator

***REMAP\_BOOT\_LOADER\_MODE*** Interrupt vectors are re-mapped to Boot ROM

**REMAP\_USER\_RAM\_MODE** Interrupt vectors are re-mapped to user Static RAM

**REMAP\_USER\_FLASH\_MODE** Interrupt vectors are not re-mapped and reside in Flash

Definition at line 177 of file syscon\_5411x.h.

#### 6.46.3.2 enum CHIP\_SYSCON\_PERIPH\_RESET\_T

Peripheral reset identifiers

Enumerator

**RESET\_FLASH** Flash Controller  
**RESET\_FMC** Flash Accelerator  
**RESET\_SPIFI** SPIFI Reset  
**RESET\_MUX** IO MUX Reset  
**RESET\_IOCON** IOCON Reset  
**RESET\_GPIO0** GPIO Port-0 Reset  
**RESET\_GPIO1** GPIO Port-1 Reset  
**RESET\_PINT** Pin Interrupt Reset  
**RESET\_GINT** Group Interrupt Reset  
**RESET\_DMA** DMA Reset  
**RESET\_CRC** CRC Engine Reset  
**RESET\_WWDT** Windowed watchdog timer  
**RESET\_ADC** ADC Reset  
**RESET\_ADC0** ADC Reset  
**RESET\_MRT** Multirate Timer  
**RESET\_SCT0** State configurable Timer  
**RESET\_SCT** State configurable Timer  
**RESET\_UTICK** Micro Tick Timer  
**RESET\_FLEXCOMM0** FlexComm 0  
**RESET\_FLEXCOMM1** FlexComm 1  
**RESET\_FLEXCOMM2** FlexComm 2  
**RESET\_FLEXCOMM3** FlexComm 3  
**RESET\_FLEXCOMM4** FlexComm 4  
**RESET\_FLEXCOMM5** FlexComm 5  
**RESET\_FLEXCOMM6** FlexComm 6  
**RESET\_FLEXCOMM7** FlexComm 7  
**RESET\_DMIC** Digital MIC  
**RESET\_TIMER2** Timer2 Reset  
**RESET\_USB** USB Reset  
**RESET\_TIMER0** Timer0 Reset  
**RESET\_TIMER1** Timer1 Reset  
**RESET\_TIMER3** TIMER0  
**RESET\_TIMER4** TIMER4 Reset

Definition at line 292 of file syscon\_5411x.h.



## 6.46.3.3 enum CHIP\_SYSCON\_WAKEUP\_T

Start enable enumerations - for enabling and disabling peripheral wakeup

Enumerator

***SYSCON\_STARTER\_WWDT\_BOD***  
***SYSCON\_STARTER\_DMA***  
***SYSCON\_STARTER\_GINT0***  
***SYSCON\_STARTER\_GINT1***  
***SYSCON\_STARTER\_PINT0***  
***SYSCON\_STARTER\_PINT1***  
***SYSCON\_STARTER\_PINT2***  
***SYSCON\_STARTER\_PINT3***  
***SYSCON\_STARTER\_UTICK***  
***SYSCON\_STARTER\_MRT***  
***SYSCON\_STARTER\_TIMER0***  
***SYSCON\_STARTER\_TIMER1***  
***SYSCON\_STARTER\_SCT0***  
***SYSCON\_STARTER\_TIMER3***  
***SYSCON\_STARTER\_FLEXCOMM0***  
***SYSCON\_STARTER\_FLEXCOMM1***  
***SYSCON\_STARTER\_FLEXCOMM2***  
***SYSCON\_STARTER\_FLEXCOMM3***  
***SYSCON\_STARTER\_FLEXCOMM4***  
***SYSCON\_STARTER\_FLEXCOMM5***  
***SYSCON\_STARTER\_FLEXCOMM6***  
***SYSCON\_STARTER\_FLEXCOMM7***  
***SYSCON\_STARTER\_ADC0\_SEQA***  
***SYSCON\_STARTER\_ADC0\_SEQB***  
***SYSCON\_STARTER\_ADC0\_THCMP***  
***SYSCON\_STARTER\_DMIC***  
***SYSCON\_STARTER\_HWVAD***  
***SYSCON\_STARTER\_USBNEEDCLK***  
***SYSCON\_STARTER\_USB***  
***SYSCON\_STARTER\_RTC***  
***SYSCON\_STARTER\_RESERVED0***  
***SYSCON\_STARTER\_MAILBOX***  
***SYSCON\_STARTER\_PINT4***  
***SYSCON\_STARTER\_PINT5***  
***SYSCON\_STARTER\_PINT6***  
***SYSCON\_STARTER\_PINT7***  
***SYSCON\_STARTER\_TIMER2***  
***SYSCON\_STARTER\_TIMER4***

Definition at line 502 of file syscon\_5411x.h.

#### 6.46.3.4 enum SYSCON\_FLASHTIM\_T

FLASH Access time definitions.

Enumerator

***SYSCON\_FLASH\_1CYCLE*** Flash accesses use 1 CPU clock  
***FLASHTIM\_20MHZ\_CPU***  
***SYSCON\_FLASH\_2CYCLE*** Flash accesses use 2 CPU clocks  
***SYSCON\_FLASH\_3CYCLE*** Flash accesses use 3 CPU clocks  
***SYSCON\_FLASH\_4CYCLE*** Flash accesses use 4 CPU clocks  
***SYSCON\_FLASH\_5CYCLE*** Flash accesses use 5 CPU clocks  
***SYSCON\_FLASH\_6CYCLE*** Flash accesses use 6 CPU clocks  
***SYSCON\_FLASH\_7CYCLE*** Flash accesses use 7 CPU clocks  
***SYSCON\_FLASH\_8CYCLE*** Flash accesses use 8 CPU clocks

Definition at line 421 of file syscon\_5411x.h.

### 6.46.4 Function Documentation

#### 6.46.4.1 \_\_STATIC\_INLINE void Chip\_SYSCON\_ClearSystemRSTStatus ( uint32\_t reset )

Clear system reset status.

Parameters

<i>reset</i>	: An Or'ed value of SYSCON_RST_* status to clear
--------------	--

Returns

Nothing

Note

This function clears the specified reset source(s).

Definition at line 284 of file syscon\_5411x.h.

#### 6.46.4.2 \_\_STATIC\_INLINE void Chip\_SYSCON\_DisableAutoClocking ( uint32\_t mask )

Disables Auto clock gating for SRAM's.

Parameters

<i>mask</i>	: Mask of RAM bits
-------------	--------------------

Returns

Nothing

Definition at line 586 of file syscon\_5411x.h.

#### 6.46.4.3 void Chip\_SYSCON\_DisableNMISource ( void )

Disable interrupt used for NMI source.

##### Returns

Nothing

Definition at line 79 of file syscon\_5411x.c.

#### 6.46.4.4 \_\_STATIC\_INLINE void Chip\_SYSCON\_DisableWakeup ( CHIP\_SYSCON\_WAKEUP\_T *periphId* )

Disables peripheral's wakeup logic.

##### Parameters

<i>periphId</i>	: Peripheral identifier
-----------------	-------------------------

##### Returns

Nothing

Definition at line 558 of file syscon\_5411x.h.

#### 6.46.4.5 void Chip\_SYSCON\_Enable\_ASYNC\_Syscon ( bool *enable* )

Enable or disable asynchronous APB bridge and subsystem.

##### Parameters

<i>enable</i>	: true to enable, false to disable
---------------	------------------------------------

##### Returns

Nothing

##### Note

This bridge must be enabled to access peripherals on the associated bridge.

Definition at line 89 of file syscon\_5411x.c.

#### 6.46.4.6 \_\_STATIC\_INLINE void Chip\_SYSCON\_EnableAutoClocking ( uint32\_t *mask* )

Re-enables Auto clock gating for SRAM's.

##### Parameters

<i>mask</i>	: Mask of RAM bits
-------------	--------------------

##### Returns

Nothing

Definition at line 596 of file syscon\_5411x.h.

#### 6.46.4.7 void Chip\_SYSCON\_EnableNMISource ( void )

Enable interrupt used for NMI source.

##### Returns

Nothing

Definition at line 69 of file syscon\_5411x.c.

#### 6.46.4.8 \_\_STATIC\_INLINE void Chip\_SYSCON\_EnableWakeup ( CHIP\_SYSCON\_WAKEUP\_T *periphId* )

Enables a pin's (PINT) wakeup logic.

##### Parameters

<i>periphId</i>	: Peripheral identifier (See <a href="#">CHIP_SYSCON_WAKEUP_T</a> )
-----------------	---

##### Returns

Nothing

Definition at line 548 of file syscon\_5411x.h.

#### 6.46.4.9 uint32\_t Chip\_SYSCON\_GetCompFreqMeas ( uint32\_t *refClockRate* )

Returns the computed value for a frequency measurement cycle.

##### Parameters

<i>refClockRate</i>	: Reference clock rate used during the frequency measurement cycle
---------------------	--

##### Returns

Computed capture value

Definition at line 100 of file syscon\_5411x.c.

#### 6.46.4.10 \_\_STATIC\_INLINE uint32\_t Chip\_SYSCON\_GetDeviceID ( void )

Return the pointer to device ID registers.

##### Returns

Pointer to device ID registers

Definition at line 567 of file syscon\_5411x.h.

#### 6.46.4.11 \_\_STATIC\_INLINE CHIP\_SYSCON\_BOOT\_MODE\_REMAP\_T Chip\_SYSCON\_GetMemoryMap ( void )

Get system remap setting.

##### Returns

System remap setting

Definition at line 197 of file syscon\_5411x.h.

6.46.4.12 `__STATIC_INLINE uint32_t Chip_SYSCON_GetPORPIOStatus ( uint8_t port )`

Read POR captured PIO status.

**Parameters**

<i>port</i>	: 0 for port 0 pins, 1 for port 1 pins, 2 for port 2 pins, etc.
-------------	---

**Returns**

captured Power-On-Reset (POR) PIO status

Definition at line 365 of file syscon\_5411x.h.

**6.46.4.13** `__STATIC_INLINE uint32_t Chip_SYSCON_GetPowerStates ( void )`

Power up one or more blocks or peripherals.

**Returns**

OR'ed values of SYSCON\_PDRUNCFG\_\* values

**Note**

A high state indicates the peripheral is powered down.

Definition at line 476 of file syscon\_5411x.h.

**6.46.4.14** `__STATIC_INLINE uint32_t Chip_SYSCON_GetRawFreqMeasCapval ( void )`

Returns the raw capture value for a frequency measurement cycle.

**Returns**

raw capture value (this is not a frequency)

Definition at line 406 of file syscon\_5411x.h.

**6.46.4.15** `__STATIC_INLINE uint32_t Chip_SYSCON_GetResetPIOStatus ( uint8_t port )`

Read reset captured PIO status.

**Parameters**

<i>port</i>	: 0 for port 0 pins, 1 for port 1 pins, 2 for port 2 pins, etc.
-------------	---

**Returns**

captured reset PIO status

**Note**

Used when reset other than a Power-On-Reset (POR) occurs.

Definition at line 376 of file syscon\_5411x.h.

6.46.4.16 `__STATIC_INLINE uint32_t Chip_SYSCON_GetSystemRSTStatus ( void )`

Get system reset status.

#### Returns

An Or'ed value of SYSCON\_RST\_\*

#### Note

This function returns the detected reset source(s).

Definition at line 273 of file syscon\_5411x.h.

6.46.4.17 `__STATIC_INLINE bool Chip_SYSCON_IsFreqMeasComplete ( void )`

Indicates when a frequency measurement cycle is complete.

#### Returns

true if a measurement cycle is active, otherwise false

Definition at line 397 of file syscon\_5411x.h.

6.46.4.18 `__STATIC_INLINE void Chip_SYSCON_Map ( CHIP_SYSCON_BOOT_MODE_REMAP_T remap )`

Re-map interrupt vectors.

#### Parameters

<i>remap</i>	: system memory map value
--------------	---------------------------

#### Returns

Nothing

Definition at line 188 of file syscon\_5411x.h.

6.46.4.19 `__STATIC_INLINE void Chip_SYSCON_PeriphReset ( CHIP_SYSCON_PERIPH_RESET_T periph )`

Resets a peripheral.

#### Parameters

<i>periph</i>	: Peripheral to reset (See <a href="#">CHIP_SYSCON_PERIPH_RESET_T</a> )
---------------	---

#### Returns

Nothing Will assert and de-assert reset for a peripheral.

Definition at line 339 of file syscon\_5411x.h.

6.46.4.20 `__STATIC_INLINE void Chip_SYSCON_PowerDown ( uint32_t powerdownmask )`

Power down one or more blocks or peripherals.

## Parameters

<i>powerdownmask</i>	: OR'ed values of SYSCON_PDRUNCFG_* values
----------------------	--

## Returns

Nothing

Definition at line 486 of file syscon\_5411x.h.

#### 6.46.4.21 void Chip\_SYSCON\_PowerUp ( uint32\_t *powerupmask* )

Power up one or more blocks or peripherals.

## Parameters

<i>powerupmask</i>	: OR'ed values of SYSCON_PDRUNCFG_* values
--------------------	--

## Returns

Nothing

Definition at line 116 of file syscon\_5411x.c.

#### 6.46.4.22 \_\_STATIC\_INLINE void Chip\_SYSCON\_SetFLASHAccess ( SYSCON\_FLASHTIM\_T *c/ks* )

Set FLASH memory access time in clocks.

## Parameters

<i>c/ks</i>	: Clock cycles for FLASH access
-------------	---------------------------------

## Returns

Nothing

Definition at line 438 of file syscon\_5411x.h.

#### 6.46.4.23 void Chip\_SYSCON\_SetNMISource ( uint32\_t *intsrc* )

Set source for non-maskable interrupt (NMI)

## Parameters

<i>intsrc</i>	: IRQ number to assign to the NMI
---------------	-----------------------------------

## Returns

Nothing

## Note

The NMI source will be disabled upon exiting this function. Use the [Chip\\_SYSCON\\_EnableNMISource\(\)](#) function to enable the NMI source.

Definition at line 51 of file syscon\_5411x.c.

#### 6.46.4.24 \_\_STATIC\_INLINE void Chip\_SYSCON\_SetSYSTCKCAL ( uint32\_t *sysCalVal* )

Set System tick timer calibration value.



## Parameters

<i>sysCalVal</i>	: System tick timer calibration value
------------------	---------------------------------------

## Returns

Nothing

Definition at line 207 of file syscon\_5411x.h.

#### 6.46.4.25 `__STATIC_INLINE void Chip_SYSCON_SetUSARTFRGCtrl ( uint8_t fmul, uint8_t fdiv )`

Set UART Fractional divider value.

## Parameters

<i>fmul</i>	: Fractional multiplier value
<i>fdiv</i>	: Fractional divider value (Must always be 0xFF)

## Returns

Nothing

Definition at line 254 of file syscon\_5411x.h.

#### 6.46.4.26 `__STATIC_INLINE void Chip_SYSCON_StartFreqMeas ( void )`

Starts a frequency measurement cycle.

## Returns

Nothing

## Note

This function is meant to be used with the [Chip\\_INMUX\\_SetFreqMeasRefClock\(\)](#) and [Chip\\_INMUX\\_SetFreqMeasTargClock\(\)](#) functions.

Definition at line 387 of file syscon\_5411x.h.

## 6.47 CHIP: LPC5411X UART Driver

### 6.47.1 Detailed Description

#### Data Structures

- struct [LPC\\_USART\\_T](#)  
*UART Registers.*
- struct [UART\\_BAUD\\_T](#)  
*UART Baud rate calculation structure.*
- struct [UART\\_STATISTICS\\_T](#)  
*UART statistics structure.*

#### Macros

- `#define ECHO_EN 1`
- `#define ECHO_DIS 0`
- `#define UART_CFG_BITMASK (0x00fddbfd)`  
*UART CFG register definitions.*
- `#define UART_CFG_ENABLE (0x01 << 0)`
- `#define UART_CFG_DATALEN_7 (0x00 << 2)`
- `#define UART_CFG_DATALEN_8 (0x01 << 2)`
- `#define UART_CFG_DATALEN_9 (0x02 << 2)`
- `#define UART_CFG_PARITY_NONE (0x00 << 4)`
- `#define UART_CFG_PARITY_EVEN (0x02 << 4)`
- `#define UART_CFG_PARITY_ODD (0x03 << 4)`
- `#define UART_CFG_STOPLEN_1 (0x00 << 6)`
- `#define UART_CFG_STOPLEN_2 (0x01 << 6)`
- `#define UART_CFG_MODE32K (0x01 << 7)`
- `#define UART_CFG_LINMODE (0x01 << 8)`
- `#define UART_CFG_CTSSEN (0x01 << 9)`
- `#define UART_CFG_SYNCEN (0x01 << 11)`
- `#define UART_CFG_CLKPOL (0x01 << 12)`
- `#define UART_CFG_SYNCMST (0x01 << 14)`
- `#define UART_CFG_LOOP (0x01 << 15)`
- `#define UART_CFG_IOMODE (0x01 << 16)`
- `#define UART_CFG_OETA (0x01 << 18)`
- `#define UART_CFG_AUTOADDR (0x01 << 19)`
- `#define UART_CFG_OESEL (0x01 << 20)`
- `#define UART_CFG_OEPOL (0x01 << 21)`
- `#define UART_CFG_RXPOL (0x01 << 22)`
- `#define UART_CFG_TXPOL (0x01 << 23)`
- `#define UART_CTRL_TXBRKEN (0x01 << 1)`  
*UART CTRL register definitions.*
- `#define UART_CTRL_ADDRDET (0x01 << 2)`
- `#define UART_CTRL_TXDIS (0x01 << 6)`
- `#define UART_CTRL_CC (0x01 << 8)`
- `#define UART_CTRL_CLRCONRX (0x01 << 9)`
- `#define UART_CTRL_AUTOBAUD (0x01 << 16)`
- `#define UART_STAT_RXIDLE (0x01 << 1)`  
*UART STAT register definitions.*
- `#define UART_STAT_TXIDLE (0x01 << 3)`
- `#define UART_STAT_CTS (0x01 << 4)`

- #define `UART_STAT_DELTACTS` (0x01 << 5)
- #define `UART_STAT_TXDISINT` (0x01 << 6)
- #define `UART_STAT_RXBRK` (0x01 << 10)
- #define `UART_STAT_DELTARXBRK` (0x01 << 11)
- #define `UART_STAT_START` (0x01 << 12)
- #define `UART_STAT_FRM_ERRINT` (0x01 << 13)
- #define `UART_STAT_PAR_ERRINT` (0x01 << 14)
- #define `UART_STAT_RXNOISEINT` (0x01 << 15)
- #define `UART_STAT_ABERR` (0x01 << 16)
- #define `UART_INT_TXIDLE` (0x01 << 3)

*UART INTENSET/INTENCLR/INTSTAT register definitions.*

- #define `UART_INT_DELTACTS` (0x01 << 5)
- #define `UART_INT_TXDIS` (0x01 << 6)
- #define `UART_INT_DELTARXBRK` (0x01 << 11)
- #define `UART_INT_START` (0x01 << 12)
- #define `UART_INT_FRAMERR` (0x01 << 13)
- #define `UART_INT_PARITYERR` (0x01 << 14)
- #define `UART_INT_RXNOISE` (0x01 << 15)
- #define `UART_INT_ABERR` (0x01 << 16)
- #define `UART_FIFOCFG_BITMASK` (0x7F033)

*UART FIFO Configuration register bits.*

- #define `UART_FIFOCFG_ENABLETX` (1 << 0)
- #define `UART_FIFOCFG_ENABLERX` (1 << 1)
- #define `UART_FIFOCFG_DMATX` (1 << 12)
- #define `UART_FIFOCFG_DMARX` (1 << 13)
- #define `UART_FIFOCFG_WAKETX` (1 << 14)
- #define `UART_FIFOCFG_WAKERX` (1 << 15)
- #define `UART_FIFOCFG_EMPTYTX` (1 << 16)
- #define `UART_FIFOCFG_EMPTYRX` (1 << 17)
- #define `UART_FIFO_DEPTH` (16) /\*\* UART-FIFO How many entries are in the FIFO \*/

*UART FIFO Status register defines.*

- #define `UART_FIFOSTAT_BITMASK` (0x1F1FFB) /\*\* UART-FIFO STAT Register BitMask \*/
- #define `UART_FIFOSTAT_TXERR` (1 << 0)
- #define `UART_FIFOSTAT_RXERR` (1 << 1)
- #define `UART_FIFOSTAT_PERIPH` (1 << 3)
- #define `UART_FIFOSTAT_TXEMPTY` (1 << 4)
- #define `UART_FIFOSTAT_TXNOTFULL` (1 << 5)
- #define `UART_FIFOSTAT_RXNOTEMPTY` (1 << 6)
- #define `UART_FIFOSTAT_RXFULL` (1 << 7)
- #define `UART_FIFOSTAT_TXLVL`(lvl) (((lvl) >> 8) & 0x1F)
- #define `UART_FIFOSTAT_RXLVL`(lvl) (((lvl) >> 16) & 0x1F)
- #define `UART_FIFOTRIG_BITMASK` (0x000f0f03) /\*\* UART FIFO trigger settings Register BitMask \*/

*UART FIFO trigger settings register defines.*

- #define `UART_FIFOTRIG_TXLVLENA` (1 << 0)
- #define `UART_FIFOTRIG_RXLVLENA` (1 << 1)
- #define `UART_FIFOTRIG_TXLVL`(lvl) ((lvl & 0x0f) << 8)
- #define `UART_FIFOTRIG_RXLVL`(lvl) ((lvl & 0x0f) << 16)
- #define `UART_FIFOINT_BITMASK` (0x001F) /\*\* FIFO interrupt Bit mask \*/

*UART FIFO Interrupt enable/disable/status [INTSET/INTCLR/INTSTAT and FIFOINTSET/FIFOINTCLR/FIFOINTSTAT registers].*

- #define `UART_FIFOINT_TXERR` (1 << 0) /\*\* TX error interrupt [BIT-0 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/
- #define `UART_FIFOINT_RXERR` (1 << 1) /\*\* RX error interrupt [BIT-1 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/

- `#define UART_FIFOINT_TXLVL (1 << 2) /** TX FIFO ready interrupt [BIT-2 of FIFOINTENSET/FIFOINT←ENCLR/FIFOINTSTAT register] */`
- `#define UART_FIFOINT_RXLVL (1 << 3) /** RX Data ready interrupt [BIT-3 of FIFOINTENSET/FIFOINT←ENCLR/FIFOINTSTAT register] */`
- `#define UART_FIFOINT_PERINT (1 << 4) /** UART peripheral interrupt [BIT-4 of FIFOINTSTAT register] */`

## Functions

- `__STATIC_INLINE void Chip_UART_Enable (LPC_USART_T *pUART)`  
*Enable the UART.*
- `__STATIC_INLINE void Chip_UART_Disable (LPC_USART_T *pUART)`  
*Disable the UART.*
- `__STATIC_INLINE void Chip_UART_TXEnable (LPC_USART_T *pUART)`  
*Enable transmission on UART TxD pin.*
- `__STATIC_INLINE void Chip_UART_TXDisable (LPC_USART_T *pUART)`  
*Disable transmission on UART TxD pin.*
- `__STATIC_INLINE uint32_t Chip_UART_AutoBaud (LPC_USART_T *pUART)`  
*Set auto baud.*
- `__STATIC_INLINE void Chip_UART_SendByte (LPC_USART_T *pUART, uint8_t data)`  
*Transmit a single data byte through the UART peripheral.*
- `__STATIC_INLINE uint32_t Chip_UART_ReadByte (LPC_USART_T *pUART)`  
*Read a single byte data from the UART peripheral.*
- `__STATIC_INLINE void Chip_UART_IntEnable (LPC_USART_T *pUART, uint32_t intMask)`  
*Enable UART interrupts.*
- `__STATIC_INLINE void Chip_UART_IntDisable (LPC_USART_T *pUART, uint32_t intMask)`  
*Disable UART interrupts.*
- `__STATIC_INLINE uint32_t Chip_UART_GetIntsEnabled (LPC_USART_T *pUART)`  
*Returns UART interrupts that are enabled.*
- `__STATIC_INLINE uint32_t Chip_UART_GetIntStatus (LPC_USART_T *pUART)`  
*Get UART interrupt status.*
- `__STATIC_INLINE void Chip_UART_ConfigData (LPC_USART_T *pUART, uint32_t config)`  
*Configure data width, parity and stop bits.*
- `__STATIC_INLINE uint32_t Chip_UART_GetStatus (LPC_USART_T *pUART)`  
*Get the UART status register.*
- `__STATIC_INLINE void Chip_UART_ClearStatus (LPC_USART_T *pUART, uint32_t stsMask)`  
*Clear the UART status register.*
- `__STATIC_INLINE uint32_t Chip_UART_GetFIFOStatus (LPC_USART_T *pUART)`  
*Get the current status of UART controller FIFO.*
- `__STATIC_INLINE void Chip_UART_ClearFIFOStatus (LPC_USART_T *pUART, uint32_t mask)`  
*Clear the FIFO status register.*
- `__STATIC_INLINE void Chip_UART_SetFIFOTrigLevel (LPC_USART_T *pUART, uint8_t tx_lvl, uint8_t rx←_lvl)`  
*Setup the trigger level for UART FIFO.*
- `__STATIC_INLINE void Chip_UART_EnableFIFOInts (LPC_USART_T *pUART, uint32_t intMask)`  
*Enable a UART FIFO interrupt.*
- `__STATIC_INLINE void Chip_UART_DisableFIFOInts (LPC_USART_T *pUART, uint32_t intMask)`  
*Disable a UART FIFO interrupt.*
- `__STATIC_INLINE uint32_t Chip_UART_GetFIFOEnabledInts (LPC_USART_T *pUART)`  
*Return enabled UART FIFO interrupts.*
- `__STATIC_INLINE uint32_t Chip_UART_GetFIFOPendingInts (LPC_USART_T *pUART)`

- Return pending UART FIFO interrupts.*
- `__STATIC_INLINE void Chip_UART_SetFIFOCfg (LPC_USART_T *pUART, uint32_t cfg)`  
*Set FIFO Configuration register.*
- `__STATIC_INLINE void Chip_UART_ClearFIFOCfg (LPC_USART_T *pUART, uint32_t cfg)`  
*Clear FIFO Configuration register.*
- `__STATIC_INLINE void Chip_UART_FlushFIFOs (LPC_USART_T *pUART)`  
*Flush FIFOs.*
- `int Chip_UART_Init (LPC_USART_T *pUART)`  
*Initialize the UART peripheral.*
- `void Chip_UART_DeInit (LPC_USART_T *pUART)`  
*Deinitialize the UART peripheral.*
- `void Chip_UART_ConfigDMA (LPC_USART_T *pUART)`  
*Configure UART for DMA.*
- `int Chip_UART_Send (LPC_USART_T *pUART, const void *data, int numBytes)`  
*Transmit a byte array through the UART peripheral (non-blocking)*
- `int Chip_UART_Read (LPC_USART_T *pUART, void *data, int numBytes)`  
*Read data through the UART peripheral (non-blocking)*
- `uint32_t Chip_UART_SetBaud (LPC_USART_T *pUART, uint32_t baudrate)`  
*Set baud rate for UART.*
- `int Chip_UART_SendBlocking (LPC_USART_T *pUART, const void *data, int numBytes)`  
*Transmit a byte array through the UART peripheral (blocking)*
- `int Chip_UART_ReadBlocking (LPC_USART_T *pUART, void *data, int numBytes)`  
*Read data through the UART peripheral (blocking)*
- `void Chip_UART_RXIntHandlerRB (LPC_USART_T *pUART, RINGBUFF_T *pRB)`  
*UART receive-only interrupt handler for ring buffers.*
- `void Chip_UART_TXIntHandlerRB (LPC_USART_T *pUART, RINGBUFF_T *pRB)`  
*UART transmit-only interrupt handler for ring buffers.*
- `uint32_t Chip_UART_SendRB (LPC_USART_T *pUART, RINGBUFF_T *pRB, const void *data, int count)`  
*Populate a transmit ring buffer and start UART transmit.*
- `int Chip_UART_ReadRB (LPC_USART_T *pUART, RINGBUFF_T *pRB, void *data, int bytes)`  
*Copy data from a receive ring buffer.*
- `void Chip_UART_IRQHandlerRB (LPC_USART_T *pUART, UART_STATISTICS_T *statistics, RINGBUFF_T *pRXRB, RINGBUFF_T *pTXRB)`  
*UART receive/transmit interrupt handler for ring buffers.*
- `void Chip_UART_IRQHandlerDMA (LPC_USART_T *pUART, UART_STATISTICS_T *statistics)`  
*UART receive/transmit interrupt handler for DMA.*

## 6.47.2 Macro Definition Documentation

### 6.47.2.1 `#define ECHO_DIS 0`

Definition at line 50 of file `uart_5411x.h`.

### 6.47.2.2 `#define ECHO_EN 1`

Definition at line 49 of file `uart_5411x.h`.

### 6.47.2.3 `#define UART_CFG_AUTOADDR (0x01 << 19)`

Automatic Address matching enable

Definition at line 109 of file `uart_5411x.h`.

#### 6.47.2.4 `#define UART_CFG_BITMASK (0x00fdbbfd)`

UART CFG register definitions.

Definition at line 90 of file `uart_5411x.h`.

#### 6.47.2.5 `#define UART_CFG_CLKPOL (0x01 << 12)`

Un\_RXD rising edge sample enable bit

Definition at line 104 of file `uart_5411x.h`.

#### 6.47.2.6 `#define UART_CFG_CTSEN (0x01 << 9)`

CTS enable bit

Definition at line 102 of file `uart_5411x.h`.

#### 6.47.2.7 `#define UART_CFG_DATALEN_7 (0x00 << 2)`

UART 7 bit length mode

Definition at line 92 of file `uart_5411x.h`.

#### 6.47.2.8 `#define UART_CFG_DATALEN_8 (0x01 << 2)`

UART 8 bit length mode

Definition at line 93 of file `uart_5411x.h`.

#### 6.47.2.9 `#define UART_CFG_DATALEN_9 (0x02 << 2)`

UART 9 bit length mode

Definition at line 94 of file `uart_5411x.h`.

#### 6.47.2.10 `#define UART_CFG_ENABLE (0x01 << 0)`

Definition at line 91 of file `uart_5411x.h`.

#### 6.47.2.11 `#define UART_CFG_IOMODE (0x01 << 16)`

enable bit standard UART / IrDA UART

Definition at line 107 of file `uart_5411x.h`.

#### 6.47.2.12 `#define UART_CFG_LINMODE (0x01 << 8)`

UART LIN MODE

Definition at line 101 of file `uart_5411x.h`.

**6.47.2.13 #define UART\_CFG\_LOOP (0x01 << 15)**

Loopback mode enable bit

Definition at line 106 of file uart\_5411x.h.

**6.47.2.14 #define UART\_CFG\_MODE32K (0x01 << 7)**

UART 32K MODE

Definition at line 100 of file uart\_5411x.h.

**6.47.2.15 #define UART\_CFG\_OEPOL (0x01 << 21)**

Output Enable Polarity

Definition at line 111 of file uart\_5411x.h.

**6.47.2.16 #define UART\_CFG\_OESEL (0x01 << 20)**

Output Enable Select

Definition at line 110 of file uart\_5411x.h.

**6.47.2.17 #define UART\_CFG\_OETA (0x01 << 18)**

Output Enable Turnaround time enable for RS-485 operation

Definition at line 108 of file uart\_5411x.h.

**6.47.2.18 #define UART\_CFG\_PARITY\_EVEN (0x02 << 4)**

Even parity

Definition at line 96 of file uart\_5411x.h.

**6.47.2.19 #define UART\_CFG\_PARITY\_NONE (0x00 << 4)**

No parity

Definition at line 95 of file uart\_5411x.h.

**6.47.2.20 #define UART\_CFG\_PARITY\_ODD (0x03 << 4)**

Odd parity

Definition at line 97 of file uart\_5411x.h.

**6.47.2.21 #define UART\_CFG\_RXPOL (0x01 << 22)**

Receive data polarity

Definition at line 112 of file uart\_5411x.h.

**6.47.2.22 #define UART\_CFG\_STOPLEN\_1 (0x00 << 6)**

UART One Stop Bit Select

Definition at line 98 of file uart\_5411x.h.

**6.47.2.23 #define UART\_CFG\_STOPLEN\_2 (0x01 << 6)**

UART Two Stop Bits Select

Definition at line 99 of file uart\_5411x.h.

**6.47.2.24 #define UART\_CFG\_SYNCEN (0x01 << 11)**

Synchronous mode enable bit

Definition at line 103 of file uart\_5411x.h.

**6.47.2.25 #define UART\_CFG\_SYNCMST (0x01 << 14)**

Select master mode (synchronous mode) enable bit

Definition at line 105 of file uart\_5411x.h.

**6.47.2.26 #define UART\_CFG\_TXPOL (0x01 << 23)**

Transmit data polarity

Definition at line 113 of file uart\_5411x.h.

**6.47.2.27 #define UART\_CTRL\_ADDRDET (0x01 << 2)**

Address detect mode enable bit

Definition at line 120 of file uart\_5411x.h.

**6.47.2.28 #define UART\_CTRL\_AUTOBAUD (0x01 << 16)**

Auto baud bit

Definition at line 124 of file uart\_5411x.h.

**6.47.2.29 #define UART\_CTRL\_CC (0x01 << 8)**

Continuous Clock mode enable bit

Definition at line 122 of file uart\_5411x.h.

**6.47.2.30 #define UART\_CTRL\_CLRCCONRX (0x01 << 9)**

Clear Continuous Clock bit

Definition at line 123 of file uart\_5411x.h.



**6.47.2.31 #define UART\_CTRL\_TXBRKEN (0x01 << 1)**

UART CTRL register definitions.

Continuous break enable bit

Definition at line 119 of file uart\_5411x.h.

**6.47.2.32 #define UART\_CTRL\_TXDIS (0x01 << 6)**

Transmit disable bit

Definition at line 121 of file uart\_5411x.h.

**6.47.2.33 #define UART\_FIFO\_DEPTH (16) /\*\* UART-FIFO How many entries are in the FIFO \*/**

UART FIFO Status register defines.

Definition at line 171 of file uart\_5411x.h.

**6.47.2.34 #define UART\_FIOCFG\_BITMASK (0x7F033)**

UART FIFO Configuration register bits.

Register mask bit

Definition at line 158 of file uart\_5411x.h.

**6.47.2.35 #define UART\_FIOCFG\_DMARX (1 << 13)**

Enable DMA RX

Definition at line 162 of file uart\_5411x.h.

**6.47.2.36 #define UART\_FIOCFG\_DMATX (1 << 12)**

Enable DMA TX

Definition at line 161 of file uart\_5411x.h.

**6.47.2.37 #define UART\_FIOCFG\_EMPTYRX (1 << 17)**

Empty the RX FIFO

Definition at line 166 of file uart\_5411x.h.

**6.47.2.38 #define UART\_FIOCFG\_EMPTYTX (1 << 16)**

Empty the TX FIFO

Definition at line 165 of file uart\_5411x.h.

**6.47.2.39 #define UART\_FIOCFG\_ENABLERX (1 << 1)**

Enable RX FIFO

Definition at line 160 of file uart\_5411x.h.

**6.47.2.40 #define UART\_FIFOCFG\_ENABLETX (1 << 0)**

Enable TX FIFO

Definition at line 159 of file uart\_5411x.h.

**6.47.2.41 #define UART\_FIFOCFG\_WAKERX (1 << 15)**

Enable wakeup triggered by RX

Definition at line 164 of file uart\_5411x.h.

**6.47.2.42 #define UART\_FIFOCFG\_WAKETX (1 << 14)**

Enable wakeup triggered by TX

Definition at line 163 of file uart\_5411x.h.

**6.47.2.43 #define UART\_FIFOINT\_BITMASK (0x001F) /\*\* FIFO interrupt Bit mask \*/**

UART FIFO Interrupt enable/disable/status [INTSET/INTCLR/INTSTAT and FIFOINTSET/FIFOINTCLR/FIFOINTSTAT registers].

Definition at line 195 of file uart\_5411x.h.

**6.47.2.44 #define UART\_FIFOINT\_PERINT (1 << 4) /\*\* UART peripheral interrupt [BIT-4 of FIFOINTSTAT register] \*/**

Definition at line 200 of file uart\_5411x.h.

**6.47.2.45 #define UART\_FIFOINT\_RXERR (1 << 1) /\*\* RX error interrupt [BIT-1 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/**

Definition at line 197 of file uart\_5411x.h.

**6.47.2.46 #define UART\_FIFOINT\_RXLVL (1 << 3) /\*\* RX Data ready interrupt [BIT-3 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/**

Definition at line 199 of file uart\_5411x.h.

**6.47.2.47 #define UART\_FIFOINT\_TXERR (1 << 0) /\*\* TX error interrupt [BIT-0 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/**

Definition at line 196 of file uart\_5411x.h.

**6.47.2.48 #define UART\_FIFOINT\_TXLVL (1 << 2) /\*\* TX FIFO ready interrupt [BIT-2 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/**

Definition at line 198 of file uart\_5411x.h.

**6.47.2.49 #define UART\_FIFOSTAT\_BITMASK (0x1F1FFB) /\*\* UART-FIFO STAT Register BitMask \*/**

Definition at line 172 of file uart\_5411x.h.

**6.47.2.50 #define UART\_FIFOSTAT\_PERIPH (1 << 3)**

Peripheral interrupt

Definition at line 175 of file uart\_5411x.h.

**6.47.2.51 #define UART\_FIFOSTAT\_RXERR (1 << 1)**

UART RX Error

Definition at line 174 of file uart\_5411x.h.

**6.47.2.52 #define UART\_FIFOSTAT\_RXFULL (1 << 7)**

RX FIFO Full

Definition at line 179 of file uart\_5411x.h.

**6.47.2.53 #define UART\_FIFOSTAT\_RXLVL( lvl ) (((lvl) >> 16) & 0x1F)**

Get TX Level from status

Definition at line 181 of file uart\_5411x.h.

**6.47.2.54 #define UART\_FIFOSTAT\_RXNOTEMPTY (1 << 6)**

RXFIFO Not empty

Definition at line 178 of file uart\_5411x.h.

**6.47.2.55 #define UART\_FIFOSTAT\_TXEMPTY (1 << 4)**

TXFIFO Empty

Definition at line 176 of file uart\_5411x.h.

**6.47.2.56 #define UART\_FIFOSTAT\_TXERR (1 << 0)**

UART TX Error

Definition at line 173 of file uart\_5411x.h.

**6.47.2.57 #define UART\_FIFOSTAT\_TXLVL( lvl ) (((lvl) >> 8) & 0x1F)**

Get TX Level from status

Definition at line 180 of file uart\_5411x.h.

**6.47.2.58 #define UART\_FIFOSTAT\_TXNOTFULL (1 << 5)**

TXFIFO Not Full

Definition at line 177 of file uart\_5411x.h.

**6.47.2.59** `#define UART_FIFOTRIG_BITMASK (0x000f0f03) /** UART FIFO trigger settings Register BitMask */`

UART FIFO trigger settings register defines.

Definition at line 186 of file `uart_5411x.h`.

**6.47.2.60** `#define UART_FIFOTRIG_RXLVL( lvl ) ((lvl & 0x0f) << 16)`

Set RX Level trigger

Definition at line 190 of file `uart_5411x.h`.

**6.47.2.61** `#define UART_FIFOTRIG_RXLVLENA (1 << 1)`

RX level enable

Definition at line 188 of file `uart_5411x.h`.

**6.47.2.62** `#define UART_FIFOTRIG_TXLVL( lvl ) ((lvl & 0x0f) << 8)`

Set TX Level trigger

Definition at line 189 of file `uart_5411x.h`.

**6.47.2.63** `#define UART_FIFOTRIG_TXLVLENA (1 << 0)`

TX level enable

Definition at line 187 of file `uart_5411x.h`.

**6.47.2.64** `#define UART_INT_ABERR (0x01 << 16)`

Auto-baud error interrupt

Definition at line 153 of file `uart_5411x.h`.

**6.47.2.65** `#define UART_INT_DELTACTS (0x01 << 5)`

Change in CTS state interrupt

Definition at line 146 of file `uart_5411x.h`.

**6.47.2.66** `#define UART_INT_DELTARXBRK (0x01 << 11)`

Change in receiver break detection interrupt

Definition at line 148 of file `uart_5411x.h`.

**6.47.2.67** `#define UART_INT_FRAMERR (0x01 << 13)`

Frame error interrupt

Definition at line 150 of file `uart_5411x.h`.

**6.47.2.68 #define UART\_INT\_PARITYERR (0x01 << 14)**

Parity error interrupt

Definition at line 151 of file uart\_5411x.h.

**6.47.2.69 #define UART\_INT\_RXNOISE (0x01 << 15)**

Received noise interrupt

Definition at line 152 of file uart\_5411x.h.

**6.47.2.70 #define UART\_INT\_START (0x01 << 12)**

Start detect interrupt

Definition at line 149 of file uart\_5411x.h.

**6.47.2.71 #define UART\_INT\_TXDIS (0x01 << 6)**

Transmitter disable interrupt

Definition at line 147 of file uart\_5411x.h.

**6.47.2.72 #define UART\_INT\_TXIDLE (0x01 << 3)**

UART INTENSET/INTENCLR/INTSTAT register definitions.

Transmit idle interrupt

Definition at line 145 of file uart\_5411x.h.

**6.47.2.73 #define UART\_STAT\_ABERR (0x01 << 16)**

Auto baud error flag

Definition at line 140 of file uart\_5411x.h.

**6.47.2.74 #define UART\_STAT\_CTS (0x01 << 4)**

Status of CTS signal

Definition at line 131 of file uart\_5411x.h.

**6.47.2.75 #define UART\_STAT\_DELTACTS (0x01 << 5)**

Change in CTS state

Definition at line 132 of file uart\_5411x.h.

**6.47.2.76 #define UART\_STAT\_DELTARXBRK (0x01 << 11)**

Change in receive break detection

Definition at line 135 of file uart\_5411x.h.

**6.47.2.77 #define UART\_STAT\_FRM\_ERRINT (0x01 << 13)**

Framing Error interrupt flag

Definition at line 137 of file uart\_5411x.h.

**6.47.2.78 #define UART\_STAT\_PAR\_ERRINT (0x01 << 14)**

Parity Error interrupt flag

Definition at line 138 of file uart\_5411x.h.

**6.47.2.79 #define UART\_STAT\_RXBRK (0x01 << 10)**

Received break

Definition at line 134 of file uart\_5411x.h.

**6.47.2.80 #define UART\_STAT\_RXIDLE (0x01 << 1)**

UART STAT register definitions.

Receiver idle

Definition at line 129 of file uart\_5411x.h.

**6.47.2.81 #define UART\_STAT\_RXNOISEINT (0x01 << 15)**

Received Noise interrupt flag

Definition at line 139 of file uart\_5411x.h.

**6.47.2.82 #define UART\_STAT\_START (0x01 << 12)**

Start detected

Definition at line 136 of file uart\_5411x.h.

**6.47.2.83 #define UART\_STAT\_TXDISINT (0x01 << 6)**

Transmitter disabled

Definition at line 133 of file uart\_5411x.h.

**6.47.2.84 #define UART\_STAT\_TXIDLE (0x01 << 3)**

Transmitter idle

Definition at line 130 of file uart\_5411x.h.

**6.47.3 Function Documentation****6.47.3.1 \_\_STATIC\_INLINE uint32\_t Chip\_UART\_AutoBaud ( LPC\_USART\_T \* pUART )**

Set auto baud.

## Parameters

<i>pUART</i>	: Pointer to selected pUART peripheral
--------------	--

## Returns

true if auto baud succeeds, false if fails

Definition at line 284 of file uart\_5411x.h.

### 6.47.3.2 `__STATIC_INLINE void Chip_UART_ClearFIFOCfg ( LPC_USART_T * pUART, uint32_t cfg )`

Clear FIFO Configuration register.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
<i>cfg</i>	: Configuration value mask (OR'ed UART_FIFOCFG_* values like <a href="#">UART_FIFOCFG_ENA</a> ↵ <a href="#">BLETX</a> )

## Returns

Nothing

Definition at line 516 of file uart\_5411x.h.

### 6.47.3.3 `__STATIC_INLINE void Chip_UART_ClearFIFOStatus ( LPC_USART_T * pUART, uint32_t mask )`

Clear the FIFO status register.

## Parameters

<i>pUART</i>	: The base of the UART peripheral on the chip
<i>mask</i>	: Mask of the status bits that needs to be cleared

## Returns

Nothing

Definition at line 436 of file uart\_5411x.h.

### 6.47.3.4 `__STATIC_INLINE void Chip_UART_ClearStatus ( LPC_USART_T * pUART, uint32_t stsMask )`

Clear the UART status register.

## Parameters

<i>pUART</i>	: Pointer to selected UARTx peripheral
<i>stsMask</i>	: OR'ed statuses to disable

## Returns

Nothing

## Note

Multiple interrupts may be pending. Mask the return value with one or more UART\_INTEN\_\* definitions to determine pending interrupts.

Definition at line 413 of file uart\_5411x.h.

6.47.3.5 `__STATIC_INLINE void Chip_UART_ConfigData ( LPC_USART_T * pUART, uint32_t config )`

Configure data width, parity and stop bits.



## Parameters

<i>pUART</i>	: Pointer to selected pUART peripheral
<i>config</i>	: UART configuration, OR'ed values of select UART_CFG_* defines

## Returns

Nothing

## Note

Select OR'ed config options for the UART from the UART\_CFG\_PARITY\_\*, UART\_CFG\_STOPLEN\_\*, and UART\_CFG\_DATALEN\_\* definitions. For example, a configuration of 8 data bits, 1 stop bit, and even (enabled) parity would be (UART\_CFG\_DATALEN\_8 | UART\_CFG\_STOPLEN\_1 | UART\_CFG\_PARITY\_EV←EN). Will not alter other bits in the CFG register.

Definition at line 386 of file uart\_5411x.h.

#### 6.47.3.6 void Chip\_UART\_ConfigDMA ( LPC\_USART\_T \* pUART )

Configure UART for DMA.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
--------------	---

## Returns

Nothing

Definition at line 174 of file uart\_5411x.c.

#### 6.47.3.7 void Chip\_UART\_DeInit ( LPC\_USART\_T \* pUART )

Deinitialize the UART peripheral.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
--------------	---

## Returns

Nothing

Definition at line 168 of file uart\_5411x.c.

#### 6.47.3.8 \_\_STATIC\_INLINE void Chip\_UART\_Disable ( LPC\_USART\_T \* pUART )

Disable the UART.

## Parameters

<i>pUART</i>	: Pointer to selected UARTx peripheral
--------------	--

## Returns

Nothing

Definition at line 254 of file uart\_5411x.h.

6.47.3.9 `__STATIC_INLINE void Chip_UART_DisableFIFOInts ( LPC_USART_T * pUART, uint32_t intMask )`

Disable a UART FIFO interrupt.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
<i>intMask</i>	: Or'ed value of UART_FIFOINT_* values to disable (See <a href="#">UART_FIFOINT_BITMASK</a> )

## Returns

Nothing

Definition at line 474 of file uart\_5411x.h.

#### 6.47.3.10 `__STATIC_INLINE void Chip_UART_Enable ( LPC_USART_T * pUART )`

Enable the UART.

## Parameters

<i>pUART</i>	: Pointer to selected UARTx peripheral
--------------	--

## Returns

Nothing

Definition at line 244 of file uart\_5411x.h.

#### 6.47.3.11 `__STATIC_INLINE void Chip_UART_EnableFIFOInts ( LPC_USART_T * pUART, uint32_t intMask )`

Enable a UART FIFO interrupt.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
<i>intMask</i>	: Or'ed value of UART_FIFOINT_* values to enable

## Returns

Nothing

Definition at line 463 of file uart\_5411x.h.

#### 6.47.3.12 `__STATIC_INLINE void Chip_UART_FlushFIFOs ( LPC_USART_T * pUART )`

Flush FIFOs.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
--------------	---

## Returns

Nothing

Definition at line 526 of file uart\_5411x.h.

#### 6.47.3.13 `__STATIC_INLINE uint32_t Chip_UART_GetFIFOEnabledInts ( LPC_USART_T * pUART )`

Return enabled UART FIFO interrupts.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
--------------	---

## Returns

An Or'ed value of UART\_FIFINT\_\* values

Definition at line 484 of file uart\_5411x.h.

#### 6.47.3.14 `__STATIC_INLINE uint32_t Chip_UART_GetFIFOPendingInts ( LPC_USART_T * pUART )`

Return pending UART FIFO interrupts.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
--------------	---

## Returns

An Or'ed value of UART\_FIFINT\_\* values

Definition at line 494 of file uart\_5411x.h.

#### 6.47.3.15 `__STATIC_INLINE uint32_t Chip_UART_GetFIFOStatus ( LPC_USART_T * pUART )`

Get the current status of UART controller FIFO.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
--------------	---

## Returns

UART Status (Or-ed bit value of UART\_FIFOSTAT\_\*)

## Note

Mask the return value with a value of type UART\_FIFOSTAT\_\* to determine if that status is active.

Definition at line 425 of file uart\_5411x.h.

#### 6.47.3.16 `__STATIC_INLINE uint32_t Chip_UART_GetIntsEnabled ( LPC_USART_T * pUART )`

Returns UART interrupts that are enabled.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
--------------	---------------------------------------

## Returns

Returns the enabled UART interrupts

## Note

Use an OR'ed value of UART\_INTEN\_\* definitions with this function to determine which interrupts are enabled. You can check for multiple enabled bits if needed.

Definition at line 357 of file uart\_5411x.h.

6.47.3.17 `__STATIC_INLINE uint32_t Chip_UART_GetIntStatus ( LPC_USART_T * pUART )`

Get UART interrupt status.

**Parameters**

<i>pUART</i>	: The base of UART peripheral on the chip
--------------	---

**Returns**

The UART interrupt status register

**Note**

This register does not indicate UART pending interrupts. It indicates the enabled interrupts. To get pending interrupts, read the UART status register.

Definition at line 370 of file uart\_5411x.h.

6.47.3.18 `__STATIC_INLINE uint32_t Chip_UART_GetStatus ( LPC_USART_T * pUART )`

Get the UART status register.

**Parameters**

<i>pUART</i>	: Pointer to selected UARTx peripheral
--------------	--

**Returns**

UART status register

**Note**

Multiple statuses may be pending. Mask the return value with one or more UART\_STAT\_\* definitions to determine statuses.

Definition at line 399 of file uart\_5411x.h.

6.47.3.19 `int Chip_UART_Init ( LPC_USART_T * pUART )`

Initialize the UART peripheral.

**Parameters**

<i>pUART</i>	: The base of UART peripheral on the chip
--------------	---

**Returns**

0 - On success; [ERR\\_FLEXCOMM\\_FUNCNOTSUPPORTED](#) or [ERR\\_FLEXCOMM\\_NOTFREE](#) on failure

Definition at line 155 of file uart\_5411x.c.

6.47.3.20 `__STATIC_INLINE void Chip_UART_IntDisable ( LPC_USART_T * pUART, uint32_t intMask )`

Disable UART interrupts.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
<i>intMask</i>	: OR'ed Interrupts to disable

## Returns

Nothing

## Note

Use an OR'ed value of UART\_INTEN\_\* definitions with this function to disable specific UART interrupts.

Definition at line 344 of file uart\_5411x.h.

6.47.3.21 `__STATIC_INLINE void Chip_UART_IntEnable ( LPC_USART_T * pUART, uint32_t intMask )`

Enable UART interrupts.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
<i>intMask</i>	: OR'ed Interrupts to enable

## Returns

Nothing

## Note

Use an OR'ed value of UART\_INTEN\_\* definitions with this function to enable specific UART interrupts.

Definition at line 331 of file uart\_5411x.h.

6.47.3.22 `void Chip_UART_IRQHandlerDMA ( LPC_USART_T * pUART, UART_STATISTICS_T * statistics )`

UART receive/transmit interrupt handler for DMA.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
<i>statistics</i>	: Pointer to statistics structure

## Returns

Nothing

## Note

This interrupt service routine provides a basic implementation of the UART interrupt handler for DMA

Definition at line 387 of file uart\_5411x.c.

6.47.3.23 `void Chip_UART_IRQHandlerRB ( LPC_USART_T * pUART, UART_STATISTICS_T * statistics, RINGBUFF_T * pRXRB, RINGBUFF_T * pTXRB )`

UART receive/transmit interrupt handler for ring buffers.

**Parameters**

<i>pUART</i>	: Pointer to selected UART peripheral
<i>statistics</i>	: Pointer to statistics structure
<i>pRXRB</i>	: Pointer to transmit ring buffer
<i>pTXRB</i>	: Pointer to receive ring buffer

**Returns**

Nothing

**Note**

This provides a basic implementation of the UART IRQ handler for support of a ring buffer implementation for transmit and receive.

Definition at line 327 of file uart\_5411x.c.

6.47.3.24 `int Chip_UART_Read ( LPC_USART_T * pUART, void * data, int numBytes )`

Read data through the UART peripheral (non-blocking)

**Parameters**

<i>pUART</i>	: Pointer to selected UART peripheral
<i>data</i>	: Pointer to bytes array to fill
<i>numBytes</i>	: Size of the passed data array

**Returns**

The actual number of bytes read

**Note**

This function reads data from the receive FIFO until either all the data has been read or the passed buffer is completely full. This function will not block. This function ignores errors.

Definition at line 211 of file uart\_5411x.c.

6.47.3.25 `int Chip_UART_ReadBlocking ( LPC_USART_T * pUART, void * data, int numBytes )`

Read data through the UART peripheral (blocking)

**Parameters**

<i>pUART</i>	: Pointer to selected UART peripheral
<i>data</i>	: Pointer to data array to fill
<i>numBytes</i>	: Size of the passed data array

**Returns**

The size of the dat array

**Note**

This function reads data from the receive FIFO until the passed buffer is completely full. The function will block until full. This function ignores errors.

Definition at line 226 of file uart\_5411x.c.



6.47.3.26 `__STATIC_INLINE uint32_t Chip_UART_ReadByte ( LPC_USART_T * pUART )`

Read a single byte data from the UART peripheral.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
--------------	---------------------------------------

## Returns

A single byte of data read

## Note

This function reads a byte from the UART receive FIFO or receive hold register regardless of UART state.

Definition at line 317 of file uart\_5411x.h.

**6.47.3.27** int Chip\_UART\_ReadRB ( LPC\_USART\_T \* *pUART*, RINGBUFF\_T \* *pRB*, void \* *data*, int *bytes* )

Copy data from a receive ring buffer.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
<i>pRB</i>	: Pointer to ring buffer structure to use
<i>data</i>	: Pointer to buffer to fill from ring buffer
<i>bytes</i>	: Size of the passed buffer in bytes

## Returns

The number of bytes placed into the ring buffer

## Note

Will move the data from the RX ring buffer up to the the maximum passed buffer size. Returns 0 if there is no data in the ring buffer.

Definition at line 319 of file uart\_5411x.c.

**6.47.3.28** void Chip\_UART\_RXIntHandlerRB ( LPC\_USART\_T \* *pUART*, RINGBUFF\_T \* *pRB* )

UART receive-only interrupt handler for ring buffers.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
<i>pRB</i>	: Pointer to ring buffer structure to use

## Returns

Nothing

## Note

If ring buffer support is desired for the receive side of data transfer, the UART interrupt should call this function for a receive based interrupt status.

Definition at line 279 of file uart\_5411x.c.

**6.47.3.29** int Chip\_UART\_Send ( LPC\_USART\_T \* *pUART*, const void \* *data*, int *numBytes* )

Transmit a byte array through the UART peripheral (non-blocking)

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
<i>data</i>	: Pointer to bytes to transmit
<i>numBytes</i>	: Number of bytes to transmit

## Returns

The actual number of bytes placed into the FIFO

## Note

This function places data into the transmit FIFO until either all the data is in the FIFO or the FIFO is full. This function will not block in the FIFO is full. The actual number of bytes placed into the FIFO is returned. This function ignores errors.

Definition at line 180 of file uart\_5411x.c.

6.47.3.30 `int Chip_UART_SendBlocking ( LPC_USART_T * pUART, const void * data, int numBytes )`

Transmit a byte array through the UART peripheral (blocking)

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
<i>data</i>	: Pointer to data to transmit
<i>numBytes</i>	: Number of bytes to transmit

## Returns

The number of bytes transmitted

## Note

This function will send or place all bytes into the transmit FIFO. This function will block until the last bytes are in the FIFO.

Definition at line 195 of file uart\_5411x.c.

6.47.3.31 `__STATIC_INLINE void Chip_UART_SendByte ( LPC_USART_T * pUART, uint8_t data )`

Transmit a single data byte through the UART peripheral.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
<i>data</i>	: Byte to transmit

## Returns

Nothing

## Note

This function attempts to place a byte into the UART transmit holding register regardless of UART state.

Definition at line 305 of file uart\_5411x.h.

6.47.3.32 `uint32_t Chip_UART_SendRB ( LPC_USART_T * pUART, RINGBUFF_T * pRB, const void * data, int count )`

Populate a transmit ring buffer and start UART transmit.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
<i>pRB</i>	: Pointer to ring buffer structure to use
<i>data</i>	: Pointer to buffer to move to ring buffer
<i>count</i>	: Number of bytes to move

## Returns

The number of bytes placed into the ring buffer

## Note

Will move the data into the TX ring buffer and start the transfer. If the number of bytes returned is less than the number of bytes to send, the ring buffer is considered full.

Definition at line 300 of file uart\_5411x.c.

#### 6.47.3.33 uint32\_t Chip\_UART\_SetBaud ( LPC\_USART\_T \* pUART, uint32\_t baudrate )

Set baud rate for UART.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
<i>baudrate</i>	Baud rate to be set

## Returns

Actual baud rate set

## Note

This function uses the FRG to generate the required clock, the FRG is shared by all the UARTs, hence calling this API for multiple UARTs can affect each other.

Definition at line 252 of file uart\_5411x.c.

#### 6.47.3.34 \_\_STATIC\_INLINE void Chip\_UART\_SetFIFOCfg ( LPC\_USART\_T \* pUART, uint32\_t cfg )

Set FIFO Configuration register.

## Parameters

<i>pUART</i>	: The base of UART peripheral on the chip
<i>cfg</i>	: Configuration value mask (OR'ed UART_FIFOCFG_* values like <a href="#">UART_FIFOCFG_ENA</a> ↔ <a href="#">BLETX</a> )

## Returns

Nothing

Definition at line 505 of file uart\_5411x.h.

#### 6.47.3.35 \_\_STATIC\_INLINE void Chip\_UART\_SetFIFOTrigLevel ( LPC\_USART\_T \* pUART, uint8\_t tx\_lvl, uint8\_t rx\_lvl )

Setup the trigger level for UART FIFO.

## Parameters

<i>pUART</i>	: The base of the UART peripheral on chip
<i>tx_lvl</i>	: TX Trigger level [Valid values 0 to 15]
<i>rx_lvl</i>	: RX Trigger level [Valid values 0 to 15]

## Returns

Nothing

## Note

When *tx\_lvl* = 0; trigger will happen when TX FIFO is empty if *tx\_lvl* = 15; trigger will happen when TX FIFO has at least one free space

When *rx\_lvl* = 0; trigger will happen when RX FIFO has at least one byte in it, if *rx\_lvl* = 15; trigger will happen when RX FIFO is full and cannot receive anymore data

Definition at line 452 of file `uart_5411x.h`.

#### 6.47.3.36 `__STATIC_INLINE void Chip_UART_TXDisable ( LPC_USART_T * pUART )`

Disable transmission on UART TxD pin.

## Parameters

<i>pUART</i>	: Pointer to selected pUART peripheral
--------------	--

## Returns

Nothing

Definition at line 274 of file `uart_5411x.h`.

#### 6.47.3.37 `__STATIC_INLINE void Chip_UART_TXEnable ( LPC_USART_T * pUART )`

Enable transmission on UART TxD pin.

## Parameters

<i>pUART</i>	: Pointer to selected pUART peripheral
--------------	--

## Returns

Nothing

Definition at line 264 of file `uart_5411x.h`.

#### 6.47.3.38 `void Chip_UART_TXIntHandlerRB ( LPC_USART_T * pUART, RINGBUFF_T * pRB )`

UART transmit-only interrupt handler for ring buffers.

## Parameters

<i>pUART</i>	: Pointer to selected UART peripheral
--------------	---------------------------------------

<i>pRB</i>	: Pointer to ring buffer structure to use
------------	---

**Returns**

Nothing

**Note**

If ring buffer support is desired for the transmit side of data transfer, the UART interrupt should call this function for a transmit based interrupt status.

Definition at line 289 of file uart\_5411x.c.

## 6.48 CHIP: LPC5411X Windowed Watchdog driver

### 6.48.1 Detailed Description

#### Data Structures

- struct [LPC\\_WWDT\\_T](#)  
*Windowed Watchdog register block structure.*

#### Macros

- #define [WWDT\\_WDMOD\\_BITMASK](#) ((uint32\_t) 0x3F)  
*Watchdog Mode register definitions.*
- #define [WWDT\\_WDMOD\\_WDEN](#) ((uint32\_t) (1 << 0))
- #define [WWDT\\_WDMOD\\_WDRESET](#) ((uint32\_t) (1 << 1))
- #define [WWDT\\_WDMOD\\_WDTOF](#) ((uint32\_t) (1 << 2))
- #define [WWDT\\_WDMOD\\_WDINT](#) ((uint32\_t) (1 << 3))
- #define [WWDT\\_WDMOD\\_WDPROTECT](#) ((uint32\_t) (1 << 4))
- #define [WWDT\\_WDMOD\\_LOCK](#) ((uint32\_t) (1 << 5))

#### Functions

- `__STATIC_INLINE void Chip\_WWDT\_Init (LPC\_WWDT\_T *pWWDT)`  
*Initialize the Watchdog timer.*
- `__STATIC_INLINE void Chip\_WWDT\_DeInit (LPC\_WWDT\_T *pWWDT)`  
*Shutdown the Watchdog timer.*
- `__STATIC_INLINE void Chip\_WWDT\_SetTimeOut (LPC\_WWDT\_T *pWWDT, uint32_t timeout)`  
*Set WDT timeout constant value used for feed.*
- `__STATIC_INLINE void Chip\_WWDT\_Feed (LPC\_WWDT\_T *pWWDT)`  
*Feed watchdog timer.*
- `__STATIC_INLINE void Chip\_WWDT\_SetWarning (LPC\_WWDT\_T *pWWDT, uint32_t timeout)`  
*Set WWDT warning interrupt.*
- `__STATIC_INLINE uint32_t Chip\_WWDT\_GetWarning (LPC\_WWDT\_T *pWWDT)`  
*Get WWDT warning interrupt.*
- `__STATIC_INLINE void Chip\_WWDT\_SetWindow (LPC\_WWDT\_T *pWWDT, uint32_t timeout)`  
*Set WWDT window time.*
- `__STATIC_INLINE uint32_t Chip\_WWDT\_GetWindow (LPC\_WWDT\_T *pWWDT)`  
*Get WWDT window time.*
- `__STATIC_INLINE void Chip\_WWDT\_SetOption (LPC\_WWDT\_T *pWWDT, uint32_t options)`  
*Enable watchdog timer options.*
- `__STATIC_INLINE void Chip\_WWDT\_UnsetOption (LPC\_WWDT\_T *pWWDT, uint32_t options)`  
*Disable/clear watchdog timer options.*
- `__STATIC_INLINE void Chip\_WWDT\_Start (LPC\_WWDT\_T *pWWDT)`  
*Enable WWDT activity.*
- `__STATIC_INLINE uint32_t Chip\_WWDT\_GetStatus (LPC\_WWDT\_T *pWWDT)`  
*Read WWDT status flag.*
- `__STATIC_INLINE void Chip\_WWDT\_ClearStatusFlag (LPC\_WWDT\_T *pWWDT, uint32_t status)`  
*Clear WWDT interrupt status flags.*
- `__STATIC_INLINE uint32_t Chip\_WWDT\_GetCurrentCount (LPC\_WWDT\_T *pWWDT)`  
*Get the current value of WDT.*



## 6.48.2 Macro Definition Documentation

### 6.48.2.1 #define WWDT\_WDMOD\_BITMASK ((uint32\_t) 0x3F)

Watchdog Mode register definitions.

Watchdog Mode Bitmask

Definition at line 61 of file `wwdt_5411x.h`.

### 6.48.2.2 #define WWDT\_WDMOD\_LOCK ((uint32\_t) (1 << 5))

WWDT lock bit

Definition at line 73 of file `wwdt_5411x.h`.

### 6.48.2.3 #define WWDT\_WDMOD\_WDEN ((uint32\_t) (1 << 0))

WWDT enable bit

Definition at line 63 of file `wwdt_5411x.h`.

### 6.48.2.4 #define WWDT\_WDMOD\_WDINT ((uint32\_t) (1 << 3))

WWDT warning interrupt flag bit

Definition at line 69 of file `wwdt_5411x.h`.

### 6.48.2.5 #define WWDT\_WDMOD\_WDPROTECT ((uint32\_t) (1 << 4))

WWDT Protect flag bit

Definition at line 71 of file `wwdt_5411x.h`.

### 6.48.2.6 #define WWDT\_WDMOD\_WDRESET ((uint32\_t) (1 << 1))

WWDT reset enable bit

Definition at line 65 of file `wwdt_5411x.h`.

### 6.48.2.7 #define WWDT\_WDMOD\_WDTOF ((uint32\_t) (1 << 2))

WWDT time-out flag bit

Definition at line 67 of file `wwdt_5411x.h`.

## 6.48.3 Function Documentation

### 6.48.3.1 \_\_STATIC\_INLINE void Chip\_WWDT\_ClearStatusFlag ( LPC\_WWDT\_T \* pWWDT, uint32\_t status )

Clear WWDT interrupt status flags.

Parameters

---

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
<i>status</i>	: Or'ed value of status flag(s) that you want to clear, should be: <ul style="list-style-type: none"> <li>• WWDT_WDMOD_WDTOF: Clear watchdog timeout flag</li> <li>• WWDT_WDMOD_WDINT: Clear watchdog warning flag</li> </ul>

**Returns**

None

Definition at line 234 of file `wwdt_5411x.h`.**6.48.3.2** `__STATIC_INLINE void Chip_WWDT_DeInit ( LPC_WWDT_T * pWWDT )`

Shutdown the Watchdog timer.

**Parameters**

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

**Returns**

None

Definition at line 97 of file `wwdt_5411x.h`.**6.48.3.3** `__STATIC_INLINE void Chip_WWDT_Feed ( LPC_WWDT_T * pWWDT )`

Feed watchdog timer.

**Parameters**

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

**Returns**

None

**Note**

If this function isn't called, a watchdog timer warning will occur. After the warning, a timeout will occur if a feed has happened. Note that if WWDT registers are modified in an interrupt then it is a good idea to prevent those interrupts when writing the feed sequence.

Definition at line 122 of file `wwdt_5411x.h`.**6.48.3.4** `__STATIC_INLINE uint32_t Chip_WWDT_GetCurrentCount ( LPC_WWDT_T * pWWDT )`

Get the current value of WDT.

**Parameters**

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

**Returns**

current value of WDT

Definition at line 250 of file `wwdt_5411x.h`.

#### 6.48.3.5 `__STATIC_INLINE uint32_t Chip_WWDT_GetStatus ( LPC_WWDT_T * pWWDT )`

Read WWDT status flag.

**Parameters**

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

**Returns**

Watchdog status, an Or'ed value of `WWDT_WDMOD_*`

Definition at line 221 of file `wwdt_5411x.h`.

#### 6.48.3.6 `__STATIC_INLINE uint32_t Chip_WWDT_GetWarning ( LPC_WWDT_T * pWWDT )`

Get WWDT warning interrupt.

**Parameters**

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

**Returns**

WWDT warning interrupt

Definition at line 146 of file `wwdt_5411x.h`.

#### 6.48.3.7 `__STATIC_INLINE uint32_t Chip_WWDT_GetWindow ( LPC_WWDT_T * pWWDT )`

Get WWDT window time.

**Parameters**

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

**Returns**

WWDT window time

Definition at line 170 of file `wwdt_5411x.h`.

#### 6.48.3.8 `__STATIC_INLINE void Chip_WWDT_Init ( LPC_WWDT_T * pWWDT )`

Initialize the Watchdog timer.

## Parameters

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

## Returns

None

Definition at line 80 of file `wwdt_5411x.h`.6.48.3.9 `__STATIC_INLINE void Chip_WWDT_SetOption ( LPC_WWDT_T * pWWDT, uint32_t options )`

Enable watchdog timer options.

## Parameters

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
<i>options</i>	: An or'ed set of options of values <code>WWDT_WDMOD_WDEN</code> , <code>WWDT_WDMOD_WDRESET</code> , and <code>WWDT_WDMOD_WDPROTECT</code>

## Returns

None

## Note

You can enable more than one option at once (ie, `WWDT_WDMOD_WDRESET | WWDT_WDMOD_WDPROTECT`), but use the `WWDT_WDMOD_WDEN` after all other options are set (or unset) with no other options. If `WWDT_WDMOD_LOCK` is used, it cannot be unset.

Definition at line 186 of file `wwdt_5411x.h`.6.48.3.10 `__STATIC_INLINE void Chip_WWDT_SetTimeOut ( LPC_WWDT_T * pWWDT, uint32_t timeout )`

Set WDT timeout constant value used for feed.

## Parameters

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
<i>timeout</i>	: WDT timeout in ticks, between <code>WWDT_TICKS_MIN</code> and <code>WWDT_TICKS_MAX</code>

## Returns

none

Definition at line 108 of file `wwdt_5411x.h`.6.48.3.11 `__STATIC_INLINE void Chip_WWDT_SetWarning ( LPC_WWDT_T * pWWDT, uint32_t timeout )`

Set WWDT warning interrupt.

## Parameters

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

<i>timeout</i>	: WDT warning in ticks, between 0 and 1023
----------------	--

**Returns**

None

**Note**

This is the number of ticks after the watchdog interrupt that the warning interrupt will be generated.

Definition at line 136 of file `wwdt_5411x.h`.

6.48.3.12 `__STATIC_INLINE void Chip_WWDT_SetWindow ( LPC_WWDT_T * pWWDT, uint32_t timeout )`

Set WWDT window time.

**Parameters**

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
<i>timeout</i>	: WDT timeout in ticks, between WWDT_TICKS_MIN and WWDT_TICKS_MAX

**Returns**

None

**Note**

The watchdog timer must be fed between the timeout from the [Chip\\_WWDT\\_SetTimeOut\(\)](#) function and this function, with this function defining the last tick before the watchdog window interrupt occurs.

Definition at line 160 of file `wwdt_5411x.h`.

6.48.3.13 `__STATIC_INLINE void Chip_WWDT_Start ( LPC_WWDT_T * pWWDT )`

Enable WWDT activity.

**Parameters**

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

**Returns**

None

Definition at line 210 of file `wwdt_5411x.h`.

6.48.3.14 `__STATIC_INLINE void Chip_WWDT_UnsetOption ( LPC_WWDT_T * pWWDT, uint32_t options )`

Disable/clear watchdog timer options.

**Parameters**

<i>pWWDT</i>	: The base of WatchDog Timer peripheral on the chip
--------------	---

<i>options</i>	: An or'ed set of options of values WWDT_WDMOD_WDEN, WWDT_WDMOD_WDRESET, and WWDT_WDMOD_WDPROTECT
----------------	---

**Returns**

None

**Note**

You can disable more than one option at once (ie, WWDT\_WDMOD\_WDRESET | WWDT\_WDMOD\_WD←OF).

Definition at line 200 of file wwdt\_5411x.h.

## 6.49 CHIP: LPC5411X flexcomm API

### 6.49.1 Detailed Description

#### Macros

- `#define FLEXCOMM_LOCK (1 << 3)`  
*FlexCOMM PSEL register bits.*
- `#define FLEXCOMM_ID_USART (1 << 4)`
- `#define FLEXCOMM_ID_SPI (1 << 5)`
- `#define FLEXCOMM_ID_I2C (1 << 6)`
- `#define FLEXCOMM_ID_I2S (1 << 7)`
- `#define ERR_FLEXCOMM_FUNCNOTSUPPORTED -1`  
*Flexcomm error and offset values.*
- `#define ERR_FLEXCOMM_NOTFREE -2`
- `#define ERR_FLEXCOMM_INVALIDBASE -3`
- `#define FLEXCOMM_PSEL_OFFSET 0xFF8`

#### Typedefs

- `typedef void LPC_FLEXCOMM_T`

#### Enumerations

- `enum FLEXCOMM_PERIPH_T {`  
`FLEXCOMM_PERIPH_NONE, FLEXCOMM_PERIPH_USART, FLEXCOMM_PERIPH_SPI, FLEXCOMM_↵`  
`_PERIPH_I2C,`  
`FLEXCOMM_PERIPH_I2S_TX, FLEXCOMM_PERIPH_I2S_RX }`  
*FLEXCOMM Peripheral functions.*

#### Functions

- `__STATIC_INLINE FLEXCOMM_PERIPH_T Chip_FLEXCOMM_GetFunc (LPC_FLEXCOMM_T *pFCO↵`  
`MM)`  
*Get currently enabled FLEXCOMM function.*
- `__STATIC_INLINE int Chip_FLEXCOMM_IsLocked (LPC_FLEXCOMM_T *pFCOMM)`  
*Checks if given FLEXCOMM is locked to a function.*
- `__STATIC_INLINE void Chip_FLEXCOMM_Lock (LPC_FLEXCOMM_T *pFCOMM)`  
*Lock FLEXCOMM to a function.*
- `int Chip_FLEXCOMM_SetPeriph (LPC_FLEXCOMM_T *pFCOMM, FLEXCOMM_PERIPH_T periph, int`  
`lock)`  
*Set FLEXCOMM to a peripheral function.*
- `int Chip_FLEXCOMM_GetIndex (LPC_FLEXCOMM_T *pFCOMM)`  
*Get index of the FLEXCOMM corresponding to the given base address.*
- `int Chip_FLEXCOMM_Init (LPC_FLEXCOMM_T *pFCOMM, FLEXCOMM_PERIPH_T periph)`  
*Initialize FlexCOMM and associate it with a given peripheral.*
- `void Chip_FLEXCOMM_DeInit (LPC_FLEXCOMM_T *pFCOMM)`  
*Uninitialize the FlexCOMM.*

## 6.49.2 Macro Definition Documentation

### 6.49.2.1 `#define ERR_FLEXCOMM_FUNCNOTSUPPORTED -1`

Flexcomm error and offset values.

Selected flexcomm does not support the function desired

Definition at line 65 of file flexcomm\_5411x.h.

### 6.49.2.2 `#define ERR_FLEXCOMM_INVALIDBASE -3`

Invalid flexcomm base address

Definition at line 67 of file flexcomm\_5411x.h.

### 6.49.2.3 `#define ERR_FLEXCOMM_NOTFREE -2`

FlexCOMM is not free (used as a different peripheral)

Definition at line 66 of file flexcomm\_5411x.h.

### 6.49.2.4 `#define FLEXCOMM_ID_I2C (1 << 6)`

I2C Support bit

Definition at line 61 of file flexcomm\_5411x.h.

### 6.49.2.5 `#define FLEXCOMM_ID_I2S (1 << 7)`

I2S Support bit

Definition at line 62 of file flexcomm\_5411x.h.

### 6.49.2.6 `#define FLEXCOMM_ID_SPI (1 << 5)`

SPI Support bit

Definition at line 60 of file flexcomm\_5411x.h.

### 6.49.2.7 `#define FLEXCOMM_ID_USART (1 << 4)`

USART Support bit

Definition at line 59 of file flexcomm\_5411x.h.

### 6.49.2.8 `#define FLEXCOMM_LOCK (1 << 3)`

FlexCOMM PSEL register bits.

Lock Bit

Definition at line 58 of file flexcomm\_5411x.h.

### 6.49.2.9 `#define FLEXCOMM_PSEL_OFFSET 0xFF8`

Offset of the flexcomm PSEL register



Definition at line 68 of file flexcomm\_5411x.h.

### 6.49.3 Typedef Documentation

#### 6.49.3.1 typedef void LPC\_FLEXCOMM\_T

Definition at line 45 of file flexcomm\_5411x.h.

### 6.49.4 Enumeration Type Documentation

#### 6.49.4.1 enum FLEXCOMM\_PERIPH\_T

FLEXCOMM Peripheral functions.

Enumerator

**FLEXCOMM\_PERIPH\_NONE** No peripheral  
**FLEXCOMM\_PERIPH\_USART** USART peripheral  
**FLEXCOMM\_PERIPH\_SPI** SPI Peripheral  
**FLEXCOMM\_PERIPH\_I2C** I2C Peripheral  
**FLEXCOMM\_PERIPH\_I2S\_TX** I2S TX Peripheral  
**FLEXCOMM\_PERIPH\_I2S\_RX** I2S RX Peripheral

Definition at line 48 of file flexcomm\_5411x.h.

### 6.49.5 Function Documentation

#### 6.49.5.1 void Chip\_FLEXCOMM\_DeInit ( LPC\_FLEXCOMM\_T \* *pFCOMM* )

Uninitialize the FlexCOMM.

Parameters

<i>pFCOMM</i>	: Base address of the flexcomm peripheral
---------------	---

Returns

Nothing

Definition at line 106 of file flexcomm\_5411x.c.

#### 6.49.5.2 \_\_STATIC\_INLINE FLEXCOMM\_PERIPH\_T Chip\_FLEXCOMM\_GetFunc ( LPC\_FLEXCOMM\_T \* *pFCOMM* )

Get currently enabled FLEXCOMM function.

Parameters

<i>pFCOMM</i>	: Base address of the flexcomm peripheral
---------------	---

Returns

Enabled flexcomm peripheral (See [FLEXCOMM\\_PERIPH\\_T](#))

Definition at line 75 of file flexcomm\_5411x.h.

6.49.5.3 int Chip\_FLEXCOMM\_GetIndex ( LPC\_FLEXCOMM\_T \* *pFCOMM* )

Get index of the FLEXCOMM corresponding to the given base address.

## Parameters

<i>pFCOMM</i>	: Base address of the flexcomm peripheral
---------------	---

## Returns

Index of FLEXCOMM, [ERR\\_FLEXCOMM\\_INVALIDBASE](#) - when base address is invalid

Definition at line 51 of file flexcomm\_5411x.c.

#### 6.49.5.4 int Chip\_FLEXCOMM\_Init ( LPC\_FLEXCOMM\_T \* *pFCOMM*, FLEXCOMM\_PERIPH\_T *periph* )

Initialize FlexCOMM and associate it with a given peripheral.

## Parameters

<i>pFCOMM</i>	: Base address of the flexcomm peripheral
<i>periph</i>	: Peripheral to set the FlexCOMM to

## Returns

0 on success, [ERR\\_FLEXCOMM\\_FUNCNOTSUPPORTED](#) when the given FlexComm does not support peripheral provided by *periph*, [ERR\\_FLEXCOMM\\_NOTFREE](#) when the flexcomm is being used already as a peripheral.

Definition at line 76 of file flexcomm\_5411x.c.

#### 6.49.5.5 \_\_STATIC\_INLINE int Chip\_FLEXCOMM\_IsLocked ( LPC\_FLEXCOMM\_T \* *pFCOMM* )

Checks if given FLEXCOMM is locked to a function.

## Parameters

<i>pFCOMM</i>	: Base address of the flexcomm peripheral
---------------	---

## Returns

1 - FLEXCOMM is locked to a function; 0 - FLEXCOMM is not locked

Definition at line 85 of file flexcomm\_5411x.h.

#### 6.49.5.6 \_\_STATIC\_INLINE void Chip\_FLEXCOMM\_Lock ( LPC\_FLEXCOMM\_T \* *pFCOMM* )

Lock FLEXCOMM to a function.

## Parameters

<i>pFCOMM</i>	: Base address of the flexcomm peripheral
---------------	---

## Returns

Nothing

## Note

Once the FLEXCOMM is locked, it can only be unlocked by a reset

Definition at line 96 of file flexcomm\_5411x.h.

6.49.5.7 `int Chip_FLEXCOMM_SetPeriph ( LPC_FLEXCOMM_T * pFCOMM, FLEXCOMM_PERIPH_T periph, int lock )`

Set FLEXCOMM to a peripheral function.

## Parameters

<i>pFCOMM</i>	: Base address of the flexcomm peripheral
<i>periph</i>	: Selected peripheral (See <a href="#">FLEXCOMM_PERIPH_T</a> )
<i>lock</i>	: 1 - Flexcomm will be locked as given peripheral, 0 - Do not lock

## See also

[Chip\\_FLEXCOMM\\_Init\(\)](#)

## Returns

0 on success, [ERR\\_FLEXCOMM\\_FUNCNOTSUPPORTED](#) when the given FlexComm does not support peripheral provided by *periph*, [ERR\\_FLEXCOMM\\_NOTFREE](#) when the flexcomm is being used already as a peripheral.

## Note

Once the FLEXCOMM is locked, it can only be unlocked by a reset

Definition at line 128 of file flexcomm\_5411x.c.

## 6.50 CHIP: LPC5411X support functions

### 6.50.1 Detailed Description

#### Functions

- void [SystemCoreClockUpdate](#) (void)  
*Update system core and ASYNC syscon clock rate, should be called if the system has a clock rate change.*
- void [Chip\\_SystemInit](#) (void)  
*Set up and initialize hardware prior to call to main()*
- void [Chip\\_SetupIrcClocking](#) (uint32\_t iFreq)  
*Clock and PLL initialization based on the internal oscillator.*
- void [Chip\\_SetupExtInClocking](#) (uint32\_t iFreq)  
*Clock and PLL initialization based on the external clock input.*
- void [Chip\\_SetupFROClocking](#) (uint32\_t iFreq)  
*Initialize the Core clock to given frequency (12, 48 or 96 MHz)*
- void [Chip\\_USB\\_Init](#) (void)  
*Initialize the USB bus.*
- `__STATIC_INLINE` void [Chip\\_USB\\_TrimOff](#) (int enable)  
*Turn of FRO clock trimming based on USB SOF.*

#### Variables

- uint32\_t [SystemCoreClock](#)  
*Current system clock rate, mainly used for peripherals in SYSCON.*

### 6.50.2 Function Documentation

#### 6.50.2.1 void [Chip\\_SetupExtInClocking](#) ( uint32\_t iFreq )

Clock and PLL initialization based on the external clock input.

##### Parameters

<i>iFreq</i>	: Rate (in Hz) to set the main system clock to
--------------	--

##### Returns

None

Definition at line 141 of file sysinit\_5411x.c.

#### 6.50.2.2 void [Chip\\_SetupFROClocking](#) ( uint32\_t iFreq )

Initialize the Core clock to given frequency (12, 48 or 96 MHz)

##### Parameters

<i>iFreq</i>	: Desired frequency (must be one of <a href="#">SYSCON_FRO12MHZ_FREQ</a> or <a href="#">SYSCON_FRO48MHZ_FREQ</a> or <a href="#">SYSCON_FRO96MHZ_FREQ</a> )
--------------	--

##### Returns

Nothing

Definition at line 70 of file sysinit\_5411x.c.

### 6.50.2.3 void Chip\_SetupIrcClocking ( uint32\_t *iFreq* )

Clock and PLL initialization based on the internal oscillator.

#### Parameters

<i>iFreq</i>	: Rate (in Hz) to set the main system clock to
--------------	--

#### Returns

None

Definition at line 97 of file sysinit\_5411x.c.

### 6.50.2.4 void Chip\_SystemInit ( void )

Set up and initialize hardware prior to call to main()

#### Returns

None

#### Note

[Chip\\_SystemInit\(\)](#) is called prior to the application and sets up system clocking prior to the application starting.

Definition at line 191 of file sysinit\_5411x.c.

### 6.50.2.5 void Chip\_USB\_Init ( void )

Initialize the USB bus.

#### Returns

Nothing

#### Note

Uses FRO HF to initialize the pin-IO and the clocks.

Definition at line 61 of file chip\_5411x.c.

### 6.50.2.6 \_\_STATIC\_INLINE void Chip\_USB\_TrimOff ( int *enable* )

Turn of FRO clock trimming based on USB SOF.

#### Parameters

<i>enable</i>	: 0 - Disable trim based on USB SOF; Non-Zero to enable it
---------------	--

#### Returns

Nothing

Definition at line 271 of file chip.h.

#### 6.50.2.7 void SystemCoreClockUpdate ( void )

Update system core and ASYNC syscon clock rate, should be called if the system has a clock rate change.

##### Returns

None

Definition at line 55 of file chip\_5411x.c.

### 6.50.3 Variable Documentation

#### 6.50.3.1 uint32\_t SystemCoreClock

Current system clock rate, mainly used for peripherals in SYSCON.

Definition at line 43 of file chip\_5411x.c.



## 6.51 CHIP: LPC5411x Chip driver build time options

### 6.51.1 Detailed Description

Some chip drivers require build-time configuration to enable and disable specific platform features. A build-time option is configured by the use of a definition passed to the compiler during the build process or by adding the definition to the `sys_config.h` file.

#### **CORE\_M4 definition**

LPC5411x devices should add a unvalued `CORE_M4` definition to the compilers argument list when building images for the M4 core. Do not add this definition as part of `sys_config.h`.

#### **CORE\_M0PLUS definition**

LPC5411x devices should add a unvalued `CORE_M0PLUS` definition to the compilers argument list when building images for the M0+ core. Do not add this definition as part of `sys_config.h`.

#### **CHIP\_LPC5411X definition**

`CHIP_LPC5411X` must be defined for all code when building with Keil and IAR for the LPC5411x device. This unvalued definition can be added to the compilers argument list or as part of `sys_config.h`.

### 6.51.2 Variable Documentation

#### 6.51.2.1 `const uint32_t ExtClockIn`

Clock rate on the CLKIN pin This value is defined externally to the chip layer and contains the value in Hz for the CLKIN pin for the board. If this pin isn't used, this rate can be 0.

## 6.52 CHIP: LPC5411x I2C driver

### 6.52.1 Detailed Description

#### Modules

- [CHIP: LPC5411X I2C master-only driver](#)
- [CHIP: LPC5411X I2C slave-only driver](#)

#### Macros

- `#define I2C_CFG_MSTEN (1 << 0)`
- `#define I2C_CFG_MSTEN (1 << 0)`
- `#define I2C_CFG_SLVEN (1 << 1)`
- `#define I2C_CFG_SLVEN (1 << 1)`
- `#define I2C_CFG_MONEN (1 << 2)`
- `#define I2C_CFG_MONEN (1 << 2)`
- `#define I2C_CFG_TIMEOUTEN (1 << 3)`
- `#define I2C_CFG_TIMEOUTEN (1 << 3)`
- `#define I2C_CFG_MONCLKSTR (1 << 4)`
- `#define I2C_CFG_MONCLKSTR (1 << 4)`
- `#define I2C_CFG_MASK ((uint32_t) 0x1F)`
- `#define I2C_CFG_MASK ((uint32_t) 0x1F)`
- `#define I2C_STAT_MSTPENDING (1 << 0)`
- `#define I2C_STAT_MSTPENDING (1 << 0)`
- `#define I2C_STAT_MSTSTATE (0x7 << 1)`
- `#define I2C_STAT_MSTSTATE (0x7 << 1)`
- `#define I2C_STAT_MSTRARBLOSS (1 << 4)`
- `#define I2C_STAT_MSTRARBLOSS (1 << 4)`
- `#define I2C_STAT_MSTSTSTPERR (1 << 6)`
- `#define I2C_STAT_MSTSTSTPERR (1 << 6)`
- `#define I2C_STAT_SLVPENDING (1 << 8)`
- `#define I2C_STAT_SLVPENDING (1 << 8)`
- `#define I2C_STAT_SLVSTATE (0x3 << 9)`
- `#define I2C_STAT_SLVSTATE (0x3 << 9)`
- `#define I2C_STAT_SLVNOTSTR (1 << 11)`
- `#define I2C_STAT_SLVNOTSTR (1 << 11)`
- `#define I2C_STAT_SLVIDX (0x3 << 12)`
- `#define I2C_STAT_SLVIDX (0x3 << 12)`
- `#define I2C_STAT_SLVSEL (1 << 14)`
- `#define I2C_STAT_SLVSEL (1 << 14)`
- `#define I2C_STAT_SLVDESEL (1 << 15)`
- `#define I2C_STAT_SLVDESEL (1 << 15)`
- `#define I2C_STAT_MONRDY (1 << 16)`
- `#define I2C_STAT_MONRDY (1 << 16)`
- `#define I2C_STAT_MONOV (1 << 17)`
- `#define I2C_STAT_MONOV (1 << 17)`
- `#define I2C_STAT_MONACTIVE (1 << 18)`
- `#define I2C_STAT_MONACTIVE (1 << 18)`
- `#define I2C_STAT_MONIDLE (1 << 19)`
- `#define I2C_STAT_MONIDLE (1 << 19)`
- `#define I2C_STAT_EVENTTIMEOUT (1 << 24)`
- `#define I2C_STAT_EVENTTIMEOUT (1 << 24)`
- `#define I2C_STAT_SCLTIMEOUT (1 << 25)`

- #define I2C\_STAT\_SCLTIMEOUT (1 << 25)
- #define I2C\_STAT\_MSTCODE\_IDLE (0)
- #define I2C\_STAT\_MSTCODE\_IDLE (0)
- #define I2C\_STAT\_MSTCODE\_RXREADY (1)
- #define I2C\_STAT\_MSTCODE\_RXREADY (1)
- #define I2C\_STAT\_MSTCODE\_TXREADY (2)
- #define I2C\_STAT\_MSTCODE\_TXREADY (2)
- #define I2C\_STAT\_MSTCODE\_NACKADR (3)
- #define I2C\_STAT\_MSTCODE\_NACKADR (3)
- #define I2C\_STAT\_MSTCODE\_NACKDAT (4)
- #define I2C\_STAT\_MSTCODE\_NACKDAT (4)
- #define I2C\_STAT\_SLVCODE\_ADDR (0)
- #define I2C\_STAT\_SLVCODE\_ADDR (0)
- #define I2C\_STAT\_SLVCODE\_RX (1)
- #define I2C\_STAT\_SLVCODE\_RX (1)
- #define I2C\_STAT\_SLVCODE\_TX (2)
- #define I2C\_STAT\_SLVCODE\_TX (2)
- #define I2C\_INTENSET\_MSTPENDING (1 << 0)
- #define I2C\_INTENSET\_MSTPENDING (1 << 0)
- #define I2C\_INTENSET\_MSTRARBLOSS (1 << 4)
- #define I2C\_INTENSET\_MSTRARBLOSS (1 << 4)
- #define I2C\_INTENSET\_MSTSTSPERR (1 << 6)
- #define I2C\_INTENSET\_MSTSTSPERR (1 << 6)
- #define I2C\_INTENSET\_SLVPENDING (1 << 8)
- #define I2C\_INTENSET\_SLVPENDING (1 << 8)
- #define I2C\_INTENSET\_SLVNOTSTR (1 << 11)
- #define I2C\_INTENSET\_SLVNOTSTR (1 << 11)
- #define I2C\_INTENSET\_SLVDESEL (1 << 15)
- #define I2C\_INTENSET\_SLVDESEL (1 << 15)
- #define I2C\_INTENSET\_MONRDY (1 << 16)
- #define I2C\_INTENSET\_MONRDY (1 << 16)
- #define I2C\_INTENSET\_MONOV (1 << 17)
- #define I2C\_INTENSET\_MONOV (1 << 17)
- #define I2C\_INTENSET\_MONIDLE (1 << 19)
- #define I2C\_INTENSET\_MONIDLE (1 << 19)
- #define I2C\_INTENSET\_EVENTTIMEOUT (1 << 24)
- #define I2C\_INTENSET\_EVENTTIMEOUT (1 << 24)
- #define I2C\_INTENSET\_SCLTIMEOUT (1 << 25)
- #define I2C\_INTENSET\_SCLTIMEOUT (1 << 25)
- #define I2C\_INTENCLR\_MSTPENDING (1 << 0)
- #define I2C\_INTENCLR\_MSTPENDING (1 << 0)
- #define I2C\_INTENCLR\_MSTRARBLOSS (1 << 4)
- #define I2C\_INTENCLR\_MSTRARBLOSS (1 << 4)
- #define I2C\_INTENCLR\_MSTSTSPERR (1 << 6)
- #define I2C\_INTENCLR\_MSTSTSPERR (1 << 6)
- #define I2C\_INTENCLR\_SLVPENDING (1 << 8)
- #define I2C\_INTENCLR\_SLVPENDING (1 << 8)
- #define I2C\_INTENCLR\_SLVNOTSTR (1 << 11)
- #define I2C\_INTENCLR\_SLVNOTSTR (1 << 11)
- #define I2C\_INTENCLR\_SLVDESEL (1 << 15)
- #define I2C\_INTENCLR\_SLVDESEL (1 << 15)
- #define I2C\_INTENCLR\_MONRDY (1 << 16)
- #define I2C\_INTENCLR\_MONRDY (1 << 16)
- #define I2C\_INTENCLR\_MONOV (1 << 17)
- #define I2C\_INTENCLR\_MONOV (1 << 17)

- #define I2C\_INTENCLR\_MONIDLE (1 << 19)
- #define I2C\_INTENCLR\_MONIDLE (1 << 19)
- #define I2C\_INTENCLR\_EVENTTIMEOUT (1 << 24)
- #define I2C\_INTENCLR\_EVENTTIMEOUT (1 << 24)
- #define I2C\_INTENCLR\_SCLTIMEOUT (1 << 25)
- #define I2C\_INTENCLR\_SCLTIMEOUT (1 << 25)
- #define I2C\_TIMEOUT\_VAL(n) (((uint32\_t) ((n) - 1) & 0xFFF0) | 0x000F)
- #define I2C\_TIMEOUT\_VAL(n) (((uint32\_t) ((n) - 1) & 0xFFF0) | 0x000F)
- #define I2C\_INTSTAT\_MSTPENDING (1 << 0)
- #define I2C\_INTSTAT\_MSTPENDING (1 << 0)
- #define I2C\_INTSTAT\_MSTRARBLOSS (1 << 4)
- #define I2C\_INTSTAT\_MSTRARBLOSS (1 << 4)
- #define I2C\_INTSTAT\_MSTSTSTPERR (1 << 6)
- #define I2C\_INTSTAT\_MSTSTSTPERR (1 << 6)
- #define I2C\_INTSTAT\_SLVPENDING (1 << 8)
- #define I2C\_INTSTAT\_SLVPENDING (1 << 8)
- #define I2C\_INTSTAT\_SLVNOTSTR (1 << 11)
- #define I2C\_INTSTAT\_SLVNOTSTR (1 << 11)
- #define I2C\_INTSTAT\_SLVDESEL (1 << 15)
- #define I2C\_INTSTAT\_SLVDESEL (1 << 15)
- #define I2C\_INTSTAT\_MONRDY (1 << 16)
- #define I2C\_INTSTAT\_MONRDY (1 << 16)
- #define I2C\_INTSTAT\_MONOV (1 << 17)
- #define I2C\_INTSTAT\_MONOV (1 << 17)
- #define I2C\_INTSTAT\_MONIDLE (1 << 19)
- #define I2C\_INTSTAT\_MONIDLE (1 << 19)
- #define I2C\_INTSTAT\_EVENTTIMEOUT (1 << 24)
- #define I2C\_INTSTAT\_EVENTTIMEOUT (1 << 24)
- #define I2C\_INTSTAT\_SCLTIMEOUT (1 << 25)
- #define I2C\_INTSTAT\_SCLTIMEOUT (1 << 25)
- #define I2C\_MSTCTL\_MSTCONTINUE (1 << 0)
- #define I2C\_MSTCTL\_MSTCONTINUE (1 << 0)
- #define I2C\_MSTCTL\_MSTSTART (1 << 1)
- #define I2C\_MSTCTL\_MSTSTART (1 << 1)
- #define I2C\_MSTCTL\_MSTSTOP (1 << 2)
- #define I2C\_MSTCTL\_MSTSTOP (1 << 2)
- #define I2C\_MSTCTL\_MSTDMA (1 << 3)
- #define I2C\_MSTCTL\_MSTDMA (1 << 3)
- #define I2C\_MSTTIME\_MSTSCLOW (0x07 << 0)
- #define I2C\_MSTTIME\_MSTSCLOW (0x07 << 0)
- #define I2C\_MSTTIME\_MSTSCHIGH (0x07 << 4)
- #define I2C\_MSTTIME\_MSTSCHIGH (0x07 << 4)
- #define I2C\_MSTDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)
- #define I2C\_MSTDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)
- #define I2C\_SLVCTL\_SLVCONTINUE (1 << 0)
- #define I2C\_SLVCTL\_SLVCONTINUE (1 << 0)
- #define I2C\_SLVCTL\_SLVNACK (1 << 1)
- #define I2C\_SLVCTL\_SLVNACK (1 << 1)
- #define I2C\_SLVCTL\_SLVDMA (1 << 3)
- #define I2C\_SLVCTL\_SLVDMA (1 << 3)
- #define I2C\_SLVDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)
- #define I2C\_SLVDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)
- #define I2C\_SLVADR\_SADISABLE (1 << 0)
- #define I2C\_SLVADR\_SADISABLE (1 << 0)
- #define I2C\_SLVADR\_SLVADR (0x7F << 1)

- `#define I2C_SLVADR_SLVADR (0x7F << 1)`
- `#define I2C_SLVADR_MASK ((uint32_t) 0x00FF)`
- `#define I2C_SLVADR_MASK ((uint32_t) 0x00FF)`
- `#define I2C_SLVQUAL_QUALMODE0 (1 << 0)`
- `#define I2C_SLVQUAL_QUALMODE0 (1 << 0)`
- `#define I2C_SLVQUAL_SLVQUAL0 (0x7F << 1)`
- `#define I2C_SLVQUAL_SLVQUAL0 (0x7F << 1)`
- `#define I2C_MONRXDAT_DATA (0xFF << 0)`
- `#define I2C_MONRXDAT_DATA (0xFF << 0)`
- `#define I2C_MONRXDAT_MONSTART (1 << 8)`
- `#define I2C_MONRXDAT_MONSTART (1 << 8)`
- `#define I2C_MONRXDAT_MONRESTART (1 << 9)`
- `#define I2C_MONRXDAT_MONRESTART (1 << 9)`
- `#define I2C_MONRXDAT_MONNACK (1 << 10)`
- `#define I2C_MONRXDAT_MONNACK (1 << 10)`

## Functions

- `__STATIC_INLINE int Chip_I2C_Init (LPC_I2C_T *pI2C)`  
*Initialize I2C Interface.*
- `__STATIC_INLINE void Chip_I2C_DeInit (LPC_I2C_T *pI2C)`  
*Shutdown I2C Interface.*
- `__STATIC_INLINE void Chip_I2C_SetClockDiv (LPC_I2C_T *pI2C, uint32_t clkdiv)`  
*Sets I2C Clock Divider registers.*
- `static INLINE uint32_t Chip_I2C_GetClockDiv (LPC_I2C_T *pI2C)`  
*Get I2C Clock Divider registers.*
- `static INLINE void Chip_I2C_EnableInt (LPC_I2C_T *pI2C, uint32_t intEn)`  
*Enable I2C Interrupts.*
- `static INLINE void Chip_I2C_DisableInt (LPC_I2C_T *pI2C, uint32_t intClr)`  
*Disable I2C Interrupts.*
- `static INLINE void Chip_I2C_ClearInt (LPC_I2C_T *pI2C, uint32_t intClr)`  
*Disable I2C Interrupts.*
- `static INLINE uint32_t Chip_I2C_GetPendingInt (LPC_I2C_T *pI2C)`  
*Returns pending I2C Interrupts.*

## Variables

- `__IO uint32_t STAT`  
*I2C register block structure.*
- `__IO uint32_t INTENSET`
- `__O uint32_t INTENCLR`
- `__IO uint32_t TIMEOUT`
- `__IO uint32_t CLKDIV`
- `__IO uint32_t INTSTAT`
- `__I uint32_t RESERVED0`
- `__IO uint32_t MSTCTL`
- `__IO uint32_t MSTTIME`
- `__IO uint32_t MSTDAT`
- `__IO uint32_t RESERVED1 [5]`
- `__IO uint32_t SLVCTL`
- `__IO uint32_t SLVDAT`
- `__IO uint32_t SLVADR [4]`

- [\\_\\_IO uint32\\_t SLVQUAL0](#)
- [\\_\\_IO uint32\\_t RESERVED2 \[9\]](#)
- [\\_\\_I uint32\\_t MONRXDAT](#)
- [\\_\\_IO uint32\\_t PSELID](#)
- [\\_\\_I uint32\\_t PID](#)
- [LPC\\_I2C\\_T](#)

## 6.52.2 Macro Definition Documentation

### 6.52.2.1 `#define I2C_CFG_MASK ((uint32_t) 0x1F)`

Configuration Register Mask

Definition at line 225 of file `i2c_common_5411x.h`.

### 6.52.2.2 `#define I2C_CFG_MASK ((uint32_t) 0x1F)`

Configuration Register Mask

Definition at line 225 of file `i2c_common_5411x.h`.

### 6.52.2.3 `#define I2C_CFG_MONCLKSTR (1 << 4)`

Monitor Clock Stretching Bit

Definition at line 224 of file `i2c_common_5411x.h`.

### 6.52.2.4 `#define I2C_CFG_MONCLKSTR (1 << 4)`

Monitor Clock Stretching Bit

Definition at line 224 of file `i2c_common_5411x.h`.

### 6.52.2.5 `#define I2C_CFG_MONEN (1 << 2)`

Monitor Enable/Disable Bit

Definition at line 222 of file `i2c_common_5411x.h`.

### 6.52.2.6 `#define I2C_CFG_MONEN (1 << 2)`

Monitor Enable/Disable Bit

Definition at line 222 of file `i2c_common_5411x.h`.

### 6.52.2.7 `#define I2C_CFG_MSTEN (1 << 0)`

Master Enable/Disable Bit

Definition at line 220 of file `i2c_common_5411x.h`.

### 6.52.2.8 `#define I2C_CFG_MSTEN (1 << 0)`

Master Enable/Disable Bit

Definition at line 220 of file `i2c_common_5411x.h`.

**6.52.2.9 #define I2C\_CFG\_SLVEN (1 << 1)**

Slave Enable/Disable Bit

Definition at line 221 of file i2c\_common\_5411x.h.

**6.52.2.10 #define I2C\_CFG\_SLVEN (1 << 1)**

Slave Enable/Disable Bit

Definition at line 221 of file i2c\_common\_5411x.h.

**6.52.2.11 #define I2C\_CFG\_TIMEOUTEN (1 << 3)**

Timeout Enable/Disable Bit

Definition at line 223 of file i2c\_common\_5411x.h.

**6.52.2.12 #define I2C\_CFG\_TIMEOUTEN (1 << 3)**

Timeout Enable/Disable Bit

Definition at line 223 of file i2c\_common\_5411x.h.

**6.52.2.13 #define I2C\_INTENCLR\_EVENTTIMEOUT (1 << 24)**

Event Timeout Interrupt Clear Bit

Definition at line 284 of file i2c\_common\_5411x.h.

**6.52.2.14 #define I2C\_INTENCLR\_EVENTTIMEOUT (1 << 24)**

Event Timeout Interrupt Clear Bit

Definition at line 284 of file i2c\_common\_5411x.h.

**6.52.2.15 #define I2C\_INTENCLR\_MONIDLE (1 << 19)**

Monitor Idle Interrupt Clear Bit

Definition at line 283 of file i2c\_common\_5411x.h.

**6.52.2.16 #define I2C\_INTENCLR\_MONIDLE (1 << 19)**

Monitor Idle Interrupt Clear Bit

Definition at line 283 of file i2c\_common\_5411x.h.

**6.52.2.17 #define I2C\_INTENCLR\_MONOV (1 << 17)**

Monitor Overflow Interrupt Clear Bit

Definition at line 282 of file i2c\_common\_5411x.h.

**6.52.2.18 #define I2C\_INTENCLR\_MONOV (1 << 17)**

Monitor Overflow Interrupt Clear Bit

Definition at line 282 of file i2c\_common\_5411x.h.

**6.52.2.19 #define I2C\_INTENCLR\_MONRDY (1 << 16)**

Monitor Ready Interrupt Clear Bit

Definition at line 281 of file i2c\_common\_5411x.h.

**6.52.2.20 #define I2C\_INTENCLR\_MONRDY (1 << 16)**

Monitor Ready Interrupt Clear Bit

Definition at line 281 of file i2c\_common\_5411x.h.

**6.52.2.21 #define I2C\_INTENCLR\_MSTPENDING (1 << 0)**

Master Pending Interrupt Clear Bit

Definition at line 275 of file i2c\_common\_5411x.h.

**6.52.2.22 #define I2C\_INTENCLR\_MSTPENDING (1 << 0)**

Master Pending Interrupt Clear Bit

Definition at line 275 of file i2c\_common\_5411x.h.

**6.52.2.23 #define I2C\_INTENCLR\_MSTRARBLOSS (1 << 4)**

Master Arbitration Loss Interrupt Clear Bit

Definition at line 276 of file i2c\_common\_5411x.h.

**6.52.2.24 #define I2C\_INTENCLR\_MSTRARBLOSS (1 << 4)**

Master Arbitration Loss Interrupt Clear Bit

Definition at line 276 of file i2c\_common\_5411x.h.

**6.52.2.25 #define I2C\_INTENCLR\_MSTSTSTPERR (1 << 6)**

Master Start Stop Error Interrupt Clear Bit

Definition at line 277 of file i2c\_common\_5411x.h.

**6.52.2.26 #define I2C\_INTENCLR\_MSTSTSTPERR (1 << 6)**

Master Start Stop Error Interrupt Clear Bit

Definition at line 277 of file i2c\_common\_5411x.h.



**6.52.2.27 #define I2C\_INTENCLR\_SCLTIMEOUT (1 << 25)**

SCL Timeout Interrupt Clear Bit

Definition at line 285 of file i2c\_common\_5411x.h.

**6.52.2.28 #define I2C\_INTENCLR\_SCLTIMEOUT (1 << 25)**

SCL Timeout Interrupt Clear Bit

Definition at line 285 of file i2c\_common\_5411x.h.

**6.52.2.29 #define I2C\_INTENCLR\_SLVDESEL (1 << 15)**

Slave Deselect Interrupt Clear Bit

Definition at line 280 of file i2c\_common\_5411x.h.

**6.52.2.30 #define I2C\_INTENCLR\_SLVDESEL (1 << 15)**

Slave Deselect Interrupt Clear Bit

Definition at line 280 of file i2c\_common\_5411x.h.

**6.52.2.31 #define I2C\_INTENCLR\_SLVNOTSTR (1 << 11)**

Slave not stretching Clock Interrupt Clear Bit

Definition at line 279 of file i2c\_common\_5411x.h.

**6.52.2.32 #define I2C\_INTENCLR\_SLVNOTSTR (1 << 11)**

Slave not stretching Clock Interrupt Clear Bit

Definition at line 279 of file i2c\_common\_5411x.h.

**6.52.2.33 #define I2C\_INTENCLR\_SLVPENDING (1 << 8)**

Slave Pending Interrupt Clear Bit

Definition at line 278 of file i2c\_common\_5411x.h.

**6.52.2.34 #define I2C\_INTENCLR\_SLVPENDING (1 << 8)**

Slave Pending Interrupt Clear Bit

Definition at line 278 of file i2c\_common\_5411x.h.

**6.52.2.35 #define I2C\_INTENSET\_EVENTTIMEOUT (1 << 24)**

Event Timeout Interrupt Enable Bit

Definition at line 269 of file i2c\_common\_5411x.h.

**6.52.2.36 #define I2C\_INTENSET\_EVENTTIMEOUT (1 << 24)**

Event Timeout Interrupt Enable Bit

Definition at line 269 of file i2c\_common\_5411x.h.

**6.52.2.37 #define I2C\_INTENSET\_MONIDLE (1 << 19)**

Monitor Idle Interrupt Enable Bit

Definition at line 268 of file i2c\_common\_5411x.h.

**6.52.2.38 #define I2C\_INTENSET\_MONIDLE (1 << 19)**

Monitor Idle Interrupt Enable Bit

Definition at line 268 of file i2c\_common\_5411x.h.

**6.52.2.39 #define I2C\_INTENSET\_MONOV (1 << 17)**

Monitor Overflow Interrupt Enable Bit

Definition at line 267 of file i2c\_common\_5411x.h.

**6.52.2.40 #define I2C\_INTENSET\_MONOV (1 << 17)**

Monitor Overflow Interrupt Enable Bit

Definition at line 267 of file i2c\_common\_5411x.h.

**6.52.2.41 #define I2C\_INTENSET\_MONRDY (1 << 16)**

Monitor Ready Interrupt Enable Bit

Definition at line 266 of file i2c\_common\_5411x.h.

**6.52.2.42 #define I2C\_INTENSET\_MONRDY (1 << 16)**

Monitor Ready Interrupt Enable Bit

Definition at line 266 of file i2c\_common\_5411x.h.

**6.52.2.43 #define I2C\_INTENSET\_MSTPENDING (1 << 0)**

Master Pending Interrupt Enable Bit

Definition at line 260 of file i2c\_common\_5411x.h.

**6.52.2.44 #define I2C\_INTENSET\_MSTPENDING (1 << 0)**

Master Pending Interrupt Enable Bit

Definition at line 260 of file i2c\_common\_5411x.h.

**6.52.2.45 #define I2C\_INTENSET\_MSTRARBLOSS (1 << 4)**

Master Arbitration Loss Interrupt Enable Bit

Definition at line 261 of file i2c\_common\_5411x.h.

**6.52.2.46 #define I2C\_INTENSET\_MSTRARBLOSS (1 << 4)**

Master Arbitration Loss Interrupt Enable Bit

Definition at line 261 of file i2c\_common\_5411x.h.

**6.52.2.47 #define I2C\_INTENSET\_MSTSTSTPERR (1 << 6)**

Master Start Stop Error Interrupt Enable Bit

Definition at line 262 of file i2c\_common\_5411x.h.

**6.52.2.48 #define I2C\_INTENSET\_MSTSTSTPERR (1 << 6)**

Master Start Stop Error Interrupt Enable Bit

Definition at line 262 of file i2c\_common\_5411x.h.

**6.52.2.49 #define I2C\_INTENSET\_SCLTIMEOUT (1 << 25)**

SCL Timeout Interrupt Enable Bit

Definition at line 270 of file i2c\_common\_5411x.h.

**6.52.2.50 #define I2C\_INTENSET\_SCLTIMEOUT (1 << 25)**

SCL Timeout Interrupt Enable Bit

Definition at line 270 of file i2c\_common\_5411x.h.

**6.52.2.51 #define I2C\_INTENSET\_SLVDESEL (1 << 15)**

Slave Deselect Interrupt Enable Bit

Definition at line 265 of file i2c\_common\_5411x.h.

**6.52.2.52 #define I2C\_INTENSET\_SLVDESEL (1 << 15)**

Slave Deselect Interrupt Enable Bit

Definition at line 265 of file i2c\_common\_5411x.h.

**6.52.2.53 #define I2C\_INTENSET\_SLVNOTSTR (1 << 11)**

Slave not stretching Clock Interrupt Enable Bit

Definition at line 264 of file i2c\_common\_5411x.h.

**6.52.2.54 #define I2C\_INTENSET\_SLVNOTSTR (1 << 11)**

Slave not stretching Clock Interrupt Enable Bit

Definition at line 264 of file i2c\_common\_5411x.h.

**6.52.2.55 #define I2C\_INTENSET\_SLVPENDING (1 << 8)**

Slave Pending Interrupt Enable Bit

Definition at line 263 of file i2c\_common\_5411x.h.

**6.52.2.56 #define I2C\_INTENSET\_SLVPENDING (1 << 8)**

Slave Pending Interrupt Enable Bit

Definition at line 263 of file i2c\_common\_5411x.h.

**6.52.2.57 #define I2C\_INTSTAT\_EVENTTIMEOUT (1 << 24)**

Event Timeout Interrupt Status Bit

Definition at line 304 of file i2c\_common\_5411x.h.

**6.52.2.58 #define I2C\_INTSTAT\_EVENTTIMEOUT (1 << 24)**

Event Timeout Interrupt Status Bit

Definition at line 304 of file i2c\_common\_5411x.h.

**6.52.2.59 #define I2C\_INTSTAT\_MONIDLE (1 << 19)**

Monitor Idle Interrupt Status Bit

Definition at line 303 of file i2c\_common\_5411x.h.

**6.52.2.60 #define I2C\_INTSTAT\_MONIDLE (1 << 19)**

Monitor Idle Interrupt Status Bit

Definition at line 303 of file i2c\_common\_5411x.h.

**6.52.2.61 #define I2C\_INTSTAT\_MONOV (1 << 17)**

Monitor Overflow Interrupt Status Bit

Definition at line 302 of file i2c\_common\_5411x.h.

**6.52.2.62 #define I2C\_INTSTAT\_MONOV (1 << 17)**

Monitor Overflow Interrupt Status Bit

Definition at line 302 of file i2c\_common\_5411x.h.

**6.52.2.63 #define I2C\_INTSTAT\_MONRDY (1 << 16)**

Monitor Ready Interrupt Status Bit

Definition at line 301 of file i2c\_common\_5411x.h.

**6.52.2.64 #define I2C\_INTSTAT\_MONRDY (1 << 16)**

Monitor Ready Interrupt Status Bit

Definition at line 301 of file i2c\_common\_5411x.h.

**6.52.2.65 #define I2C\_INTSTAT\_MSTPENDING (1 << 0)**

Master Pending Interrupt Status Bit

Definition at line 295 of file i2c\_common\_5411x.h.

**6.52.2.66 #define I2C\_INTSTAT\_MSTPENDING (1 << 0)**

Master Pending Interrupt Status Bit

Definition at line 295 of file i2c\_common\_5411x.h.

**6.52.2.67 #define I2C\_INTSTAT\_MSTRARBLOSS (1 << 4)**

Master Arbitration Loss Interrupt Status Bit

Definition at line 296 of file i2c\_common\_5411x.h.

**6.52.2.68 #define I2C\_INTSTAT\_MSTRARBLOSS (1 << 4)**

Master Arbitration Loss Interrupt Status Bit

Definition at line 296 of file i2c\_common\_5411x.h.

**6.52.2.69 #define I2C\_INTSTAT\_MSTSTSTPERR (1 << 6)**

Master Start Stop Error Interrupt Status Bit

Definition at line 297 of file i2c\_common\_5411x.h.

**6.52.2.70 #define I2C\_INTSTAT\_MSTSTSTPERR (1 << 6)**

Master Start Stop Error Interrupt Status Bit

Definition at line 297 of file i2c\_common\_5411x.h.

**6.52.2.71 #define I2C\_INTSTAT\_SCLTIMEOUT (1 << 25)**

SCL Timeout Interrupt Status Bit

Definition at line 305 of file i2c\_common\_5411x.h.

**6.52.2.72 #define I2C\_INTSTAT\_SCLTIMEOUT (1 << 25)**

SCL Timeout Interrupt Status Bit

Definition at line 305 of file i2c\_common\_5411x.h.

**6.52.2.73 #define I2C\_INTSTAT\_SLVDESEL (1 << 15)**

Slave Deselect Interrupt Status Bit

Definition at line 300 of file i2c\_common\_5411x.h.

**6.52.2.74 #define I2C\_INTSTAT\_SLVDESEL (1 << 15)**

Slave Deselect Interrupt Status Bit

Definition at line 300 of file i2c\_common\_5411x.h.

**6.52.2.75 #define I2C\_INTSTAT\_SLVNOTSTR (1 << 11)**

Slave not stretching Clock Interrupt Status Bit

Definition at line 299 of file i2c\_common\_5411x.h.

**6.52.2.76 #define I2C\_INTSTAT\_SLVNOTSTR (1 << 11)**

Slave not stretching Clock Interrupt Status Bit

Definition at line 299 of file i2c\_common\_5411x.h.

**6.52.2.77 #define I2C\_INTSTAT\_SLVPENDING (1 << 8)**

Slave Pending Interrupt Status Bit

Definition at line 298 of file i2c\_common\_5411x.h.

**6.52.2.78 #define I2C\_INTSTAT\_SLVPENDING (1 << 8)**

Slave Pending Interrupt Status Bit

Definition at line 298 of file i2c\_common\_5411x.h.

**6.52.2.79 #define I2C\_MONRXDAT\_DATA (0xFF << 0)**

Monitor Function Receive Data Field

Definition at line 354 of file i2c\_common\_5411x.h.

**6.52.2.80 #define I2C\_MONRXDAT\_DATA (0xFF << 0)**

Monitor Function Receive Data Field

Definition at line 354 of file i2c\_common\_5411x.h.

**6.52.2.81 #define I2C\_MONRXDAT\_MONNACK (1 << 10)**

Monitor Received Nack Bit

Definition at line 357 of file i2c\_common\_5411x.h.

**6.52.2.82 #define I2C\_MONRXDAT\_MONNACK (1 << 10)**

Monitor Received Nack Bit

Definition at line 357 of file i2c\_common\_5411x.h.

**6.52.2.83 #define I2C\_MONRXDAT\_MONRESTART (1 << 9)**

Monitor Received Repeated Start Bit

Definition at line 356 of file i2c\_common\_5411x.h.

**6.52.2.84 #define I2C\_MONRXDAT\_MONRESTART (1 << 9)**

Monitor Received Repeated Start Bit

Definition at line 356 of file i2c\_common\_5411x.h.

**6.52.2.85 #define I2C\_MONRXDAT\_MONSTART (1 << 8)**

Monitor Received Start Bit

Definition at line 355 of file i2c\_common\_5411x.h.

**6.52.2.86 #define I2C\_MONRXDAT\_MONSTART (1 << 8)**

Monitor Received Start Bit

Definition at line 355 of file i2c\_common\_5411x.h.

**6.52.2.87 #define I2C\_MSTCTL\_MSTCONTINUE (1 << 0)**

Master Continue Bit

Definition at line 310 of file i2c\_common\_5411x.h.

**6.52.2.88 #define I2C\_MSTCTL\_MSTCONTINUE (1 << 0)**

Master Continue Bit

Definition at line 310 of file i2c\_common\_5411x.h.

**6.52.2.89 #define I2C\_MSTCTL\_MSTDMA (1 << 3)**

Master DMA Enable Bit

Definition at line 313 of file i2c\_common\_5411x.h.

**6.52.2.90 #define I2C\_MSTCTL\_MSTDMA (1 << 3)**

Master DMA Enable Bit

Definition at line 313 of file i2c\_common\_5411x.h.

**6.52.2.91 #define I2C\_MSTCTL\_MSTSTART (1 << 1)**

Master Start Control Bit

Definition at line 311 of file i2c\_common\_5411x.h.

**6.52.2.92 #define I2C\_MSTCTL\_MSTSTART (1 << 1)**

Master Start Control Bit

Definition at line 311 of file i2c\_common\_5411x.h.

**6.52.2.93 #define I2C\_MSTCTL\_MSTSTOP (1 << 2)**

Master Stop Control Bit

Definition at line 312 of file i2c\_common\_5411x.h.

**6.52.2.94 #define I2C\_MSTCTL\_MSTSTOP (1 << 2)**

Master Stop Control Bit

Definition at line 312 of file i2c\_common\_5411x.h.

**6.52.2.95 #define I2C\_MSTDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)**

Master data mask

Definition at line 324 of file i2c\_common\_5411x.h.

**6.52.2.96 #define I2C\_MSTDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)**

Master data mask

Definition at line 324 of file i2c\_common\_5411x.h.

**6.52.2.97 #define I2C\_MSTTIME\_MSTSCLHIGH (0x07 << 4)**

Master SCL High Time field

Definition at line 319 of file i2c\_common\_5411x.h.

**6.52.2.98 #define I2C\_MSTTIME\_MSTSCLHIGH (0x07 << 4)**

Master SCL High Time field

Definition at line 319 of file i2c\_common\_5411x.h.



6.52.2.99 `#define I2C_MSTTIME_MSTSCLOW (0x07 << 0)`

Master SCL Low Time field

Definition at line 318 of file `i2c_common_5411x.h`.

6.52.2.100 `#define I2C_MSTTIME_MSTSCLOW (0x07 << 0)`

Master SCL Low Time field

Definition at line 318 of file `i2c_common_5411x.h`.

6.52.2.101 `#define I2C_SLVADR_MASK ((uint32_t) 0x00FF)`

Slave Address Mask

Definition at line 343 of file `i2c_common_5411x.h`.

6.52.2.102 `#define I2C_SLVADR_MASK ((uint32_t) 0x00FF)`

Slave Address Mask

Definition at line 343 of file `i2c_common_5411x.h`.

6.52.2.103 `#define I2C_SLVADR_SADISABLE (1 << 0)`

Slave Address n Disable Bit

Definition at line 341 of file `i2c_common_5411x.h`.

6.52.2.104 `#define I2C_SLVADR_SADISABLE (1 << 0)`

Slave Address n Disable Bit

Definition at line 341 of file `i2c_common_5411x.h`.

6.52.2.105 `#define I2C_SLVADR_SLVADR (0x7F << 1)`

Slave Address field

Definition at line 342 of file `i2c_common_5411x.h`.

6.52.2.106 `#define I2C_SLVADR_SLVADR (0x7F << 1)`

Slave Address field

Definition at line 342 of file `i2c_common_5411x.h`.

6.52.2.107 `#define I2C_SLVCTL_SLVCONTINUE (1 << 0)`

Slave Continue Bit

Definition at line 329 of file `i2c_common_5411x.h`.

6.52.2.108 `#define I2C_SLVCTL_SLVCONTINUE (1 << 0)`

Slave Continue Bit

Definition at line 329 of file `i2c_common_5411x.h`.

6.52.2.109 `#define I2C_SLVCTL_SLVDMA (1 << 3)`

Slave DMA Enable Bit

Definition at line 331 of file `i2c_common_5411x.h`.

6.52.2.110 `#define I2C_SLVCTL_SLVDMA (1 << 3)`

Slave DMA Enable Bit

Definition at line 331 of file `i2c_common_5411x.h`.

6.52.2.111 `#define I2C_SLVCTL_SLVNACK (1 << 1)`

Slave NACK Bit

Definition at line 330 of file `i2c_common_5411x.h`.

6.52.2.112 `#define I2C_SLVCTL_SLVNACK (1 << 1)`

Slave NACK Bit

Definition at line 330 of file `i2c_common_5411x.h`.

6.52.2.113 `#define I2C_SLVDAT_DATAMASK ((uint32_t) 0x00FF << 0)`

Slave data mask

Definition at line 336 of file `i2c_common_5411x.h`.

6.52.2.114 `#define I2C_SLVDAT_DATAMASK ((uint32_t) 0x00FF << 0)`

Slave data mask

Definition at line 336 of file `i2c_common_5411x.h`.

6.52.2.115 `#define I2C_SLVQUAL_QUALMODE0 (1 << 0)`

Slave Qualifier Mode Enable Bit

Definition at line 348 of file `i2c_common_5411x.h`.

6.52.2.116 `#define I2C_SLVQUAL_QUALMODE0 (1 << 0)`

Slave Qualifier Mode Enable Bit

Definition at line 348 of file `i2c_common_5411x.h`.

6.52.2.117 `#define I2C_SLVQUAL_SLVQUAL0 (0x7F << 1)`

Slave Qualifier Address for Address 0

Definition at line 349 of file `i2c_common_5411x.h`.

6.52.2.118 `#define I2C_SLVQUAL_SLVQUAL0 (0x7F << 1)`

Slave Qualifier Address for Address 0

Definition at line 349 of file `i2c_common_5411x.h`.

6.52.2.119 `#define I2C_STAT_EVENTTIMEOUT (1 << 24)`

Event Timeout Interrupt Flag

Definition at line 244 of file `i2c_common_5411x.h`.

6.52.2.120 `#define I2C_STAT_EVENTTIMEOUT (1 << 24)`

Event Timeout Interrupt Flag

Definition at line 244 of file `i2c_common_5411x.h`.

6.52.2.121 `#define I2C_STAT_MONACTIVE (1 << 18)`

Monitor Active Flag

Definition at line 242 of file `i2c_common_5411x.h`.

6.52.2.122 `#define I2C_STAT_MONACTIVE (1 << 18)`

Monitor Active Flag

Definition at line 242 of file `i2c_common_5411x.h`.

6.52.2.123 `#define I2C_STAT_MONIDLE (1 << 19)`

Monitor Idle Flag

Definition at line 243 of file `i2c_common_5411x.h`.

6.52.2.124 `#define I2C_STAT_MONIDLE (1 << 19)`

Monitor Idle Flag

Definition at line 243 of file `i2c_common_5411x.h`.

6.52.2.125 `#define I2C_STAT_MONOV (1 << 17)`

Monitor Overflow Flag

Definition at line 241 of file `i2c_common_5411x.h`.

6.52.2.126 `#define I2C_STAT_MONOV (1 << 17)`

Monitor Overflow Flag

Definition at line 241 of file `i2c_common_5411x.h`.

6.52.2.127 `#define I2C_STAT_MONRDY (1 << 16)`

Monitor Ready Bit

Definition at line 240 of file `i2c_common_5411x.h`.

6.52.2.128 `#define I2C_STAT_MONRDY (1 << 16)`

Monitor Ready Bit

Definition at line 240 of file `i2c_common_5411x.h`.

6.52.2.129 `#define I2C_STAT_MSTCODE_IDLE (0)`

Master Idle State Code

Definition at line 247 of file `i2c_common_5411x.h`.

6.52.2.130 `#define I2C_STAT_MSTCODE_IDLE (0)`

Master Idle State Code

Definition at line 247 of file `i2c_common_5411x.h`.

6.52.2.131 `#define I2C_STAT_MSTCODE_NACKADR (3)`

Master NACK by slave on address State Code

Definition at line 250 of file `i2c_common_5411x.h`.

6.52.2.132 `#define I2C_STAT_MSTCODE_NACKADR (3)`

Master NACK by slave on address State Code

Definition at line 250 of file `i2c_common_5411x.h`.

6.52.2.133 `#define I2C_STAT_MSTCODE_NACKDAT (4)`

Master NACK by slave on data State Code

Definition at line 251 of file `i2c_common_5411x.h`.

6.52.2.134 `#define I2C_STAT_MSTCODE_NACKDAT (4)`

Master NACK by slave on data State Code

Definition at line 251 of file `i2c_common_5411x.h`.

**6.52.2.135 #define I2C\_STAT\_MSTCODE\_RXREADY (1)**

Master Receive Ready State Code

Definition at line 248 of file i2c\_common\_5411x.h.

**6.52.2.136 #define I2C\_STAT\_MSTCODE\_RXREADY (1)**

Master Receive Ready State Code

Definition at line 248 of file i2c\_common\_5411x.h.

**6.52.2.137 #define I2C\_STAT\_MSTCODE\_TXREADY (2)**

Master Transmit Ready State Code

Definition at line 249 of file i2c\_common\_5411x.h.

**6.52.2.138 #define I2C\_STAT\_MSTCODE\_TXREADY (2)**

Master Transmit Ready State Code

Definition at line 249 of file i2c\_common\_5411x.h.

**6.52.2.139 #define I2C\_STAT\_MSTPENDING (1 << 0)**

Master Pending Status Bit

Definition at line 230 of file i2c\_common\_5411x.h.

**6.52.2.140 #define I2C\_STAT\_MSTPENDING (1 << 0)**

Master Pending Status Bit

Definition at line 230 of file i2c\_common\_5411x.h.

**6.52.2.141 #define I2C\_STAT\_MSTRARBLOSS (1 << 4)**

Master Arbitration Loss Bit

Definition at line 232 of file i2c\_common\_5411x.h.

**6.52.2.142 #define I2C\_STAT\_MSTRARBLOSS (1 << 4)**

Master Arbitration Loss Bit

Definition at line 232 of file i2c\_common\_5411x.h.

**6.52.2.143 #define I2C\_STAT\_MSTSTATE (0x7 << 1)**

Master State Code

Definition at line 231 of file i2c\_common\_5411x.h.

6.52.2.144 `#define I2C_STAT_MSTSTATE (0x7 << 1)`

Master State Code

Definition at line 231 of file `i2c_common_5411x.h`.

6.52.2.145 `#define I2C_STAT_MSTSTSPERR (1 << 6)`

Master Start Stop Error Bit

Definition at line 233 of file `i2c_common_5411x.h`.

6.52.2.146 `#define I2C_STAT_MSTSTSPERR (1 << 6)`

Master Start Stop Error Bit

Definition at line 233 of file `i2c_common_5411x.h`.

6.52.2.147 `#define I2C_STAT_SCLTIMEOUT (1 << 25)`

SCL Timeout Interrupt Flag

Definition at line 245 of file `i2c_common_5411x.h`.

6.52.2.148 `#define I2C_STAT_SCLTIMEOUT (1 << 25)`

SCL Timeout Interrupt Flag

Definition at line 245 of file `i2c_common_5411x.h`.

6.52.2.149 `#define I2C_STAT_SLVCODE_ADDR (0)`

Master Idle State Code

Definition at line 253 of file `i2c_common_5411x.h`.

6.52.2.150 `#define I2C_STAT_SLVCODE_ADDR (0)`

Master Idle State Code

Definition at line 253 of file `i2c_common_5411x.h`.

6.52.2.151 `#define I2C_STAT_SLVCODE_RX (1)`

Received data is available Code

Definition at line 254 of file `i2c_common_5411x.h`.

6.52.2.152 `#define I2C_STAT_SLVCODE_RX (1)`

Received data is available Code

Definition at line 254 of file `i2c_common_5411x.h`.

**6.52.2.153 #define I2C\_STAT\_SLVCODE\_TX (2)**

Data can be transmitted Code

Definition at line 255 of file i2c\_common\_5411x.h.

**6.52.2.154 #define I2C\_STAT\_SLVCODE\_TX (2)**

Data can be transmitted Code

Definition at line 255 of file i2c\_common\_5411x.h.

**6.52.2.155 #define I2C\_STAT\_SLVDESEL (1 << 15)**

Slave Deselect Bit

Definition at line 239 of file i2c\_common\_5411x.h.

**6.52.2.156 #define I2C\_STAT\_SLVDESEL (1 << 15)**

Slave Deselect Bit

Definition at line 239 of file i2c\_common\_5411x.h.

**6.52.2.157 #define I2C\_STAT\_SLVIDX (0x3 << 12)**

Slave Address Index

Definition at line 237 of file i2c\_common\_5411x.h.

**6.52.2.158 #define I2C\_STAT\_SLVIDX (0x3 << 12)**

Slave Address Index

Definition at line 237 of file i2c\_common\_5411x.h.

**6.52.2.159 #define I2C\_STAT\_SLVNOTSTR (1 << 11)**

Slave not stretching Clock Bit

Definition at line 236 of file i2c\_common\_5411x.h.

**6.52.2.160 #define I2C\_STAT\_SLVNOTSTR (1 << 11)**

Slave not stretching Clock Bit

Definition at line 236 of file i2c\_common\_5411x.h.

**6.52.2.161 #define I2C\_STAT\_SLVPENDING (1 << 8)**

Slave Pending Status Bit

Definition at line 234 of file i2c\_common\_5411x.h.

6.52.2.162 `#define I2C_STAT_SLVPENDING (1 << 8)`

Slave Pending Status Bit

Definition at line 234 of file `i2c_common_5411x.h`.

6.52.2.163 `#define I2C_STAT_SLVSEL (1 << 14)`

Slave Selected Bit

Definition at line 238 of file `i2c_common_5411x.h`.

6.52.2.164 `#define I2C_STAT_SLVSEL (1 << 14)`

Slave Selected Bit

Definition at line 238 of file `i2c_common_5411x.h`.

6.52.2.165 `#define I2C_STAT_SLVSTATE (0x3 << 9)`

Slave State Code

Definition at line 235 of file `i2c_common_5411x.h`.

6.52.2.166 `#define I2C_STAT_SLVSTATE (0x3 << 9)`

Slave State Code

Definition at line 235 of file `i2c_common_5411x.h`.

6.52.2.167 `#define I2C_TIMEOUT_VAL( n ) (((uint32_t)((n) - 1) & 0xFFF0) | 0x000F)`

Macro for Timeout value register

Definition at line 290 of file `i2c_common_5411x.h`.

6.52.2.168 `#define I2C_TIMEOUT_VAL( n ) (((uint32_t)((n) - 1) & 0xFFF0) | 0x000F)`

Macro for Timeout value register

Definition at line 290 of file `i2c_common_5411x.h`.

## 6.52.3 Function Documentation

6.52.3.1 `static INLINE void Chip_I2C_ClearInt ( LPC_I2C_T * pl2C, uint32_t intClr )` `[static]`

Disable I2C Interrupts.

Parameters

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>intClr</i>	: ORed Value of I2C_INTENSET_* values to disable

Returns

Nothing



**Note**

It is recommended to use the [Chip\\_I2C\\_DisableInt\(\)](#) function instead of this function.

Definition at line 447 of file i2c\_common\_5411x.h.

**6.52.3.2** `__STATIC_INLINE void Chip_I2C_DeInit ( LPC_I2C_T * pl2C )`

Shutdown I2C Interface.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

Nothing

**Note**

This function disables the I2C clock for both the master and slave interfaces if the I2C channel.

Definition at line 379 of file i2c\_common\_5411x.h.

**6.52.3.3** `static INLINE void Chip_I2C_DisableInt ( LPC_I2C_T * pl2C, uint32_t intClr ) [static]`

Disable I2C Interrupts.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>intClr</i>	: ORed Value of I2C_INTENSET_* values to disable

**Returns**

Nothing

Definition at line 434 of file i2c\_common\_5411x.h.

**6.52.3.4** `static INLINE void Chip_I2C_EnableInt ( LPC_I2C_T * pl2C, uint32_t intEn ) [static]`

Enable I2C Interrupts.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
<i>intEn</i>	: ORed Value of I2C_INTENSET_* values to enable

**Returns**

Nothing

Definition at line 423 of file i2c\_common\_5411x.h.

**6.52.3.5** `static INLINE uint32_t Chip_I2C_GetClockDiv ( LPC_I2C_T * pl2C ) [static]`

Get I2C Clock Divider registers.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

Clock Divider value

**Note**

Return the divider value for the I2C block It is the CLKDIV register value + 1

Definition at line 412 of file i2c\_common\_5411x.h.

**6.52.3.6** `static INLINE uint32_t Chip_I2C_GetPendingInt ( LPC_I2C_T * pl2C ) [static]`

Returns pending I2C Interrupts.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

All pending interrupts, mask with I2C\_INTENSET\_\* to determine specific interrupts

Definition at line 457 of file i2c\_common\_5411x.h.

**6.52.3.7** `__STATIC_INLINE int Chip_I2C_Init ( LPC_I2C_T * pl2C )`

Initialize I2C Interface.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

**Returns**

0 - on success; [ERR\\_FLEXCOMM\\_FUNCNOTSUPPORTED](#) or [ERR\\_FLEXCOMM\\_NOTFREE](#) on failure

**Note**

This function enables the I2C clock for both the master and slave interfaces if the I2C channel.

Definition at line 367 of file i2c\_common\_5411x.h.

**6.52.3.8** `__STATIC_INLINE void Chip_I2C_SetClockDiv ( LPC_I2C_T * pl2C, uint32_t clkdiv )`

Sets I2C Clock Divider registers.

**Parameters**

<i>pl2C</i>	: Pointer to selected I2C peripheral
-------------	--------------------------------------

<i>clkdiv</i>	: Clock Divider value for I2C, value is between (1 - 65536)
---------------	---

**Returns**

Nothing

**Note**

The clock to I2C block is determined by the following formula (I2C\_PCLK is the frequency of the system clock):

$I2C \text{ Clock Frequency} = (I2C\_PCLK) / clkdiv;$

This divider must be setup for both the master and slave modes of the controller.

Definition at line 395 of file `i2c_common_5411x.h`.

**6.52.4 Variable Documentation****6.52.4.1 `__IO uint32_t CLKDIV`**

I2C Clock Divider Register

Definition at line 55 of file `i2c_common_5411x.h`.

**6.52.4.2 `__O uint32_t INTENCLR`**

I2C Interrupt Enable Clear Register common for Master, Slave and Monitor

Definition at line 53 of file `i2c_common_5411x.h`.

**6.52.4.3 `__IO uint32_t INTENSET`**

I2C Interrupt Enable Set Register common for Master, Slave and Monitor

Definition at line 52 of file `i2c_common_5411x.h`.

**6.52.4.4 `__IO uint32_t INTSTAT`**

I2C Interrupt Status Register

Definition at line 56 of file `i2c_common_5411x.h`.

**6.52.4.5 `LPC_I2C_T`**

Definition at line 73 of file `i2c_common_5411x.h`.

**6.52.4.6 `__I uint32_t MONRXDAT`**

I2C Monitor Data Register

Definition at line 67 of file `i2c_common_5411x.h`.

**6.52.4.7 `__IO uint32_t MSTCTL`**

I2C Master Control Register

Definition at line 58 of file `i2c_common_5411x.h`.

#### 6.52.4.8 `__IO uint32_t MSTDAT`

I2C Master Data Register

Definition at line 60 of file `i2c_common_5411x.h`.

#### 6.52.4.9 `__IO uint32_t MSTTIME`

I2C Master Time Register for SCL

Definition at line 59 of file `i2c_common_5411x.h`.

#### 6.52.4.10 `__I uint32_t PID`

Offset: 0xFFC Module identification register

Definition at line 72 of file `i2c_common_5411x.h`.

#### 6.52.4.11 `__IO uint32_t PSELID`

Offset: 0xFF8 Peripheral select/identification register

Definition at line 71 of file `i2c_common_5411x.h`.

#### 6.52.4.12 `__I uint32_t RESERVED0`

Definition at line 57 of file `i2c_common_5411x.h`.

#### 6.52.4.13 `__IO uint32_t RESERVED1[5]`

Definition at line 61 of file `i2c_common_5411x.h`.

#### 6.52.4.14 `__IO uint32_t RESERVED2[9]`

Definition at line 66 of file `i2c_common_5411x.h`.

#### 6.52.4.15 `__IO uint32_t SLVADR[4]`

I2C Slave Address Registers

Definition at line 64 of file `i2c_common_5411x.h`.

#### 6.52.4.16 `__IO uint32_t SLVCTL`

I2C Slave Control Register

Definition at line 62 of file `i2c_common_5411x.h`.

#### 6.52.4.17 `__IO uint32_t SLVDAT`

I2C Slave Data Register

Definition at line 63 of file `i2c_common_5411x.h`.

**6.52.4.18 \_\_IO uint32\_t SLVQUAL0**

I2C Slave Address Qualifier 0 Register

Definition at line 65 of file i2c\_common\_5411x.h.

**6.52.4.19 \_\_IO uint32\_t STAT**

I2C register block structure.

I2C Configuration Register common for Master, Slave and Monitor I2C Status Register common for Master, Slave and Monitor

Definition at line 51 of file i2c\_common\_5411x.h.

**6.52.4.20 \_\_IO uint32\_t TIMEOUT**

I2C Timeout value Register

Definition at line 54 of file i2c\_common\_5411x.h.

## 6.53 CHIP: RTC tick to (a more) Universal Time conversion functions

### 6.53.1 Detailed Description

This driver converts between a RTC 1-second tick value and a Universal time format in a structure of type 'struct tm'.

#### Macros

- `#define TM_YEAR_BASE` (1970)
- `#define TM_DAYOFWEEK` (4)

#### Functions

- void `ConvertRtcTime` (uint32\_t rtcTick, struct tm \*pTime)  
*Converts a RTC tick time to Universal time.*
- void `ConvertTimeRtc` (struct tm \*pTime, uint32\_t \*rtcTick)  
*Converts a Universal time to RTC tick time.*

### 6.53.2 Macro Definition Documentation

#### 6.53.2.1 `#define TM_DAYOFWEEK` (4)

Definition at line 54 of file rtc\_ut.h.

#### 6.53.2.2 `#define TM_YEAR_BASE` (1970)

Definition at line 53 of file rtc\_ut.h.

### 6.53.3 Function Documentation

#### 6.53.3.1 void `ConvertRtcTime` ( uint32\_t *rtcTick*, struct tm \* *pTime* )

Converts a RTC tick time to Universal time.

##### Parameters

<i>rtcTick</i>	: Current RTC time value
<i>pTime</i>	: Pointer to time structure to fill

##### Returns

Nothing

##### Note

When setting time, the 'tm\_wday', 'tm\_yday', and 'tm\_isdst' fields are not used.

Definition at line 130 of file rtc\_ut.c.

#### 6.53.3.2 void `ConvertTimeRtc` ( struct tm \* *pTime*, uint32\_t \* *rtcTick* )

Converts a Universal time to RTC tick time.

## Parameters

<i>pTime</i>	: Pointer to time structure to use
<i>rtcTick</i>	: Pointer to RTC time value to fill

## Returns

Nothing

## Note

When converting time, the 'tm\_isdst' field is not populated by the conversion function.

Definition at line 154 of file rtc\_ut.c.

## 6.54 CHIP: Simple ring buffer implementation

### 6.54.1 Detailed Description

#### Data Structures

- struct [RINGBUFF\\_T](#)  
*Ring buffer structure.*

#### Macros

- #define [RB\\_VHEAD](#)(rb) (\*(volatile uint32\_t \*) &(rb)->head)
- #define [RB\\_VTAIL](#)(rb) (\*(volatile uint32\_t \*) &(rb)->tail)

#### Functions

- int [RingBuffer\\_Init](#) ([RINGBUFF\\_T](#) \*RingBuff, void \*buffer, int itemSize, int count, void \*(\*cpyFunc)(void \*dst, const void \*src, uint32\_t len))  
*Initialize ring buffer.*
- \_\_STATIC\_INLINE void [RingBuffer\\_Flush](#) ([RINGBUFF\\_T](#) \*RingBuff)  
*Resets the ring buffer to empty.*
- \_\_STATIC\_INLINE int [RingBuffer\\_GetSize](#) ([RINGBUFF\\_T](#) \*RingBuff)  
*Return size the ring buffer.*
- \_\_STATIC\_INLINE int [RingBuffer\\_GetCount](#) ([RINGBUFF\\_T](#) \*RingBuff)  
*Return number of items in the ring buffer.*
- \_\_STATIC\_INLINE int [RingBuffer\\_GetFree](#) ([RINGBUFF\\_T](#) \*RingBuff)  
*Return number of free items in the ring buffer.*
- \_\_STATIC\_INLINE int [RingBuffer\\_IsFull](#) ([RINGBUFF\\_T](#) \*RingBuff)  
*Return number of items in the ring buffer.*
- \_\_STATIC\_INLINE int [RingBuffer\\_IsEmpty](#) ([RINGBUFF\\_T](#) \*RingBuff)  
*Return empty status of ring buffer.*
- int [RingBuffer\\_Insert](#) ([RINGBUFF\\_T](#) \*RingBuff, const void \*data)  
*Insert a single item into ring buffer.*
- int [RingBuffer\\_InsertMult](#) ([RINGBUFF\\_T](#) \*RingBuff, const void \*data, int num)  
*Insert an array of items into ring buffer.*
- int [RingBuffer\\_Pop](#) ([RINGBUFF\\_T](#) \*RingBuff, void \*data)  
*Pop an item from the ring buffer.*
- int [RingBuffer\\_PopMult](#) ([RINGBUFF\\_T](#) \*RingBuff, void \*data, int num)  
*Pop an array of items from the ring buffer.*

### 6.54.2 Macro Definition Documentation

#### 6.54.2.1 #define RB\_VHEAD( rb ) (\*(volatile uint32\_t \*) &(rb)->head)

volatile typecasted head index

Definition at line 59 of file ring\_buffer.h.

#### 6.54.2.2 #define RB\_VTAIL( rb ) (\*(volatile uint32\_t \*) &(rb)->tail)

volatile typecasted tail index

Definition at line 65 of file ring\_buffer.h.



### 6.54.3 Function Documentation

#### 6.54.3.1 `__STATIC_INLINE void RingBuffer_Flush ( RINGBUFF_T * RingBuff )`

Resets the ring buffer to empty.

##### Parameters

<i>RingBuff</i>	: Pointer to ring buffer
-----------------	--------------------------

##### Returns

Nothing

Definition at line 90 of file ring\_buffer.h.

#### 6.54.3.2 `__STATIC_INLINE int RingBuffer_GetCount ( RINGBUFF_T * RingBuff )`

Return number of items in the ring buffer.

##### Parameters

<i>RingBuff</i>	: Pointer to ring buffer
-----------------	--------------------------

##### Returns

Number of items in the ring buffer

Definition at line 110 of file ring\_buffer.h.

#### 6.54.3.3 `__STATIC_INLINE int RingBuffer_GetFree ( RINGBUFF_T * RingBuff )`

Return number of free items in the ring buffer.

##### Parameters

<i>RingBuff</i>	: Pointer to ring buffer
-----------------	--------------------------

##### Returns

Number of free items in the ring buffer

Definition at line 120 of file ring\_buffer.h.

#### 6.54.3.4 `__STATIC_INLINE int RingBuffer_GetSize ( RINGBUFF_T * RingBuff )`

Return size the ring buffer.

##### Parameters

<i>RingBuff</i>	: Pointer to ring buffer
-----------------	--------------------------

##### Returns

Size of the ring buffer in bytes

Definition at line 100 of file ring\_buffer.h.

6.54.3.5 `int RingBuffer_Init ( RINGBUFF_T * RingBuff, void * buffer, int itemSize, int count, void (*)(void *dst, const void *src, uint32_t len) cpyFunc )`

Initialize ring buffer.

## Parameters

<i>RingBuff</i>	: Pointer to ring buffer to initialize
<i>buffer</i>	: Pointer to buffer to associate with RingBuff
<i>itemSize</i>	: Size of each buffer item size
<i>count</i>	: Size of ring buffer
<i>cpyFunc</i>	: Call-back function that copies data (if NULL library <i>memcpy</i> will be used)

## Note

Memory pointed by *buffer* must have correct alignment of *itemSize*, and *count* must be a power of 2 and must at least be 2 or greater. *len* of the *cpyFunc* is in bytes.

## Returns

Nothing

Definition at line 55 of file ring\_buffer.c.

#### 6.54.3.6 int RingBuffer\_Insert ( RINGBUFF\_T \* RingBuff, const void \* data )

Insert a single item into ring buffer.

## Parameters

<i>RingBuff</i>	: Pointer to ring buffer
<i>data</i>	: pointer to item

## Returns

1 when successfully inserted, 0 on error (Buffer not initialized using [RingBuffer\\_Init\(\)](#) or attempted to insert when buffer is full)

Definition at line 74 of file ring\_buffer.c.

#### 6.54.3.7 int RingBuffer\_InsertMult ( RINGBUFF\_T \* RingBuff, const void \* data, int num )

Insert an array of items into ring buffer.

## Parameters

<i>RingBuff</i>	: Pointer to ring buffer
<i>data</i>	: Pointer to first element of the item array
<i>num</i>	: Number of items in the array

## Returns

number of items successfully inserted, 0 on error (Buffer not initialized using [RingBuffer\\_Init\(\)](#) or attempted to insert when buffer is full)

Definition at line 91 of file ring\_buffer.c.

#### 6.54.3.8 \_\_STATIC\_INLINE int RingBuffer\_IsEmpty ( RINGBUFF\_T \* RingBuff )

Return empty status of ring buffer.

## Parameters

<i>RingBuff</i>	: Pointer to ring buffer
-----------------	--------------------------

## Returns

1 if the ring buffer is empty, otherwise 0

Definition at line 140 of file ring\_buffer.h.

#### 6.54.3.9 `__STATIC_INLINE int RingBuffer_IsFull ( RINGBUFF_T * RingBuff )`

Return number of items in the ring buffer.

## Parameters

<i>RingBuff</i>	: Pointer to ring buffer
-----------------	--------------------------

## Returns

1 if the ring buffer is full, otherwise 0

Definition at line 130 of file ring\_buffer.h.

#### 6.54.3.10 `int RingBuffer_Pop ( RINGBUFF_T * RingBuff, void * data )`

Pop an item from the ring buffer.

## Parameters

<i>RingBuff</i>	: Pointer to ring buffer
<i>data</i>	: Pointer to memory where popped item be stored

## Returns

1 when item popped successfully onto *data*, 0 When error (Buffer not initialized using [RingBuffer\\_Init\(\)](#) or attempted to pop item when the buffer is empty)

Definition at line 129 of file ring\_buffer.c.

#### 6.54.3.11 `int RingBuffer_PopMult ( RINGBUFF_T * RingBuff, void * data, int num )`

Pop an array of items from the ring buffer.

## Parameters

<i>RingBuff</i>	: Pointer to ring buffer
<i>data</i>	: Pointer to memory where popped items be stored
<i>num</i>	: Max number of items array <i>data</i> can hold

## Returns

Number of items popped onto *data*, 0 on error (Buffer not initialized using [RingBuffer\\_Init\(\)](#) or attempted to pop when the buffer is empty)

Definition at line 146 of file ring\_buffer.c.

## 6.55 CHIP: Stopwatch primitives.

### 6.55.1 Detailed Description

#### Functions

- void [StopWatch\\_Init](#) (void)  
*Initialize stopwatch.*
- uint32\_t [StopWatch\\_Start](#) (void)  
*Start a stopwatch.*
- \_\_STATIC\_INLINE uint32\_t [StopWatch\\_Elapsed](#) (uint32\_t startTime)  
*Returns number of ticks elapsed since stopwatch was started.*
- uint32\_t [StopWatch\\_TicksPerSecond](#) (void)  
*Returns number of ticks per second of the stopwatch timer.*
- uint32\_t [StopWatch\\_TicksToMs](#) (uint32\_t ticks)  
*Converts from stopwatch ticks to mS.*
- uint32\_t [StopWatch\\_TicksToUs](#) (uint32\_t ticks)  
*Converts from stopwatch ticks to uS.*
- uint32\_t [StopWatch\\_MsToTicks](#) (uint32\_t mS)  
*Converts from mS to stopwatch ticks.*
- uint32\_t [StopWatch\\_UsToTicks](#) (uint32\_t uS)  
*Converts from uS to stopwatch ticks.*
- \_\_STATIC\_INLINE void [StopWatch\\_DelayTicks](#) (uint32\_t ticks)  
*Delays the given number of ticks using stopwatch primitives.*
- \_\_STATIC\_INLINE void [StopWatch\\_DelayMs](#) (uint32\_t mS)  
*Delays the given number of mS using stopwatch primitives.*
- \_\_STATIC\_INLINE void [StopWatch\\_DelayUs](#) (uint32\_t uS)  
*Delays the given number of uS using stopwatch primitives.*

### 6.55.2 Function Documentation

#### 6.55.2.1 \_\_STATIC\_INLINE void StopWatch\_DelayMs ( uint32\_t mS )

Delays the given number of mS using stopwatch primitives.

##### Parameters

<i>mS</i>	: Number of mS to delay
-----------	-------------------------

##### Returns

Nothing

Definition at line 114 of file stopwatch.h.

#### 6.55.2.2 \_\_STATIC\_INLINE void StopWatch\_DelayTicks ( uint32\_t ticks )

Delays the given number of ticks using stopwatch primitives.

**Parameters**

<i>ticks</i>	: Number of ticks to delay
--------------	----------------------------

**Returns**

Nothing

Definition at line 103 of file stopwatch.h.

**6.55.2.3   \_\_STATIC\_INLINE void StopWatch\_DelayUs ( uint32\_t uS )**

Delays the given number of uS using stopwatch primitives.

**Parameters**

<i>uS</i>	: Number of uS to delay
-----------	-------------------------

**Returns**

Nothing

Definition at line 126 of file stopwatch.h.

**6.55.2.4   \_\_STATIC\_INLINE uint32\_t StopWatch\_Elapsed ( uint32\_t startTime )**

Returns number of ticks elapsed since stopwatch was started.

**Parameters**

<i>startTime</i>	: Time returned by <a href="#">StopWatch_Start()</a> .
------------------	--

**Returns**

Number of ticks elapsed since stopwatch was started

Definition at line 59 of file stopwatch.h.

**6.55.2.5   void StopWatch\_Init ( void )**

Initialize stopwatch.

**Returns**

Nothing

Definition at line 60 of file stopwatch\_5411x.c.

**6.55.2.6   uint32\_t StopWatch\_MsToTicks ( uint32\_t mS )**

Converts from mS to stopwatch ticks.

**Parameters**

---

<i>mS</i>	: Duration in mS to convert to ticks.
-----------	---------------------------------------

**Returns**

Number of ticks in given number of mS

Definition at line 100 of file stopwatch\_5411x.c.

**6.55.2.7 uint32\_t StopWatch\_Start ( void )**

Start a stopwatch.

**Returns**

Current cycle count

Definition at line 75 of file stopwatch\_5411x.c.

**6.55.2.8 uint32\_t StopWatch\_TicksPerSecond ( void )**

Returns number of ticks per second of the stopwatch timer.

**Returns**

Number of ticks per second of the stopwatch timer

Definition at line 82 of file stopwatch\_5411x.c.

**6.55.2.9 uint32\_t StopWatch\_TicksToMs ( uint32\_t ticks )**

Converts from stopwatch ticks to mS.

**Parameters**

<i>ticks</i>	: Duration in ticks to convert to mS.
--------------	---------------------------------------

**Returns**

Number of mS in given number of ticks

Definition at line 88 of file stopwatch\_5411x.c.

**6.55.2.10 uint32\_t StopWatch\_TicksToUs ( uint32\_t ticks )**

Converts from stopwatch ticks to uS.

**Parameters**

<i>ticks</i>	: Duration in ticks to convert to uS.
--------------	---------------------------------------

**Returns**

Number of uS in given number of ticks

Definition at line 94 of file stopwatch\_5411x.c.

**6.55.2.11 uint32\_t StopWatch\_UsToTicks ( uint32\_t uS )**

Converts from uS to stopwatch ticks.

**Parameters**

<i>uS</i>	: Duration in uS to convert to ticks.
-----------	---------------------------------------

**Returns**

Number of ticks in given number of uS

Definition at line 106 of file stopwatch\_5411x.c.



## 6.56 CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers

### 6.56.1 Detailed Description

#### Enumerations

- enum `LPC5411X_M0_IRQn_Type` {  
`Reset_IRQn` = -15, `NonMaskableInt_IRQn` = -14, `HardFault_IRQn` = -13, `SVCall_IRQn` = -5,  
`PendSV_IRQn` = -2, `SysTick_IRQn` = -1, `WDTBOD_IRQn`, `DMA_IRQn`,  
`GINT0_IRQn`, `GINT1_IRQn`, `PIN_INT0_IRQn`, `PIN_INT1_IRQn`,  
`PIN_INT2_IRQn`, `PIN_INT3_IRQn`, `UTICK_IRQn`, `MRT_IRQn`,  
`CT32B0_IRQn`, `CT32B1_IRQn`, `SCT0_IRQn`, `CT32B3_IRQn`,  
`FLEXCOMM0_IRQn`, `FLEXCOMM1_IRQn`, `FLEXCOMM2_IRQn`, `FLEXCOMM3_IRQn`,  
`FLEXCOMM4_IRQn`, `FLEXCOMM5_IRQn`, `FLEXCOMM6_IRQn`, `FLEXCOMM7_IRQn`,  
`ADC_SEQA_IRQn`, `ADC_SEQB_IRQn`, `ADC_THCMP_IRQn`, `DMIC_IRQn`,  
`HWVAD`, `USBACT_IRQn`, `USB_IRQn`, `RTC_IRQn`,  
`Reserved_IRQn`, `MAILBOX_IRQn` }

### 6.56.2 Enumeration Type Documentation

#### 6.56.2.1 enum LPC5411X\_M0\_IRQn\_Type

##### Enumerator

- Reset\_IRQn*** 1 Reset Vector, invoked on Power up and warm reset
- NonMaskableInt\_IRQn*** 2 Non Maskable Interrupt
- HardFault\_IRQn*** 3 Cortex-M0 Hard Fault Interrupt
- SVCall\_IRQn*** 11 Cortex-M0 SV Call Interrupt
- PendSV\_IRQn*** 14 Cortex-M0 Pend SV Interrupt
- SysTick\_IRQn*** 15 Cortex-M0 System Tick Interrupt
- WDTBOD\_IRQn*** WWDT
- DMA\_IRQn*** DMA
- GINT0\_IRQn*** GINT0
- GINT1\_IRQn*** GINT1
- PIN\_INT0\_IRQn*** PININT0
- PIN\_INT1\_IRQn*** PININT1
- PIN\_INT2\_IRQn*** PININT2
- PIN\_INT3\_IRQn*** PININT3
- UTICK\_IRQn*** Micro-tick Timer interrupt
- MRT\_IRQn*** Multi-rate timer interrupt
- CT32B0\_IRQn*** CTMR0
- CT32B1\_IRQn*** CTMR1
- SCT0\_IRQn*** SCT
- CT32B3\_IRQn*** CTMR3
- FLEXCOMM0\_IRQn*** FLEXCOMM0
- FLEXCOMM1\_IRQn*** FLEXCOMM1
- FLEXCOMM2\_IRQn*** FLEXCOMM2
- FLEXCOMM3\_IRQn*** FLEXCOMM3
- FLEXCOMM4\_IRQn*** FLEXCOMM4
- FLEXCOMM5\_IRQn*** FLEXCOMM5

***FLEXCOMM6\_IRQn*** FLEXCOMM6  
***FLEXCOMM7\_IRQn*** FLEXCOMM7  
***ADC\_SEQA\_IRQn*** ADC0 sequence A completion  
***ADC\_SEQB\_IRQn*** ADC0 sequence B completion  
***ADC\_THCMP\_IRQn*** ADC0 threshold compare and error  
***DMIC\_IRQn*** Digital Mic  
***HWVAD*** Hardware Voice activity detect  
***USBACT\_IRQn*** USB Activity  
***USB\_IRQn*** USB  
***RTC\_IRQn*** RTC alarm and wake-up interrupts  
***Reserved\_IRQn*** Reserved Interrupt  
***MAILBOX\_IRQn*** Mailbox

Definition at line 77 of file cmsis\_5411x\_m0.h.

## 6.57 CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers

### 6.57.1 Detailed Description

#### Enumerations

- enum `LPC5411X_IRQn_Type` {  
`Reset_IRQn` = -15, `NonMaskableInt_IRQn` = -14, `HardFault_IRQn` = -13, `MemoryManagement_IRQn` = -12,  
`BusFault_IRQn` = -11, `UsageFault_IRQn` = -10, `SVCALL_IRQn` = -5, `DebugMonitor_IRQn` = -4,  
`PendSV_IRQn` = -2, `SysTick_IRQn` = -1, `WDTBOD_IRQn`, `DMA_IRQn`,  
`GINT0_IRQn`, `GINT1_IRQn`, `PIN_INT0_IRQn`, `PIN_INT1_IRQn`,  
`PIN_INT2_IRQn`, `PIN_INT3_IRQn`, `UTICK_IRQn`, `MRT_IRQn`,  
`CT32B0_IRQn`, `CT32B1_IRQn`, `SCT0_IRQn`, `CT32B3_IRQn`,  
`FLEXCOMM0_IRQn`, `FLEXCOMM1_IRQn`, `FLEXCOMM2_IRQn`, `FLEXCOMM3_IRQn`,  
`FLEXCOMM4_IRQn`, `FLEXCOMM5_IRQn`, `FLEXCOMM6_IRQn`, `FLEXCOMM7_IRQn`,  
`ADC_SEQA_IRQn`, `ADC_SEQB_IRQn`, `ADC_THCMP_IRQn`, `DMIC_IRQn`,  
`HWVAD_IRQn`, `USBACT_IRQn`, `USB_IRQn`, `RTC_IRQn`,  
`Reserved_IRQn`, `MAILBOX_IRQn`, `PIN_INT4_IRQn`, `PIN_INT5_IRQn`,  
`PIN_INT6_IRQn`, `PIN_INT7_IRQn`, `CT32B2_IRQn`, `CT32B4_IRQn`,  
`Reserved1_IRQn`, `SPIFI_IRQn` }

### 6.57.2 Enumeration Type Documentation

#### 6.57.2.1 enum LPC5411X\_IRQn\_Type

##### Enumerator

- Reset\_IRQn*** 1 Reset Vector, invoked on Power up and warm reset
- NonMaskableInt\_IRQn*** 2 Non maskable Interrupt, cannot be stopped or preempted
- HardFault\_IRQn*** 3 Hard Fault, all classes of Fault
- MemoryManagement\_IRQn*** 4 Memory Management, MPU mismatch, including Access Violation and No Match
- BusFault\_IRQn*** 5 Bus Fault, Pre-Fetch-, Memory Access Fault, other address/memory related Fault
- UsageFault\_IRQn*** 6 Usage Fault, i.e. Undef Instruction, Illegal State Transition
- SVCALL\_IRQn*** 11 System Service Call via SVC instruction
- DebugMonitor\_IRQn*** 12 Debug Monitor
- PendSV\_IRQn*** 14 Pendable request for system service
- SysTick\_IRQn*** 15 System Tick Timer
- WDTBOD\_IRQn*** WWDT
- DMA\_IRQn*** DMA
- GINT0\_IRQn*** GINT0
- GINT1\_IRQn*** GINT1
- PIN\_INT0\_IRQn*** PININT0
- PIN\_INT1\_IRQn*** PININT1
- PIN\_INT2\_IRQn*** PININT2
- PIN\_INT3\_IRQn*** PININT3
- UTICK\_IRQn*** Micro-tick Timer interrupt
- MRT\_IRQn*** Multi-rate timer interrupt
- CT32B0\_IRQn*** CTMR0
- CT32B1\_IRQn*** CTMR1
- SCT0\_IRQn*** SCT

***CT32B3\_IRQn*** CTMR3  
***FLEXCOMM0\_IRQn*** FLEXCOMM0  
***FLEXCOMM1\_IRQn*** FLEXCOMM1  
***FLEXCOMM2\_IRQn*** FLEXCOMM2  
***FLEXCOMM3\_IRQn*** FLEXCOMM3  
***FLEXCOMM4\_IRQn*** FLEXCOMM4  
***FLEXCOMM5\_IRQn*** FLEXCOMM5  
***FLEXCOMM6\_IRQn*** FLEXCOMM6  
***FLEXCOMM7\_IRQn*** FLEXCOMM7  
***ADC\_SEQA\_IRQn*** ADC0 sequence A completion  
***ADC\_SEQB\_IRQn*** ADC0 sequence B completion  
***ADC\_THCMP\_IRQn*** ADC0 threshold compare and error  
***DMIC\_IRQn*** Digital Mic  
***HWVAD\_IRQn*** Hardware Voice activity detect  
***USBACT\_IRQn*** USB Activity  
***USB\_IRQn*** USB  
***RTC\_IRQn*** RTC alarm and wake-up interrupts  
***Reserved\_IRQn*** Reserved Interrupt  
***MAILBOX\_IRQn*** Mailbox  
***PIN\_INT4\_IRQn*** External Interrupt 4  
***PIN\_INT5\_IRQn*** External Interrupt 5  
***PIN\_INT6\_IRQn*** External Interrupt 6  
***PIN\_INT7\_IRQn*** External Interrupt 7  
***CT32B2\_IRQn*** CTMR2  
***CT32B4\_IRQn*** CTMR4  
***Reserved1\_IRQn*** Reserved Interrupt  
***SPIFI\_IRQn*** SPI Flash interface

Definition at line 75 of file cmsis\_5411x.h.

## 6.58 CMSIS Core Instruction Interface

Access to dedicated instructions

## 6.59 CMSIS Core Register Access Functions

## 6.60 CMSIS Global Defines

**IO Type Qualifiers** are used

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

## 6.61 CMSIS SIMD Intrinsics

Access to dedicated SIMD instructions



## 6.62 CMSIS support

ARM CMSIS DSP Library

## 6.63 Chip specific drivers

### 6.63.1 Detailed Description

Chip specific drivers are unique to a specific device. Chip specific drivers may use an IP driver as it's base driver or a custom implementation if that peripheral or IP on the chip is unique (ie, clocking).

#### Modules

- [Common components used with chip drivers](#)
- [LPC5411x Chip specific drivers](#)

## 6.64 Code Red LPCXpresso support in LPCOpen

LPCXpresso 6

## 6.65 Common FreeRTOS functions shared with multiple platforms

## 6.66 Common components used with chip drivers

### 6.66.1 Detailed Description

#### Modules

- [CHIP: Common Chip ISP/IAP commands and return codes](#)
- [CHIP: FPU initialization](#)
- [CHIP: LPC Common Types](#)
- [CHIP: RTC tick to \(a more\) Universal Time conversion functions](#)
- [CHIP: Simple ring buffer implementation](#)
- [CHIP: Stopwatch primitives.](#)

## 6.67 Community support for LPCOpen

You can also submit questions and comments on the LPCware.com forums at the:

[LPCware community forums](#).

If you find an issue with the LPCOpen code or an example, you can submit bug reports for LPCOpen code at:

[NXP Technical support](#)

Note: You will need to create a free LPCware.com account to use the community forums.

## 6.68 Copyright (C) 2013 NXP Semiconductors. All rights reserved.

Copyright (C) 2013 NXP Semiconductors. All rights reserved.

## 6.69 Core Debug Registers (CoreDebug)

### 6.69.1 Detailed Description

Cortex-M0+ Core Debug Registers (DCB registers, SHCSR, and DFSR) are only accessible over DAP and not via processor. Therefore they are not covered by the Cortex-M0 header file.

Type definitions for the Core Debug Registers.

#### Data Structures

- struct [CoreDebug\\_Type](#)

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

#### Macros

- #define [CoreDebug\\_DHCSR\\_DBGKEY\\_Pos](#) 16
- #define [CoreDebug\\_DHCSR\\_DBGKEY\\_Msk](#) (0xFFFFUL << CoreDebug\_DHCSR\_DBGKEY\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_RESET\\_ST\\_Pos](#) 25
- #define [CoreDebug\\_DHCSR\\_S\\_RESET\\_ST\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_RESET\_ST\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_RETIRE\\_ST\\_Pos](#) 24
- #define [CoreDebug\\_DHCSR\\_S\\_RETIRE\\_ST\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_RETIRE\_ST\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_LOCKUP\\_Pos](#) 19
- #define [CoreDebug\\_DHCSR\\_S\\_LOCKUP\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_LOCKUP\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_SLEEP\\_Pos](#) 18
- #define [CoreDebug\\_DHCSR\\_S\\_SLEEP\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_SLEEP\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_HALT\\_Pos](#) 17
- #define [CoreDebug\\_DHCSR\\_S\\_HALT\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_HALT\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_REGRDY\\_Pos](#) 16
- #define [CoreDebug\\_DHCSR\\_S\\_REGRDY\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_REGRDY\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_SNAPSTALL\\_Pos](#) 5
- #define [CoreDebug\\_DHCSR\\_C\\_SNAPSTALL\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_SNAPSTALL\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_MASKINTS\\_Pos](#) 3
- #define [CoreDebug\\_DHCSR\\_C\\_MASKINTS\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_MASKINTS\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_STEP\\_Pos](#) 2
- #define [CoreDebug\\_DHCSR\\_C\\_STEP\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_STEP\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_HALT\\_Pos](#) 1
- #define [CoreDebug\\_DHCSR\\_C\\_HALT\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_HALT\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_DEBUGEN\\_Pos](#) 0
- #define [CoreDebug\\_DHCSR\\_C\\_DEBUGEN\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_DEBUGEN\_Pos)
- #define [CoreDebug\\_DCRSR\\_REGWnR\\_Pos](#) 16
- #define [CoreDebug\\_DCRSR\\_REGWnR\\_Msk](#) (1UL << CoreDebug\_DCRSR\_REGWnR\_Pos)
- #define [CoreDebug\\_DCRSR\\_REGSEL\\_Pos](#) 0
- #define [CoreDebug\\_DCRSR\\_REGSEL\\_Msk](#) (0x1FUL << CoreDebug\_DCRSR\_REGSEL\_Pos)
- #define [CoreDebug\\_DEMCR\\_TRCENA\\_Pos](#) 24
- #define [CoreDebug\\_DEMCR\\_TRCENA\\_Msk](#) (1UL << CoreDebug\_DEMCR\_TRCENA\_Pos)
- #define [CoreDebug\\_DEMCR\\_MON\\_REQ\\_Pos](#) 19
- #define [CoreDebug\\_DEMCR\\_MON\\_REQ\\_Msk](#) (1UL << CoreDebug\_DEMCR\_MON\_REQ\_Pos)
- #define [CoreDebug\\_DEMCR\\_MON\\_STEP\\_Pos](#) 18
- #define [CoreDebug\\_DEMCR\\_MON\\_STEP\\_Msk](#) (1UL << CoreDebug\_DEMCR\_MON\_STEP\_Pos)
- #define [CoreDebug\\_DEMCR\\_MON\\_PEND\\_Pos](#) 17
- #define [CoreDebug\\_DEMCR\\_MON\\_PEND\\_Msk](#) (1UL << CoreDebug\_DEMCR\_MON\_PEND\_Pos)
- #define [CoreDebug\\_DEMCR\\_MON\\_EN\\_Pos](#) 16
- #define [CoreDebug\\_DEMCR\\_MON\\_EN\\_Msk](#) (1UL << CoreDebug\_DEMCR\_MON\_EN\_Pos)



- `#define CoreDebug_DEMCR_VC_HARDERR_Pos 10`
- `#define CoreDebug_DEMCR_VC_HARDERR_Msk (1UL << CoreDebug_DEMCR_VC_HARDERR_Pos)`
- `#define CoreDebug_DEMCR_VC_INTERR_Pos 9`
- `#define CoreDebug_DEMCR_VC_INTERR_Msk (1UL << CoreDebug_DEMCR_VC_INTERR_Pos)`
- `#define CoreDebug_DEMCR_VC_BUSERR_Pos 8`
- `#define CoreDebug_DEMCR_VC_BUSERR_Msk (1UL << CoreDebug_DEMCR_VC_BUSERR_Pos)`
- `#define CoreDebug_DEMCR_VC_STATERR_Pos 7`
- `#define CoreDebug_DEMCR_VC_STATERR_Msk (1UL << CoreDebug_DEMCR_VC_STATERR_Pos)`
- `#define CoreDebug_DEMCR_VC_CHKERR_Pos 6`
- `#define CoreDebug_DEMCR_VC_CHKERR_Msk (1UL << CoreDebug_DEMCR_VC_CHKERR_Pos)`
- `#define CoreDebug_DEMCR_VC_NOCPERR_Pos 5`
- `#define CoreDebug_DEMCR_VC_NOCPERR_Msk (1UL << CoreDebug_DEMCR_VC_NOCPERR_Pos)`
- `#define CoreDebug_DEMCR_VC_MMERR_Pos 4`
- `#define CoreDebug_DEMCR_VC_MMERR_Msk (1UL << CoreDebug_DEMCR_VC_MMERR_Pos)`
- `#define CoreDebug_DEMCR_VC_CORERESET_Pos 0`
- `#define CoreDebug_DEMCR_VC_CORERESET_Msk (1UL << CoreDebug_DEMCR_VC_CORERESET_Pos)`

## 6.69.2 Macro Definition Documentation

### 6.69.2.1 `#define CoreDebug_DCRSR_REGSEL_Msk (0x1FUL << CoreDebug_DCRSR_REGSEL_Pos)`

CoreDebug DCRSR: REGSEL Mask

Definition at line 1321 of file core\_cm4.h.

### 6.69.2.2 `#define CoreDebug_DCRSR_REGSEL_Pos 0`

CoreDebug DCRSR: REGSEL Position

Definition at line 1320 of file core\_cm4.h.

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

CoreDebug DCRSR: REGWnR Mask

Definition at line 1318 of file core\_cm4.h.

### 6.69.2.4 `#define CoreDebug_DCRSR_REGWnR_Pos 16`

CoreDebug DCRSR: REGWnR Position

Definition at line 1317 of file core\_cm4.h.

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

CoreDebug DEMCR: MON\_EN Mask

Definition at line 1337 of file core\_cm4.h.

### 6.69.2.6 `#define CoreDebug_DEMCR_MON_EN_Pos 16`

CoreDebug DEMCR: MON\_EN Position

Definition at line 1336 of file core\_cm4.h.

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

CoreDebug DEMCR: MON\_PEND Mask

Definition at line 1334 of file core\_cm4.h.

**6.69.2.8** `#define CoreDebug_DEMCR_MON_PEND_Pos 17`

CoreDebug DEMCR: MON\_PEND Position

Definition at line 1333 of file core\_cm4.h.

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

CoreDebug DEMCR: MON\_REQ Mask

Definition at line 1328 of file core\_cm4.h.

**6.69.2.10** `#define CoreDebug_DEMCR_MON_REQ_Pos 19`

CoreDebug DEMCR: MON\_REQ Position

Definition at line 1327 of file core\_cm4.h.

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

CoreDebug DEMCR: MON\_STEP Mask

Definition at line 1331 of file core\_cm4.h.

**6.69.2.12** `#define CoreDebug_DEMCR_MON_STEP_Pos 18`

CoreDebug DEMCR: MON\_STEP Position

Definition at line 1330 of file core\_cm4.h.

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

CoreDebug DEMCR: TRCENA Mask

Definition at line 1325 of file core\_cm4.h.

**6.69.2.14** `#define CoreDebug_DEMCR_TRCENA_Pos 24`

CoreDebug DEMCR: TRCENA Position

Definition at line 1324 of file core\_cm4.h.

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

CoreDebug DEMCR: VC\_BUSERR Mask

Definition at line 1346 of file core\_cm4.h.

**6.69.2.16 #define CoreDebug\_DEMCR\_VC\_BUSERR\_Pos 8**

CoreDebug DEMCR: VC\_BUSERR Position

Definition at line 1345 of file core\_cm4.h.

**6.69.2.17 #define CoreDebug\_DEMCR\_VC\_CHKERR\_Msk (1UL << CoreDebug\_DEMCR\_VC\_CHKERR\_Pos)**

CoreDebug DEMCR: VC\_CHKERR Mask

Definition at line 1352 of file core\_cm4.h.

**6.69.2.18 #define CoreDebug\_DEMCR\_VC\_CHKERR\_Pos 6**

CoreDebug DEMCR: VC\_CHKERR Position

Definition at line 1351 of file core\_cm4.h.

**6.69.2.19 #define CoreDebug\_DEMCR\_VC\_CORERESET\_Msk (1UL << CoreDebug\_DEMCR\_VC\_CORERESET\_Pos)**

CoreDebug DEMCR: VC\_CORERESET Mask

Definition at line 1361 of file core\_cm4.h.

**6.69.2.20 #define CoreDebug\_DEMCR\_VC\_CORERESET\_Pos 0**

CoreDebug DEMCR: VC\_CORERESET Position

Definition at line 1360 of file core\_cm4.h.

**6.69.2.21 #define CoreDebug\_DEMCR\_VC\_HARDERR\_Msk (1UL << CoreDebug\_DEMCR\_VC\_HARDERR\_Pos)**

CoreDebug DEMCR: VC\_HARDERR Mask

Definition at line 1340 of file core\_cm4.h.

**6.69.2.22 #define CoreDebug\_DEMCR\_VC\_HARDERR\_Pos 10**

CoreDebug DEMCR: VC\_HARDERR Position

Definition at line 1339 of file core\_cm4.h.

**6.69.2.23 #define CoreDebug\_DEMCR\_VC\_INTERR\_Msk (1UL << CoreDebug\_DEMCR\_VC\_INTERR\_Pos)**

CoreDebug DEMCR: VC\_INTERR Mask

Definition at line 1343 of file core\_cm4.h.

**6.69.2.24 #define CoreDebug\_DEMCR\_VC\_INTERR\_Pos 9**

CoreDebug DEMCR: VC\_INTERR Position

Definition at line 1342 of file core\_cm4.h.

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

CoreDebug DEMCR: VC\_MMERR Mask

Definition at line 1358 of file core\_cm4.h.

**6.69.2.26** `#define CoreDebug_DEMCR_VC_MMERR_Pos 4`

CoreDebug DEMCR: VC\_MMERR Position

Definition at line 1357 of file core\_cm4.h.

**6.69.2.27** `#define CoreDebug_DEMCR_VC_NOCERR_Msk (1UL << CoreDebug_DEMCR_VC_NOCERR_Pos)`

CoreDebug DEMCR: VC\_NOCERR Mask

Definition at line 1355 of file core\_cm4.h.

**6.69.2.28** `#define CoreDebug_DEMCR_VC_NOCERR_Pos 5`

CoreDebug DEMCR: VC\_NOCERR Position

Definition at line 1354 of file core\_cm4.h.

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

CoreDebug DEMCR: VC\_STATERR Mask

Definition at line 1349 of file core\_cm4.h.

**6.69.2.30** `#define CoreDebug_DEMCR_VC_STATERR_Pos 7`

CoreDebug DEMCR: VC\_STATERR Position

Definition at line 1348 of file core\_cm4.h.

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

CoreDebug DHCSR: C\_DEBUGEN Mask

Definition at line 1314 of file core\_cm4.h.

**6.69.2.32** `#define CoreDebug_DHCSR_C_DEBUGEN_Pos 0`

CoreDebug DHCSR: C\_DEBUGEN Position

Definition at line 1313 of file core\_cm4.h.

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

CoreDebug DHCSR: C\_HALT Mask

Definition at line 1311 of file core\_cm4.h.

**6.69.2.34 #define CoreDebug\_DHCSR\_C\_HALT\_Pos 1**

CoreDebug DHCSR: C\_HALT Position

Definition at line 1310 of file core\_cm4.h.

**6.69.2.35 #define CoreDebug\_DHCSR\_C\_MASKINTS\_Msk (1UL << CoreDebug\_DHCSR\_C\_MASKINTS\_Pos)**

CoreDebug DHCSR: C\_MASKINTS Mask

Definition at line 1305 of file core\_cm4.h.

**6.69.2.36 #define CoreDebug\_DHCSR\_C\_MASKINTS\_Pos 3**

CoreDebug DHCSR: C\_MASKINTS Position

Definition at line 1304 of file core\_cm4.h.

**6.69.2.37 #define CoreDebug\_DHCSR\_C\_SNAPSTALL\_Msk (1UL << CoreDebug\_DHCSR\_C\_SNAPSTALL\_Pos)**

CoreDebug DHCSR: C\_SNAPSTALL Mask

Definition at line 1302 of file core\_cm4.h.

**6.69.2.38 #define CoreDebug\_DHCSR\_C\_SNAPSTALL\_Pos 5**

CoreDebug DHCSR: C\_SNAPSTALL Position

Definition at line 1301 of file core\_cm4.h.

**6.69.2.39 #define CoreDebug\_DHCSR\_C\_STEP\_Msk (1UL << CoreDebug\_DHCSR\_C\_STEP\_Pos)**

CoreDebug DHCSR: C\_STEP Mask

Definition at line 1308 of file core\_cm4.h.

**6.69.2.40 #define CoreDebug\_DHCSR\_C\_STEP\_Pos 2**

CoreDebug DHCSR: C\_STEP Position

Definition at line 1307 of file core\_cm4.h.

**6.69.2.41 #define CoreDebug\_DHCSR\_DBGKEY\_Msk (0xFFFFUL << CoreDebug\_DHCSR\_DBGKEY\_Pos)**

CoreDebug DHCSR: DBGKEY Mask

Definition at line 1281 of file core\_cm4.h.

**6.69.2.42 #define CoreDebug\_DHCSR\_DBGKEY\_Pos 16**

CoreDebug DHCSR: DBGKEY Position

Definition at line 1280 of file core\_cm4.h.

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

CoreDebug DHCSR: S\_HALT Mask

Definition at line 1296 of file core\_cm4.h.

**6.69.2.44** `#define CoreDebug_DHCSR_S_HALT_Pos 17`

CoreDebug DHCSR: S\_HALT Position

Definition at line 1295 of file core\_cm4.h.

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

CoreDebug DHCSR: S\_LOCKUP Mask

Definition at line 1290 of file core\_cm4.h.

**6.69.2.46** `#define CoreDebug_DHCSR_S_LOCKUP_Pos 19`

CoreDebug DHCSR: S\_LOCKUP Position

Definition at line 1289 of file core\_cm4.h.

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

CoreDebug DHCSR: S\_REGRDY Mask

Definition at line 1299 of file core\_cm4.h.

**6.69.2.48** `#define CoreDebug_DHCSR_S_REGRDY_Pos 16`

CoreDebug DHCSR: S\_REGRDY Position

Definition at line 1298 of file core\_cm4.h.

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

CoreDebug DHCSR: S\_RESET\_ST Mask

Definition at line 1284 of file core\_cm4.h.

**6.69.2.50** `#define CoreDebug_DHCSR_S_RESET_ST_Pos 25`

CoreDebug DHCSR: S\_RESET\_ST Position

Definition at line 1283 of file core\_cm4.h.

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

CoreDebug DHCSR: S\_RETIRE\_ST Mask

Definition at line 1287 of file core\_cm4.h.

6.69.2.52 `#define CoreDebug_DHCSR_S_RETIRE_ST_Pos 24`

CoreDebug DHCSR: S\_RETIRE\_ST Position

Definition at line 1286 of file core\_cm4.h.

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

CoreDebug DHCSR: S\_SLEEP Mask

Definition at line 1293 of file core\_cm4.h.

6.69.2.54 `#define CoreDebug_DHCSR_S_SLEEP_Pos 18`

CoreDebug DHCSR: S\_SLEEP Position

Definition at line 1292 of file core\_cm4.h.

## 6.70 Core Definitions

### 6.70.1 Detailed Description

Definitions for base addresses, unions, and structures.

#### Macros

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

### 6.70.2 Macro Definition Documentation

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

Core Debug configuration struct

Definition at line 1389 of file core\_cm4.h.

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

Core Debug Base Address

Definition at line 1377 of file core\_cm4.h.

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

DWT configuration struct

Definition at line 1387 of file core\_cm4.h.



**6.70.2.4 #define DWT\_BASE (0xE0001000UL)**

DWT Base Address

Definition at line 1375 of file core\_cm4.h.

**6.70.2.5 #define ITM ((ITM\_Type \*) ITM\_BASE )**

ITM configuration struct

Definition at line 1386 of file core\_cm4.h.

**6.70.2.6 #define ITM\_BASE (0xE0000000UL)**

ITM Base Address

Definition at line 1374 of file core\_cm4.h.

**6.70.2.7 #define NVIC ((NVIC\_Type \*) NVIC\_BASE )**

NVIC configuration struct

Definition at line 583 of file core\_cm0plus.h.

**6.70.2.8 #define NVIC ((NVIC\_Type \*) NVIC\_BASE )**

NVIC configuration struct

Definition at line 1385 of file core\_cm4.h.

**6.70.2.9 #define NVIC\_BASE (SCS\_BASE + 0x0100UL)**

NVIC Base Address

Definition at line 578 of file core\_cm0plus.h.

**6.70.2.10 #define NVIC\_BASE (SCS\_BASE + 0x0100UL)**

NVIC Base Address

Definition at line 1379 of file core\_cm4.h.

**6.70.2.11 #define SCB ((SCB\_Type \*) SCB\_BASE )**

SCB configuration struct

Definition at line 581 of file core\_cm0plus.h.

**6.70.2.12 #define SCB ((SCB\_Type \*) SCB\_BASE )**

SCB configuration struct

Definition at line 1383 of file core\_cm4.h.

**6.70.2.13 #define SCB\_BASE (SCS\_BASE + 0x0D00UL)**

System Control Block Base Address

Definition at line 579 of file core\_cm0plus.h.

**6.70.2.14 #define SCB\_BASE (SCS\_BASE + 0x0D00UL)**

System Control Block Base Address

Definition at line 1380 of file core\_cm4.h.

**6.70.2.15 #define SCnSCB ((SCnSCB\_Type \*) SCS\_BASE )**

System control Register not in SCB

Definition at line 1382 of file core\_cm4.h.

**6.70.2.16 #define SCS\_BASE (0xE000E000UL)**

System Control Space Base Address

Definition at line 576 of file core\_cm0plus.h.

**6.70.2.17 #define SCS\_BASE (0xE000E000UL)**

System Control Space Base Address

Definition at line 1373 of file core\_cm4.h.

**6.70.2.18 #define SysTick ((SysTick\_Type \*) SysTick\_BASE )**

SysTick configuration struct

Definition at line 582 of file core\_cm0plus.h.

**6.70.2.19 #define SysTick ((SysTick\_Type \*) SysTick\_BASE )**

SysTick configuration struct

Definition at line 1384 of file core\_cm4.h.

**6.70.2.20 #define SysTick\_BASE (SCS\_BASE + 0x0010UL)**

SysTick Base Address

Definition at line 577 of file core\_cm0plus.h.

**6.70.2.21 #define SysTick\_BASE (SCS\_BASE + 0x0010UL)**

SysTick Base Address

Definition at line 1378 of file core\_cm4.h.

**6.70.2.22 #define TPI ((TPI\_Type \*) TPI\_BASE )**

TPI configuration struct

Definition at line 1388 of file core\_cm4.h.

**6.70.2.23 #define TPI\_BASE (0xE0040000UL)**

TPI Base Address

Definition at line 1376 of file core\_cm4.h.

## 6.71 Data Watchpoint and Trace (DWT)

### 6.71.1 Detailed Description

Type definitions for the Data Watchpoint and Trace (DWT)

#### Data Structures

- struct [DWT\\_Type](#)

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

#### Macros

- #define [DWT\\_CTRL\\_NUMCOMP\\_Pos](#) 28
- #define [DWT\\_CTRL\\_NUMCOMP\\_Msk](#) (0xFUL << DWT\_CTRL\_NUMCOMP\_Pos)
- #define [DWT\\_CTRL\\_NOTRCPKT\\_Pos](#) 27
- #define [DWT\\_CTRL\\_NOTRCPKT\\_Msk](#) (0x1UL << DWT\_CTRL\_NOTRCPKT\_Pos)
- #define [DWT\\_CTRL\\_NOEXTTRIG\\_Pos](#) 26
- #define [DWT\\_CTRL\\_NOEXTTRIG\\_Msk](#) (0x1UL << DWT\_CTRL\_NOEXTTRIG\_Pos)
- #define [DWT\\_CTRL\\_NOCYCCNT\\_Pos](#) 25
- #define [DWT\\_CTRL\\_NOCYCCNT\\_Msk](#) (0x1UL << DWT\_CTRL\_NOCYCCNT\_Pos)
- #define [DWT\\_CTRL\\_NOPRFCNT\\_Pos](#) 24
- #define [DWT\\_CTRL\\_NOPRFCNT\\_Msk](#) (0x1UL << DWT\_CTRL\_NOPRFCNT\_Pos)
- #define [DWT\\_CTRL\\_CYCEVTENA\\_Pos](#) 22
- #define [DWT\\_CTRL\\_CYCEVTENA\\_Msk](#) (0x1UL << DWT\_CTRL\_CYCEVTENA\_Pos)
- #define [DWT\\_CTRL\\_FOLDEVTENA\\_Pos](#) 21
- #define [DWT\\_CTRL\\_FOLDEVTENA\\_Msk](#) (0x1UL << DWT\_CTRL\_FOLDEVTENA\_Pos)
- #define [DWT\\_CTRL\\_LSUEVTENA\\_Pos](#) 20
- #define [DWT\\_CTRL\\_LSUEVTENA\\_Msk](#) (0x1UL << DWT\_CTRL\_LSUEVTENA\_Pos)
- #define [DWT\\_CTRL\\_SLEEPEVTENA\\_Pos](#) 19
- #define [DWT\\_CTRL\\_SLEEPEVTENA\\_Msk](#) (0x1UL << DWT\_CTRL\_SLEEPEVTENA\_Pos)
- #define [DWT\\_CTRL\\_EXCEVTENA\\_Pos](#) 18
- #define [DWT\\_CTRL\\_EXCEVTENA\\_Msk](#) (0x1UL << DWT\_CTRL\_EXCEVTENA\_Pos)
- #define [DWT\\_CTRL\\_CPIEVTENA\\_Pos](#) 17
- #define [DWT\\_CTRL\\_CPIEVTENA\\_Msk](#) (0x1UL << DWT\_CTRL\_CPIEVTENA\_Pos)
- #define [DWT\\_CTRL\\_EXCTRCENA\\_Pos](#) 16
- #define [DWT\\_CTRL\\_EXCTRCENA\\_Msk](#) (0x1UL << DWT\_CTRL\_EXCTRCENA\_Pos)
- #define [DWT\\_CTRL\\_PCSAMPLENA\\_Pos](#) 12
- #define [DWT\\_CTRL\\_PCSAMPLENA\\_Msk](#) (0x1UL << DWT\_CTRL\_PCSAMPLENA\_Pos)
- #define [DWT\\_CTRL\\_SYNCTAP\\_Pos](#) 10
- #define [DWT\\_CTRL\\_SYNCTAP\\_Msk](#) (0x3UL << DWT\_CTRL\_SYNCTAP\_Pos)
- #define [DWT\\_CTRL\\_CYCTAP\\_Pos](#) 9
- #define [DWT\\_CTRL\\_CYCTAP\\_Msk](#) (0x1UL << DWT\_CTRL\_CYCTAP\_Pos)
- #define [DWT\\_CTRL\\_POSTINIT\\_Pos](#) 5
- #define [DWT\\_CTRL\\_POSTINIT\\_Msk](#) (0xFUL << DWT\_CTRL\_POSTINIT\_Pos)
- #define [DWT\\_CTRL\\_POSTPRESET\\_Pos](#) 1
- #define [DWT\\_CTRL\\_POSTPRESET\\_Msk](#) (0xFUL << DWT\_CTRL\_POSTPRESET\_Pos)
- #define [DWT\\_CTRL\\_CYCCNTENA\\_Pos](#) 0
- #define [DWT\\_CTRL\\_CYCCNTENA\\_Msk](#) (0x1UL << DWT\_CTRL\_CYCCNTENA\_Pos)
- #define [DWT\\_CPICNT\\_CPICNT\\_Pos](#) 0
- #define [DWT\\_CPICNT\\_CPICNT\\_Msk](#) (0xFFUL << DWT\_CPICNT\_CPICNT\_Pos)
- #define [DWT\\_EXCCNT\\_EXCCNT\\_Pos](#) 0
- #define [DWT\\_EXCCNT\\_EXCCNT\\_Msk](#) (0xFFUL << DWT\_EXCCNT\_EXCCNT\_Pos)

- `#define DWT_SLEEPCNT_SLEEPCNT_Pos 0`
- `#define DWT_SLEEPCNT_SLEEPCNT_Msk (0xFFUL << DWT_SLEEPCNT_SLEEPCNT_Pos)`
- `#define DWT_LSUCNT_LSUCNT_Pos 0`
- `#define DWT_LSUCNT_LSUCNT_Msk (0xFFUL << DWT_LSUCNT_LSUCNT_Pos)`
- `#define DWT_FOLDCNT_FOLDCNT_Pos 0`
- `#define DWT_FOLDCNT_FOLDCNT_Msk (0xFFUL << DWT_FOLDCNT_FOLDCNT_Pos)`
- `#define DWT_MASK_MASK_Pos 0`
- `#define DWT_MASK_MASK_Msk (0x1FUL << DWT_MASK_MASK_Pos)`
- `#define DWT_FUNCTION_MATCHED_Pos 24`
- `#define DWT_FUNCTION_MATCHED_Msk (0x1UL << DWT_FUNCTION_MATCHED_Pos)`
- `#define DWT_FUNCTION_DATAVADDR1_Pos 16`
- `#define DWT_FUNCTION_DATAVADDR1_Msk (0xFUL << DWT_FUNCTION_DATAVADDR1_Pos)`
- `#define DWT_FUNCTION_DATAVADDR0_Pos 12`
- `#define DWT_FUNCTION_DATAVADDR0_Msk (0xFUL << DWT_FUNCTION_DATAVADDR0_Pos)`
- `#define DWT_FUNCTION_DATAVSIZE_Pos 10`
- `#define DWT_FUNCTION_DATAVSIZE_Msk (0x3UL << DWT_FUNCTION_DATAVSIZE_Pos)`
- `#define DWT_FUNCTION_LNK1ENA_Pos 9`
- `#define DWT_FUNCTION_LNK1ENA_Msk (0x1UL << DWT_FUNCTION_LNK1ENA_Pos)`
- `#define DWT_FUNCTION_DATAVMATCH_Pos 8`
- `#define DWT_FUNCTION_DATAVMATCH_Msk (0x1UL << DWT_FUNCTION_DATAVMATCH_Pos)`
- `#define DWT_FUNCTION_CYCMATCH_Pos 7`
- `#define DWT_FUNCTION_CYCMATCH_Msk (0x1UL << DWT_FUNCTION_CYCMATCH_Pos)`
- `#define DWT_FUNCTION_EMITRANGE_Pos 5`
- `#define DWT_FUNCTION_EMITRANGE_Msk (0x1UL << DWT_FUNCTION_EMITRANGE_Pos)`
- `#define DWT_FUNCTION_FUNCTION_Pos 0`
- `#define DWT_FUNCTION_FUNCTION_Msk (0xFUL << DWT_FUNCTION_FUNCTION_Pos)`

## 6.71.2 Macro Definition Documentation

### 6.71.2.1 `#define DWT_CPICNT_CPICNT_Msk (0xFFUL << DWT_CPICNT_CPICNT_Pos)`

DWT CPICNT: CPICNT Mask

Definition at line 858 of file core\_cm4.h.

### 6.71.2.2 `#define DWT_CPICNT_CPICNT_Pos 0`

DWT CPICNT: CPICNT Position

Definition at line 857 of file core\_cm4.h.

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

DWT CTRL: CPIEVTENA Mask

Definition at line 833 of file core\_cm4.h.

### 6.71.2.4 `#define DWT_CTRL_CPIEVTENA_Pos 17`

DWT CTRL: CPIEVTENA Position

Definition at line 832 of file core\_cm4.h.

6.71.2.5 `#define DWT_CTRL_CYCCNTENA_Msk (0x1UL << DWT_CTRL_CYCCNTENA_Pos)`

DWT CTRL: CYCCNTENA Mask

Definition at line 854 of file core\_cm4.h.

6.71.2.6 `#define DWT_CTRL_CYCCNTENA_Pos 0`

DWT CTRL: CYCCNTENA Position

Definition at line 853 of file core\_cm4.h.

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

DWT CTRL: CYCEVTENA Mask

Definition at line 818 of file core\_cm4.h.

6.71.2.8 `#define DWT_CTRL_CYCEVTENA_Pos 22`

DWT CTRL: CYCEVTENA Position

Definition at line 817 of file core\_cm4.h.

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

DWT CTRL: CYCTAP Mask

Definition at line 845 of file core\_cm4.h.

6.71.2.10 `#define DWT_CTRL_CYCTAP_Pos 9`

DWT CTRL: CYCTAP Position

Definition at line 844 of file core\_cm4.h.

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

DWT CTRL: EXCEVTENA Mask

Definition at line 830 of file core\_cm4.h.

6.71.2.12 `#define DWT_CTRL_EXCEVTENA_Pos 18`

DWT CTRL: EXCEVTENA Position

Definition at line 829 of file core\_cm4.h.

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

DWT CTRL: EXCTRCENA Mask

Definition at line 836 of file core\_cm4.h.

6.71.2.14 `#define DWT_CTRL_EXCTRCENA_Pos 16`

DWT CTRL: EXCTRCENA Position

Definition at line 835 of file core\_cm4.h.

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

DWT CTRL: FOLDEVTENA Mask

Definition at line 821 of file core\_cm4.h.

6.71.2.16 `#define DWT_CTRL_FOLDEVTENA_Pos 21`

DWT CTRL: FOLDEVTENA Position

Definition at line 820 of file core\_cm4.h.

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

DWT CTRL: LSUEVTENA Mask

Definition at line 824 of file core\_cm4.h.

6.71.2.18 `#define DWT_CTRL_LSUEVTENA_Pos 20`

DWT CTRL: LSUEVTENA Position

Definition at line 823 of file core\_cm4.h.

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

DWT CTRL: NOCYCCNT Mask

Definition at line 812 of file core\_cm4.h.

6.71.2.20 `#define DWT_CTRL_NOCYCCNT_Pos 25`

DWT CTRL: NOCYCCNT Position

Definition at line 811 of file core\_cm4.h.

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

DWT CTRL: NOEXTTRIG Mask

Definition at line 809 of file core\_cm4.h.

6.71.2.22 `#define DWT_CTRL_NOEXTTRIG_Pos 26`

DWT CTRL: NOEXTTRIG Position

Definition at line 808 of file core\_cm4.h.

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

DWT CTRL: NOPRFCNT Mask

Definition at line 815 of file core\_cm4.h.

6.71.2.24 `#define DWT_CTRL_NOPRFCNT_Pos 24`

DWT CTRL: NOPRFCNT Position

Definition at line 814 of file core\_cm4.h.

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

DWT CTRL: NOTRCPKT Mask

Definition at line 806 of file core\_cm4.h.

6.71.2.26 `#define DWT_CTRL_NOTRCPKT_Pos 27`

DWT CTRL: NOTRCPKT Position

Definition at line 805 of file core\_cm4.h.

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

DWT CTRL: NUMCOMP Mask

Definition at line 803 of file core\_cm4.h.

6.71.2.28 `#define DWT_CTRL_NUMCOMP_Pos 28`

DWT CTRL: NUMCOMP Position

Definition at line 802 of file core\_cm4.h.

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

DWT CTRL: PCSAMPLENA Mask

Definition at line 839 of file core\_cm4.h.

6.71.2.30 `#define DWT_CTRL_PCSAMPLENA_Pos 12`

DWT CTRL: PCSAMPLENA Position

Definition at line 838 of file core\_cm4.h.

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

DWT CTRL: POSTINIT Mask

Definition at line 848 of file core\_cm4.h.



**6.71.2.32 #define DWT\_CTRL\_POSTINIT\_Pos 5**

DWT CTRL: POSTINIT Position

Definition at line 847 of file core\_cm4.h.

**6.71.2.33 #define DWT\_CTRL\_POSTPRESET\_Msk (0xFUL << DWT\_CTRL\_POSTPRESET\_Pos)**

DWT CTRL: POSTPRESET Mask

Definition at line 851 of file core\_cm4.h.

**6.71.2.34 #define DWT\_CTRL\_POSTPRESET\_Pos 1**

DWT CTRL: POSTPRESET Position

Definition at line 850 of file core\_cm4.h.

**6.71.2.35 #define DWT\_CTRL\_SLEEPEVTENA\_Msk (0x1UL << DWT\_CTRL\_SLEEPEVTENA\_Pos)**

DWT CTRL: SLEEPEVTENA Mask

Definition at line 827 of file core\_cm4.h.

**6.71.2.36 #define DWT\_CTRL\_SLEEPEVTENA\_Pos 19**

DWT CTRL: SLEEPEVTENA Position

Definition at line 826 of file core\_cm4.h.

**6.71.2.37 #define DWT\_CTRL\_SYNCTAP\_Msk (0x3UL << DWT\_CTRL\_SYNCTAP\_Pos)**

DWT CTRL: SYNCTAP Mask

Definition at line 842 of file core\_cm4.h.

**6.71.2.38 #define DWT\_CTRL\_SYNCTAP\_Pos 10**

DWT CTRL: SYNCTAP Position

Definition at line 841 of file core\_cm4.h.

**6.71.2.39 #define DWT\_EXCCNT\_EXCCNT\_Msk (0xFFUL << DWT\_EXCCNT\_EXCCNT\_Pos)**

DWT EXCCNT: EXCCNT Mask

Definition at line 862 of file core\_cm4.h.

**6.71.2.40 #define DWT\_EXCCNT\_EXCCNT\_Pos 0**

DWT EXCCNT: EXCCNT Position

Definition at line 861 of file core\_cm4.h.

6.71.2.41 **#define DWT\_FOLDCNT\_FOLDCNT\_Msk** (0xFFUL << DWT\_FOLDCNT\_FOLDCNT\_Pos)

DWT FOLDCNT: FOLDCNT Mask

Definition at line 874 of file core\_cm4.h.

6.71.2.42 **#define DWT\_FOLDCNT\_FOLDCNT\_Pos** 0

DWT FOLDCNT: FOLDCNT Position

Definition at line 873 of file core\_cm4.h.

6.71.2.43 **#define DWT\_FUNCTION\_CYCMATCH\_Msk** (0x1UL << DWT\_FUNCTION\_CYCMATCH\_Pos)

DWT FUNCTION: CYCMATCH Mask

Definition at line 900 of file core\_cm4.h.

6.71.2.44 **#define DWT\_FUNCTION\_CYCMATCH\_Pos** 7

DWT FUNCTION: CYCMATCH Position

Definition at line 899 of file core\_cm4.h.

6.71.2.45 **#define DWT\_FUNCTION\_DATAVADDR0\_Msk** (0xFUL << DWT\_FUNCTION\_DATAVADDR0\_Pos)

DWT FUNCTION: DATAVADDR0 Mask

Definition at line 888 of file core\_cm4.h.

6.71.2.46 **#define DWT\_FUNCTION\_DATAVADDR0\_Pos** 12

DWT FUNCTION: DATAVADDR0 Position

Definition at line 887 of file core\_cm4.h.

6.71.2.47 **#define DWT\_FUNCTION\_DATAVADDR1\_Msk** (0xFUL << DWT\_FUNCTION\_DATAVADDR1\_Pos)

DWT FUNCTION: DATAVADDR1 Mask

Definition at line 885 of file core\_cm4.h.

6.71.2.48 **#define DWT\_FUNCTION\_DATAVADDR1\_Pos** 16

DWT FUNCTION: DATAVADDR1 Position

Definition at line 884 of file core\_cm4.h.

6.71.2.49 **#define DWT\_FUNCTION\_DATAVMATCH\_Msk** (0x1UL << DWT\_FUNCTION\_DATAVMATCH\_Pos)

DWT FUNCTION: DATAVMATCH Mask

Definition at line 897 of file core\_cm4.h.

**6.71.2.50 #define DWT\_FUNCTION\_DATAVMATCH\_Pos 8**

DWT FUNCTION: DATAVMATCH Position

Definition at line 896 of file core\_cm4.h.

**6.71.2.51 #define DWT\_FUNCTION\_DATAVSIZE\_Msk (0x3UL << DWT\_FUNCTION\_DATAVSIZE\_Pos)**

DWT FUNCTION: DATAVSIZE Mask

Definition at line 891 of file core\_cm4.h.

**6.71.2.52 #define DWT\_FUNCTION\_DATAVSIZE\_Pos 10**

DWT FUNCTION: DATAVSIZE Position

Definition at line 890 of file core\_cm4.h.

**6.71.2.53 #define DWT\_FUNCTION\_EMITRANGE\_Msk (0x1UL << DWT\_FUNCTION\_EMITRANGE\_Pos)**

DWT FUNCTION: EMITRANGE Mask

Definition at line 903 of file core\_cm4.h.

**6.71.2.54 #define DWT\_FUNCTION\_EMITRANGE\_Pos 5**

DWT FUNCTION: EMITRANGE Position

Definition at line 902 of file core\_cm4.h.

**6.71.2.55 #define DWT\_FUNCTION\_FUNCTION\_Msk (0xFUL << DWT\_FUNCTION\_FUNCTION\_Pos)**

DWT FUNCTION: FUNCTION Mask

Definition at line 906 of file core\_cm4.h.

**6.71.2.56 #define DWT\_FUNCTION\_FUNCTION\_Pos 0**

DWT FUNCTION: FUNCTION Position

Definition at line 905 of file core\_cm4.h.

**6.71.2.57 #define DWT\_FUNCTION\_LNK1ENA\_Msk (0x1UL << DWT\_FUNCTION\_LNK1ENA\_Pos)**

DWT FUNCTION: LNK1ENA Mask

Definition at line 894 of file core\_cm4.h.

**6.71.2.58 #define DWT\_FUNCTION\_LNK1ENA\_Pos 9**

DWT FUNCTION: LNK1ENA Position

Definition at line 893 of file core\_cm4.h.

6.71.2.59 **#define DWT\_FUNCTION\_MATCHED\_Msk** (0x1UL << DWT\_FUNCTION\_MATCHED\_Pos)

DWT FUNCTION: MATCHED Mask

Definition at line 882 of file core\_cm4.h.

6.71.2.60 **#define DWT\_FUNCTION\_MATCHED\_Pos** 24

DWT FUNCTION: MATCHED Position

Definition at line 881 of file core\_cm4.h.

6.71.2.61 **#define DWT\_LSUCNT\_LSUCNT\_Msk** (0xFFUL << DWT\_LSUCNT\_LSUCNT\_Pos)

DWT LSUCNT: LSUCNT Mask

Definition at line 870 of file core\_cm4.h.

6.71.2.62 **#define DWT\_LSUCNT\_LSUCNT\_Pos** 0

DWT LSUCNT: LSUCNT Position

Definition at line 869 of file core\_cm4.h.

6.71.2.63 **#define DWT\_MASK\_MASK\_Msk** (0x1FUL << DWT\_MASK\_MASK\_Pos)

DWT MASK: MASK Mask

Definition at line 878 of file core\_cm4.h.

6.71.2.64 **#define DWT\_MASK\_MASK\_Pos** 0

DWT MASK: MASK Position

Definition at line 877 of file core\_cm4.h.

6.71.2.65 **#define DWT\_SLEEPCNT\_SLEEPCNT\_Msk** (0xFFUL << DWT\_SLEEPCNT\_SLEEPCNT\_Pos)

DWT SLEEPCNT: SLEEPCNT Mask

Definition at line 866 of file core\_cm4.h.

6.71.2.66 **#define DWT\_SLEEPCNT\_SLEEPCNT\_Pos** 0

DWT SLEEPCNT: SLEEPCNT Position

Definition at line 865 of file core\_cm4.h.

## 6.72 Defines and Type Definitions

### 6.72.1 Detailed Description

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

#### Modules

- [Core Debug Registers \(CoreDebug\)](#)  
*Cortex-M0+ Core Debug Registers (DCB registers, SHCSR, and DFSR) are only accessible over DAP and not via processor. Therefore they are not covered by the Cortex-M0 header file.*
- [Core Definitions](#)  
*Definitions for base addresses, unions, and structures.*
- [Data Watchpoint and Trace \(DWT\)](#)  
*Type definitions for the Data Watchpoint and Trace (DWT)*
- [Instrumentation Trace Macrocell \(ITM\)](#)  
*Type definitions for the Instrumentation Trace Macrocell (ITM)*
- [Nested Vectored Interrupt Controller \(NVIC\)](#)  
*Type definitions for the NVIC Registers.*
- [Status and Control Registers](#)  
*Core Register type definitions.*
- [System Control Block \(SCB\)](#)  
*Type definitions for the System Control Block Registers.*
- [System Controls not in SCB \(SCnSCB\)](#)  
*Type definitions for the System Control and ID Register not in the SCB.*
- [System Tick Timer \(SysTick\)](#)  
*Type definitions for the System Timer Registers.*
- [Trace Port Interface \(TPI\)](#)  
*Type definitions for the Trace Port Interface (TPI)*

## 6.73 Functions and Instructions Reference

### 6.73.1 Detailed Description

#### Modules

- [CMSIS Core Register Access Functions](#)
- [ITM Functions](#)  
*Functions that access the ITM debug interface.*
- [NVIC Functions](#)  
*Functions that manage interrupts and exceptions via the NVIC.*
- [SysTick Functions](#)  
*Functions that configure the System.*

## 6.74 IAR EWARM support in LPCOpen

IAR Embedded Workbench for ARM

## 6.75 ITM Functions

### 6.75.1 Detailed Description

Functions that access the ITM debug interface.

#### Macros

- `#define ITM_RXBUFFER_EMPTY 0x5AA55AA5`

#### Functions

- `__STATIC_INLINE uint32_t ITM_SendChar (uint32_t ch)`  
*ITM Send Character.*
- `__STATIC_INLINE int32_t ITM_ReceiveChar (void)`  
*ITM Receive Character.*
- `__STATIC_INLINE int32_t ITM_CheckChar (void)`  
*ITM Check Character.*

#### Variables

- `volatile int32_t ITM_RxBuffer`

### 6.75.2 Macro Definition Documentation

#### 6.75.2.1 `#define ITM_RXBUFFER_EMPTY 0x5AA55AA5`

Value identifying `ITM_RxBuffer` is ready for next character.

Definition at line 1704 of file `core_cm4.h`.

### 6.75.3 Function Documentation

#### 6.75.3.1 `__STATIC_INLINE int32_t ITM_CheckChar ( void )`

ITM Check Character.

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

#### Returns

- 0 No character available.
- 1 Character available.

Definition at line 1755 of file `core_cm4.h`.

#### 6.75.3.2 `__STATIC_INLINE int32_t ITM_ReceiveChar ( void )`

ITM Receive Character.

The function inputs a character via the external variable `ITM_RxBuffer`.



**Returns**

Received character.  
-1 No character pending.

Definition at line 1736 of file core\_cm4.h.

**6.75.3.3   \_\_STATIC\_INLINE uint32\_t ITM\_SendChar ( uint32\_t *ch* )**

ITM Send Character.

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

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

**Parameters**

<i>in</i>	<i>ch</i>	Character to transmit.
-----------	-----------	------------------------

**Returns**

Character to transmit.

Definition at line 1717 of file core\_cm4.h.

**6.75.4   Variable Documentation****6.75.4.1   volatile int32\_t ITM\_RxBuffer**

External variable to receive characters.

## 6.76 Instrumentation Trace Macrocell (ITM)

### 6.76.1 Detailed Description

Type definitions for the Instrumentation Trace Macrocell (ITM)

#### Data Structures

- struct [ITM\\_Type](#)

*Structure type to access the Instrumentation Trace Macrocell Register (ITM).*

#### Macros

- #define [ITM\\_TPR\\_PRIVMASK\\_Pos](#) 0
- #define [ITM\\_TPR\\_PRIVMASK\\_Msk](#) (0xFUL << ITM\_TPR\_PRIVMASK\_Pos)
- #define [ITM\\_TCR\\_BUSY\\_Pos](#) 23
- #define [ITM\\_TCR\\_BUSY\\_Msk](#) (1UL << ITM\_TCR\_BUSY\_Pos)
- #define [ITM\\_TCR\\_TraceBusID\\_Pos](#) 16
- #define [ITM\\_TCR\\_TraceBusID\\_Msk](#) (0x7FUL << ITM\_TCR\_TraceBusID\_Pos)
- #define [ITM\\_TCR\\_GTSFREQ\\_Pos](#) 10
- #define [ITM\\_TCR\\_GTSFREQ\\_Msk](#) (3UL << ITM\_TCR\_GTSFREQ\_Pos)
- #define [ITM\\_TCR\\_TSPrescale\\_Pos](#) 8
- #define [ITM\\_TCR\\_TSPrescale\\_Msk](#) (3UL << ITM\_TCR\_TSPrescale\_Pos)
- #define [ITM\\_TCR\\_SWOENA\\_Pos](#) 4
- #define [ITM\\_TCR\\_SWOENA\\_Msk](#) (1UL << ITM\_TCR\_SWOENA\_Pos)
- #define [ITM\\_TCR\\_DWTENA\\_Pos](#) 3
- #define [ITM\\_TCR\\_DWTENA\\_Msk](#) (1UL << ITM\_TCR\_DWTENA\_Pos)
- #define [ITM\\_TCR\\_SYNCENA\\_Pos](#) 2
- #define [ITM\\_TCR\\_SYNCENA\\_Msk](#) (1UL << ITM\_TCR\_SYNCENA\_Pos)
- #define [ITM\\_TCR\\_TSENA\\_Pos](#) 1
- #define [ITM\\_TCR\\_TSENA\\_Msk](#) (1UL << ITM\_TCR\_TSENA\_Pos)
- #define [ITM\\_TCR\\_ITMENA\\_Pos](#) 0
- #define [ITM\\_TCR\\_ITMENA\\_Msk](#) (1UL << ITM\_TCR\_ITMENA\_Pos)
- #define [ITM\\_IWR\\_ATVALIDM\\_Pos](#) 0
- #define [ITM\\_IWR\\_ATVALIDM\\_Msk](#) (1UL << ITM\_IWR\_ATVALIDM\_Pos)
- #define [ITM\\_IRR\\_ATREADYM\\_Pos](#) 0
- #define [ITM\\_IRR\\_ATREADYM\\_Msk](#) (1UL << ITM\_IRR\_ATREADYM\_Pos)
- #define [ITM\\_IMCR\\_INTEGRATION\\_Pos](#) 0
- #define [ITM\\_IMCR\\_INTEGRATION\\_Msk](#) (1UL << ITM\_IMCR\_INTEGRATION\_Pos)
- #define [ITM\\_LSR\\_ByteAcc\\_Pos](#) 2
- #define [ITM\\_LSR\\_ByteAcc\\_Msk](#) (1UL << ITM\_LSR\_ByteAcc\_Pos)
- #define [ITM\\_LSR\\_Access\\_Pos](#) 1
- #define [ITM\\_LSR\\_Access\\_Msk](#) (1UL << ITM\_LSR\_Access\_Pos)
- #define [ITM\\_LSR\\_Present\\_Pos](#) 0
- #define [ITM\\_LSR\\_Present\\_Msk](#) (1UL << ITM\_LSR\_Present\_Pos)

### 6.76.2 Macro Definition Documentation

#### 6.76.2.1 #define ITM\_IMCR\_INTEGRATION\_Msk (1UL << ITM\_IMCR\_INTEGRATION\_Pos)

ITM IMCR: INTEGRATION Mask

Definition at line 751 of file core\_cm4.h.

**6.76.2.2 #define ITM\_IMCR\_INTEGRATION\_Pos 0**

ITM IMCR: INTEGRATION Position

Definition at line 750 of file core\_cm4.h.

**6.76.2.3 #define ITM\_IRR\_ATREADYM\_Msk (1UL << ITM\_IRR\_ATREADYM\_Pos)**

ITM IRR: ATREADYM Mask

Definition at line 747 of file core\_cm4.h.

**6.76.2.4 #define ITM\_IRR\_ATREADYM\_Pos 0**

ITM IRR: ATREADYM Position

Definition at line 746 of file core\_cm4.h.

**6.76.2.5 #define ITM\_IWR\_ATVALIDM\_Msk (1UL << ITM\_IWR\_ATVALIDM\_Pos)**

ITM IWR: ATVALIDM Mask

Definition at line 743 of file core\_cm4.h.

**6.76.2.6 #define ITM\_IWR\_ATVALIDM\_Pos 0**

ITM IWR: ATVALIDM Position

Definition at line 742 of file core\_cm4.h.

**6.76.2.7 #define ITM\_LSR\_Access\_Msk (1UL << ITM\_LSR\_Access\_Pos)**

ITM LSR: Access Mask

Definition at line 758 of file core\_cm4.h.

**6.76.2.8 #define ITM\_LSR\_Access\_Pos 1**

ITM LSR: Access Position

Definition at line 757 of file core\_cm4.h.

**6.76.2.9 #define ITM\_LSR\_ByteAcc\_Msk (1UL << ITM\_LSR\_ByteAcc\_Pos)**

ITM LSR: ByteAcc Mask

Definition at line 755 of file core\_cm4.h.

**6.76.2.10 #define ITM\_LSR\_ByteAcc\_Pos 2**

ITM LSR: ByteAcc Position

Definition at line 754 of file core\_cm4.h.

**6.76.2.11 #define ITM\_LSR\_Present\_Msk (1UL << ITM\_LSR\_Present\_Pos)**

ITM LSR: Present Mask

Definition at line 761 of file core\_cm4.h.

**6.76.2.12 #define ITM\_LSR\_Present\_Pos 0**

ITM LSR: Present Position

Definition at line 760 of file core\_cm4.h.

**6.76.2.13 #define ITM\_TCR\_BUSY\_Msk (1UL << ITM\_TCR\_BUSY\_Pos)**

ITM TCR: BUSY Mask

Definition at line 715 of file core\_cm4.h.

**6.76.2.14 #define ITM\_TCR\_BUSY\_Pos 23**

ITM TCR: BUSY Position

Definition at line 714 of file core\_cm4.h.

**6.76.2.15 #define ITM\_TCR\_DWTENA\_Msk (1UL << ITM\_TCR\_DWTENA\_Pos)**

ITM TCR: DWTENA Mask

Definition at line 730 of file core\_cm4.h.

**6.76.2.16 #define ITM\_TCR\_DWTENA\_Pos 3**

ITM TCR: DWTENA Position

Definition at line 729 of file core\_cm4.h.

**6.76.2.17 #define ITM\_TCR\_GTSFREQ\_Msk (3UL << ITM\_TCR\_GTSFREQ\_Pos)**

ITM TCR: Global timestamp frequency Mask

Definition at line 721 of file core\_cm4.h.

**6.76.2.18 #define ITM\_TCR\_GTSFREQ\_Pos 10**

ITM TCR: Global timestamp frequency Position

Definition at line 720 of file core\_cm4.h.

**6.76.2.19 #define ITM\_TCR\_ITMENA\_Msk (1UL << ITM\_TCR\_ITMENA\_Pos)**

ITM TCR: ITM Enable bit Mask

Definition at line 739 of file core\_cm4.h.

6.76.2.20 `#define ITM_TCR_ITMENA_Pos 0`

ITM TCR: ITM Enable bit Position

Definition at line 738 of file core\_cm4.h.

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

ITM TCR: SWOENA Mask

Definition at line 727 of file core\_cm4.h.

6.76.2.22 `#define ITM_TCR_SWOENA_Pos 4`

ITM TCR: SWOENA Position

Definition at line 726 of file core\_cm4.h.

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

ITM TCR: SYNCENA Mask

Definition at line 733 of file core\_cm4.h.

6.76.2.24 `#define ITM_TCR_SYNCENA_Pos 2`

ITM TCR: SYNCENA Position

Definition at line 732 of file core\_cm4.h.

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

ITM TCR: ATBID Mask

Definition at line 718 of file core\_cm4.h.

6.76.2.26 `#define ITM_TCR_TraceBusID_Pos 16`

ITM TCR: ATBID Position

Definition at line 717 of file core\_cm4.h.

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

ITM TCR: TSENA Mask

Definition at line 736 of file core\_cm4.h.

6.76.2.28 `#define ITM_TCR_TSENA_Pos 1`

ITM TCR: TSENA Position

Definition at line 735 of file core\_cm4.h.

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

ITM TCR: TSPrescale Mask

Definition at line 724 of file core\_cm4.h.

6.76.2.30 `#define ITM_TCR_TSPrescale_Pos 8`

ITM TCR: TSPrescale Position

Definition at line 723 of file core\_cm4.h.

6.76.2.31 `#define ITM_TPR_PRIVMASK_Msk (0xFUL << ITM_TPR_PRIVMASK_Pos)`

ITM TPR: PRIVMASK Mask

Definition at line 711 of file core\_cm4.h.

6.76.2.32 `#define ITM_TPR_PRIVMASK_Pos 0`

ITM TPR: PRIVMASK Position

Definition at line 710 of file core\_cm4.h.

## 6.77 Keil uVision support in LPCOpen

MDK-ARM Microcontroller Development Kit (uVision4)

## 6.78 LPC Public Macros

### 6.78.1 Detailed Description

#### Macros

- `#define _BIT(n) (1 << (n))`
- `#define _SBF(f, v) ((v) << (f))`
- `#define _BITMASK(field_width) ( _BIT(field_width) - 1)`
- `#define NULL ((void *) 0)`
- `#define NELEMENTS(array) (sizeof(array) / sizeof(array[0]))`
- `#define STATIC static`
- `#define EXTERN extern`
- `#define MAX(a, b) (((a) > (b)) ? (a) : (b))`
- `#define MIN(a, b) (((a) < (b)) ? (a) : (b))`

### 6.78.2 Macro Definition Documentation

#### 6.78.2.1 `#define _BIT( n )(1 << (n))`

Definition at line 104 of file `lpc_types.h`.

#### 6.78.2.2 `#define _BITMASK( field_width )( _BIT(field_width) - 1)`

Definition at line 129 of file `lpc_types.h`.

#### 6.78.2.3 `#define _SBF( f, v )((v) << (f))`

Definition at line 112 of file `lpc_types.h`.

#### 6.78.2.4 `#define EXTERN extern`

Definition at line 142 of file `lpc_types.h`.

#### 6.78.2.5 `#define MAX( a, b )(((a) > (b)) ? (a) : (b))`

Definition at line 145 of file `lpc_types.h`.

#### 6.78.2.6 `#define MIN( a, b )(((a) < (b)) ? (a) : (b))`

Definition at line 148 of file `lpc_types.h`.

#### 6.78.2.7 `#define NELEMENTS( array )(sizeof(array) / sizeof(array[0]))`

Definition at line 137 of file `lpc_types.h`.

#### 6.78.2.8 `#define NULL ((void *) 0)`

Definition at line 133 of file `lpc_types.h`.



#### 6.78.2.9 #define STATIC static

Definition at line 140 of file lpc\_types.h.

## 6.79 LPC Public Types

### 6.79.1 Detailed Description

#### Macros

- `#define PARAM_SETSTATE(State) ((State == RESET) || (State == SET))`
- `#define PARAM_FUNCTIONALSTATE(State) ((State == DISABLE) || (State == ENABLE))`
- `#define INLINE inline`
- `#define ALIGN(x) __attribute__((aligned(x)))`
- `#define WEAK __attribute__((weak))`

#### Typedefs

- `typedef enum FlagStatus IntStatus`
- `typedef enum FlagStatus SetState`
- `typedef void(* PFV) ()`
- `typedef int32_t(* PFI) ()`
- `typedef char CHAR`
- `typedef uint8_t UNS_8`
- `typedef int8_t INT_8`
- `typedef uint16_t UNS_16`
- `typedef int16_t INT_16`
- `typedef uint32_t UNS_32`
- `typedef int32_t INT_32`
- `typedef int64_t INT_64`
- `typedef uint64_t UNS_64`
- `typedef bool BOOL_32`
- `typedef bool BOOL_16`
- `typedef bool BOOL_8`

#### Enumerations

- `enum Bool { FALSE = 0, TRUE = !FALSE }`  
*Boolean Type definition.*
- `enum FlagStatus { RESET = 0, SET = !RESET }`  
*Boolean Type definition.*
- `enum FunctionalState { DISABLE = 0, ENABLE = !DISABLE }`  
*Functional State Definition.*
- `enum Status { ERROR = 0, SUCCESS = !ERROR }`
- `enum TRANSFER_BLOCK_T { NONE_BLOCKING = 0, BLOCKING }`

### 6.79.2 Macro Definition Documentation

#### 6.79.2.1 `#define ALIGN( x ) __attribute__((aligned(x)))`

Definition at line 213 of file `lpc_types.h`.

#### 6.79.2.2 `#define INLINE inline`

Definition at line 205 of file `lpc_types.h`.

6.79.2.3 `#define PARAM_FUNCTIONALSTATE( State ) ((State == DISABLE) || (State == ENABLE))`

Definition at line 69 of file `lpc_types.h`.

6.79.2.4 `#define PARAM_SETSTATE( State ) ((State == RESET) || (State == SET))`

Definition at line 63 of file `lpc_types.h`.

6.79.2.5 `#define WEAK __attribute__((weak))`

Definition at line 214 of file `lpc_types.h`.

### 6.79.3 Typedef Documentation

6.79.3.1 `typedef bool BOOL_16`

16 bit boolean type

Definition at line 196 of file `lpc_types.h`.

6.79.3.2 `typedef bool BOOL_32`

32 bit boolean type

Definition at line 193 of file `lpc_types.h`.

6.79.3.3 `typedef bool BOOL_8`

8 bit boolean type

Definition at line 199 of file `lpc_types.h`.

6.79.3.4 `typedef char CHAR`

LPC type for character type

Definition at line 161 of file `lpc_types.h`.

6.79.3.5 `typedef int16_t INT_16`

LPC type for 16 bit signed value

Definition at line 173 of file `lpc_types.h`.

6.79.3.6 `typedef int32_t INT_32`

LPC type for 32 bit signed value

Definition at line 179 of file `lpc_types.h`.

6.79.3.7 `typedef int64_t INT_64`

LPC type for 64 bit signed value

Definition at line 182 of file `lpc_types.h`.

#### 6.79.3.8 typedef int8\_t INT\_8

LPC type for 8 bit signed value

Definition at line 167 of file lpc\_types.h.

#### 6.79.3.9 typedef enum FlagStatus IntStatus

#### 6.79.3.10 typedef int32\_t(\* PFI) ()

Pointer to Function returning int32\_t (any number of parameters)

Definition at line 88 of file lpc\_types.h.

#### 6.79.3.11 typedef void(\* PFV) ()

Pointer to Function returning Void (any number of parameters)

Definition at line 85 of file lpc\_types.h.

#### 6.79.3.12 typedef enum FlagStatus SetState

#### 6.79.3.13 typedef uint16\_t UNS\_16

LPC type for 16 bit unsigned value

Definition at line 170 of file lpc\_types.h.

#### 6.79.3.14 typedef uint32\_t UNS\_32

LPC type for 32 bit unsigned value

Definition at line 176 of file lpc\_types.h.

#### 6.79.3.15 typedef uint64\_t UNS\_64

LPC type for 64 bit unsigned value

Definition at line 185 of file lpc\_types.h.

#### 6.79.3.16 typedef uint8\_t UNS\_8

LPC type for 8 bit unsigned value

Definition at line 164 of file lpc\_types.h.

### 6.79.4 Enumeration Type Documentation

#### 6.79.4.1 enum Bool

Boolean Type definition.

Enumerator

***FALSE***

***TRUE***

Definition at line 50 of file lpc\_types.h.

#### 6.79.4.2 enum FlagStatus

Boolean Type definition.

Flag Status and Interrupt Flag Status type definition

Enumerator

***RESET***

***SET***

Definition at line 62 of file lpc\_types.h.

#### 6.79.4.3 enum FunctionalState

Functional State Definition.

Enumerator

***DISABLE***

***ENABLE***

Definition at line 68 of file lpc\_types.h.

#### 6.79.4.4 enum Status

@ Status type definition

Enumerator

***ERROR***

***SUCCESS***

Definition at line 74 of file lpc\_types.h.

#### 6.79.4.5 enum TRANSFER\_BLOCK\_T

Read/Write transfer type mode (Block or non-block)

Enumerator

***NONE\_BLOCKING*** None Blocking type

***BLOCKING*** Blocking type

Definition at line 79 of file lpc\_types.h.

## 6.80 LPC5411X multi-core use in LPCOpen

The LPC5411x device family has both a Cortex M4F and a Cortex M0+ CPU core that can execute applications simultaneously and signal events and status to the other core via the mailbox peripheral. One core is configured as the system master - usually the M4F core, while the other core is configured as the system slave. On system reset, the master core boots to FLASH at address 0x00000000, while the slave core sleeps until it is restarted by the master core. When the slave core is next reset by the master core application, it boots into the startup code at address 0x00000000. *Because both the M4F and M0 cores boot into the same FLASH address, the startup code must work correctly on both cores. The LPCOpen startup code that boots in FLASH is written using CM0+ instructions only and works on both CM4F and CM0+ cores.*

### LPC5411x master and slave core boot processes

Either core can be the master or slave core depending on the configuration of your device (part number). In some parts, only one core may be available (single core).

On system reset, both cores boot into the code at address 0x00000000 at the same time. For this initial boot after reset, ROM is mapped to address 0x00000000. The ROM code will eventually remap FLASH to address 0x00000000, boot the master core in FLASH address 0x00000000, and idle the slave core in ROM using the 'Wait For Interrupt' (WFI) instruction. The slave will remain in the WFI state until reset by the master core.

### LPCOpen startup code

LPCOpen startup code placed at FLASH address 0x00000000 is used for both the master and slave core. The startup code, when run on the master core, will start the execution of the application immediately - usually `SystemInit()` and then `main()`. The startup code, when run on the slave core, will look at the special execution and stack address provided by the master core. If the execution address is non-0x0, the stack address will be loaded from the value provided by the master core and then execution will be handed off to the address provided by the master core. If the execution address is 0x0, the slave code will sleep via WFI in the startup code in FLASH.

The core type - either M4F or M0+ - and the master or slave configuration can be determined at run-time for an application using the following functions:

```
/**
 * @brief Determine which MCU this code is running on
 * @return true if executing on the CM4, or false if executing on the CM0+
 */
bool Chip_CPU_IsM4Core(void);

/**
 * @brief Determine if this core is a slave or master
 * @return true if this MCU is operating as the master, or false if operating as a slave
 */
bool Chip_CPU_IsMasterCore(void);
@endverbatim
*
* The slave core can be reset and booted with a custom execution and stack
* address with the following functions:<br>
@verbatim
/**
 * @brief Setup M0+ boot and reset M0+ core
 * @param coentry : Pointer to boot entry point for M0+ core
 * @param costackptr : Pointer to where stack should be located for M0+ core
 * @return Nothing
 * @note Will setup boot stack and entry point, enable M0+ clock and then
 * reset M0+ core.
 */
void Chip_CPU_CM0Boot(uint32_t *coentry, uint32_t *costackptr);

/**
 * @brief Setup M4 boot and reset M4 core
 * @param coentry : Pointer to boot entry point for M4 core
 * @param costackptr : Pointer to where stack should be located for M4 core
 * @return Nothing
 * @note Will setup boot stack and entry point, enable M4 clock and then
 * reset M0+ core.
 */
```

```
*/  
void Chip_CPU_CM4Boot(uint32_t *coentry, uint32_t *costackptr);
```

**Special notes:**

**Do not use the core boot functions to boot the same core the function is running on - it will not work! All of the multi-core support functions are part of the powerlib library.**

The example code below safely boots the M0+ core from the M4F core, but can be executed on both cores safely.

```
/* Do not boot if on slave core */  
if (Chip_CPU_IsMasterCore() == true) {  
    /* Boot 'other' core, whichever is the slave */  
    if (Chip_CPU_IsM4Core()) {  
        /* This is M4 and master, so boot M0+ */  
        Chip_CPU_CM0Boot(bootAddr, stackAddr);  
    }  
}
```

**Debugger settings for IAR Embedded Workbench and Keil ARM MDK**

The images below show the necessary settings to debug the images in the M4F and M0+ cores. *LPCXpresso is not shown, but provides integrated support in the tool for both cores.*

*Debugger settings for both cores are almost exactly the same except for the CPU number for the target. Keil refers to this selection as "AP" in the CMSIS-DAP debugger dialog, while IAR uses "CPU number on target" on its CMSIS-DAP selection dialog. The name and description of this option may vary between tool chains and debuggers. The M4F core should use a value of '0', while the M0+ core should use a value of '1'.*

**IAR Embedded Workbench Cortex M4F debugger settings**

The image below shows the debugger setup specific to CMSIS-DAP and the M4F core. Other debuggers setup in IAR are similar.

**IAR Embedded Workbench Cortex M0+ debugger settings**

The image below shows the debugger setup specific to CMSIS-DAP and the M0+ core. Other debuggers setup in IAR are similar.

**Keil ARM MDK Cortex M4F debugger settings**

The image below shows the debugger setup specific to CMSIS-DAP and the M4F core. Other debuggers setup in Keil are similar.

**Keil ARM MDK Cortex M0+ debugger settings**

The image below shows the debugger setup specific to CMSIS-DAP and the M0+ core. Other debuggers setup in Keil are similar.

## 6.81 LPC5411x Chip specific drivers

### 6.81.1 Detailed Description

#### Modules

- [CHIP: LPC5411X A/D conversion driver](#)
- [CHIP: LPC5410x family CMSIS include files](#)
- [CHIP: LPC5411X 32-bit Timer driver](#)
- [CHIP: LPC5411X CPU multi-core support driver](#)
- [CHIP: LPC5411X Clock Driver](#)
- [CHIP: LPC5411X Cyclic Redundancy Check Engine driver](#)
- [CHIP: LPC5411X DMA Engine driver \(legacy\)](#)
- [CHIP: LPC5411X DMA Service driver](#)
- [CHIP: LPC5411X DMIC driver](#)
- [CHIP: LPC5411X Enhanced boot block support](#)
- [CHIP: LPC5411X GPIO driver](#)
- [CHIP: LPC5411X GPIO group driver](#)
- [CHIP: LPC5411X IOCON register block and driver](#)
- [CHIP: LPC5411X Input Mux Registers and Driver](#)
- [CHIP: LPC5411X Mailbox M4/M0+ driver](#)
- [CHIP: LPC5411X Micro Tick driver](#)
- [CHIP: LPC5411X Multi-Rate Timer driver](#)
- [CHIP: LPC5411X PLL Driver](#)
- [CHIP: LPC5411X Peripheral addresses and register set declarations](#)
- [CHIP: LPC5411X Pin Interrupt and Pattern Match driver](#)
- [CHIP: LPC5411X Power LIBRARY functions](#)
- [CHIP: LPC5411X Power Management declarations and functions](#)
- [CHIP: LPC5411X ROM API declarations and functions](#)
- [CHIP: LPC5411X Real Time clock](#)
- [CHIP: LPC5411X SPI driver](#)
- [CHIP: LPC5411X State Configurable Timer PWM driver](#)
- [CHIP: LPC5411X State Configurable Timer driver](#)
- [CHIP: LPC5411X System and Control Driver](#)
- [CHIP: LPC5411X UART Driver](#)
- [CHIP: LPC5411X Windowed Watchdog driver](#)
- [CHIP: LPC5411X flexcomm API](#)
- [CHIP: LPC5411X support functions](#)
- [CHIP: LPC5411x Chip driver build time options](#)
- [CHIP: LPC5411x I2C driver](#)
- [LPC5411X multi-core use in LPCOpen](#)



## 6.82 LPCOpen download and installation information

### 6.82.1 Detailed Description

#### LPCOpen source and projects package

The LPCOpen platform source and project packages are released as a ZIP files targeted for specific platforms and toolchains. The latest version of the LPCOpen package can be download at:

[LPCOpen software package page on LPCware.com](#)

#### LPCOpen software API documentation

The LPCOpen software API documentation package consists of descriptions of examples and software driver APIs. The LPCOpen software API documentation packages are available online as a separate downloadable ZIP file that is not part of the LPCOpen source and projects package.

[LPCOpen documentation package page on LPCware.com](#)

#### LPCOpen installation - PLEASE READ THIS!

FIXME - need install insts for flat releases and LPCXpresso releases - covered by QS guides? Installing LPCOpen is simple! Just unzip the LPCOpen ZIP file somewhere on your host system!

All projects and examples are path relative and will automatically adapt to where the files are installed.

***It is VERY HIGHLY recommended to install LPCOpen close to the root directory, as the arguments passed to the build tools can be get fairly long and cause environment issues. If you install the LPCOpen platform directory too deep in your filesystem, the paths passed as arguments to build programs may exceed the limit for passed arguments.***

***The recommended installation directory is: c:/LPCOpen***

#### Optional components needed for some LPCOpen examples

[emWin download and installation](#)

#### LPCOpen release history

[LPCOpen versioning and release history](#)

Copyright (C) 2013 NXP Semiconductors. All rights reserved.

#### Modules

- [emWin download and installation](#)

## 6.83 LPCOpen versioning and release history

For history and known issues, please check out the comprehensive version and history page at [www.LPCware.com](http://www.LPCware.com)

## 6.84 NVIC Functions

### 6.84.1 Detailed Description

Functions that manage interrupts and exceptions via the NVIC.

#### Macros

- `#define _BIT_SHIFT(IRQn) ( (((uint32_t)(IRQn) ) & 0x03) * 8 )`
- `#define _SHP_IDX(IRQn) ( (((uint32_t)(IRQn) & 0x0F)-8) >> 2) )`
- `#define _IP_IDX(IRQn) ( ((uint32_t)(IRQn) >> 2) )`

#### Functions

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

### 6.84.2 Macro Definition Documentation

#### 6.84.2.1 `#define _BIT_SHIFT( IRQn ) ( (((uint32_t)(IRQn) ) & 0x03) * 8 )`

Definition at line 615 of file core\_cm0plus.h.

6.84.2.2 `#define _IP_IDX( IRQn ) ( ((uint32_t)(IRQn) >> 2) )`

Definition at line 617 of file `core_cm0plus.h`.

6.84.2.3 `#define _SHP_IDX( IRQn ) ( (((uint32_t)(IRQn) & 0x0F)-8) >> 2) )`

Definition at line 616 of file `core_cm0plus.h`.

### 6.84.3 Function Documentation

6.84.3.1 `__STATIC_INLINE void NVIC_ClearPendingIRQ ( IRQn_Type IRQn )`

Clear Pending Interrupt.

The function clears the pending bit of an external interrupt.

Parameters

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

Definition at line 678 of file `core_cm0plus.h`.

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

Decode Priority.

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

Parameters

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

Definition at line 1620 of file `core_cm4.h`.

6.84.3.3 `__STATIC_INLINE void NVIC_DisableIRQ ( IRQn_Type IRQn )`

Disable External Interrupt.

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

Parameters

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

Definition at line 638 of file `core_cm0plus.h`.

6.84.3.4 `__STATIC_INLINE void NVIC_EnableIRQ ( IRQn_Type IRQn )`

Enable External Interrupt.

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

## Parameters

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

Definition at line 626 of file core\_cm0plus.h.

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

Encode Priority.

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

## Parameters

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

## Returns

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

Definition at line 1592 of file core\_cm4.h.

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

Get Active Interrupt.

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

## Parameters

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

## Returns

0 Interrupt status is not active.  
1 Interrupt status is active.

Definition at line 1535 of file core\_cm4.h.

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

Get Pending Interrupt.

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

## Parameters

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

## Returns

0 Interrupt status is not pending.  
1 Interrupt status is pending.

Definition at line 654 of file core\_cm0plus.h.

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

Get Interrupt Priority.

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

##### Parameters

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

##### Returns

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

Definition at line 715 of file `core_cm0plus.h`.

#### 6.84.3.9 `__STATIC_INLINE uint32_t NVIC_GetPriorityGrouping ( void )`

Get Priority Grouping.

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

##### Returns

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

Definition at line 1455 of file `core_cm4.h`.

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

Set Pending Interrupt.

The function sets the pending bit of an external interrupt.

##### Parameters

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

Definition at line 666 of file `core_cm0plus.h`.

#### 6.84.3.11 `__STATIC_INLINE void NVIC_SetPriority ( IRQn_Type IRQn, uint32_t priority )`

Set Interrupt Priority.

The function sets the priority of an interrupt.

##### Note

The priority cannot be set for every core interrupt.

##### Parameters

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

Definition at line 693 of file `core_cm0plus.h`.

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

Set Priority Grouping.

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

**Parameters**

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

Definition at line 1435 of file core\_cm4.h.

**6.84.3.13** `__STATIC_INLINE void NVIC_SystemReset ( void )`

System Reset.

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

Definition at line 729 of file core\_cm0plus.h.

## 6.85 NXP LPCXpresso LPC54114 LQFP board

### Documentation links

[LPCXpresso Boards Overview](#)

### Board software Support

BOARD\_LPCXPRESSO\_54114

### Jumper configuration

The recommended jumper configuration for the LPCXpresso LQFP LPC54114 is below:

JP1 open

JP2 shorted on LOC pins (pins 1-2)

P1 open

JP5 open (see CMSIS-DAP/BRIDGE firmware installation)

JP6 open

JP9 on pins 2-3

JP10 shorted

### Board recovery using the ISP pin

If the board's FLASH can no longer be programmed, the image in FLASH may be corrupt. To recover the board, the board must be reset with the ISP pin held low. To do this, reset and/or program the board with switch SW2 (ISP0) held down. The switch can be released once programming starts.

Copyright (C) 2013 NXP Semiconductors. All rights reserved.



## 6.86 Nested Vectored Interrupt Controller (NVIC)

### 6.86.1 Detailed Description

Type definitions for the NVIC Registers.

#### Data Structures

- struct [NVIC\\_Type](#)

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

#### Macros

- #define [NVIC\\_STIR\\_INTID\\_Pos](#) 0
- #define [NVIC\\_STIR\\_INTID\\_Msk](#) (0x1FFUL << NVIC\_STIR\_INTID\_Pos)

### 6.86.2 Macro Definition Documentation

#### 6.86.2.1 #define NVIC\_STIR\_INTID\_Msk (0x1FFUL << NVIC\_STIR\_INTID\_Pos)

STIR: INTLINESNUM Mask

Definition at line 355 of file core\_cm4.h.

#### 6.86.2.2 #define NVIC\_STIR\_INTID\_Pos 0

STIR: INTLINESNUM Position

Definition at line 354 of file core\_cm4.h.

## 6.87 RTOS support code

### 6.87.1 Detailed Description

RTOS support code includes custom drivers or support files for various Real-time operating systems and NXP devices.

#### Modules

- [RTOS: FreeRTOS support code](#)

## 6.88 RTOS: FreeRTOS support code

### 6.88.1 Detailed Description

**FreeRTOS** an open-source RTOS

#### Modules

- [Common FreeRTOS functions shared with multiple platforms](#)

## 6.89 Status and Control Registers

### 6.89.1 Detailed Description

Core Register type definitions.

#### Data Structures

- union [APSR\\_Type](#)  
*Union type to access the Application Program Status Register (APSR).*
- union [IPSR\\_Type](#)  
*Union type to access the Interrupt Program Status Register (IPSR).*
- union [xPSR\\_Type](#)  
*Union type to access the Special-Purpose Program Status Registers (xPSR).*
- union [CONTROL\\_Type](#)  
*Union type to access the Control Registers (CONTROL).*

## 6.90 Supported toolchains in LPCOpen

### 6.90.1 Detailed Description

The LPCOpen platform supports the following toolchains. Not all examples support all toolchains. See the example pages for more information about which toolchains are supported for specific examples.

Quickstart guides can be found online at the LPCOpen quickstart [page](#).

[Keil uVision support in LPCOpen](#) [IAR EWARM support in LPCOpen](#) [Code Red LPCXpresso support in LPCOpen](#)

### Modules

- [Code Red LPCXpresso support in LPCOpen](#)
- [IAR EWARM support in LPCOpen](#)
- [Keil uVision support in LPCOpen](#)

## 6.91 SysTick Functions

### 6.91.1 Detailed Description

Functions that configure the System.

#### Functions

- `__STATIC_INLINE uint32_t SysTick_Config (uint32_t ticks)`  
*System Tick Configuration.*

### 6.91.2 Function Documentation

#### 6.91.2.1 `__STATIC_INLINE uint32_t SysTick_Config ( uint32_t ticks )`

System Tick Configuration.

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

#### Parameters

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

#### Returns

- 0 Function succeeded.
- 1 Function failed.

#### Note

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

Definition at line 767 of file `core_cm0plus.h`.

## 6.92 System Control Block (SCB)

### 6.92.1 Detailed Description

Type definitions for the System Control Block Registers.

#### Data Structures

- struct [SCB\\_Type](#)  
*Structure type to access the System Control Block (SCB).*

#### Macros

- #define [SCB\\_CPUID\\_IMPLEMENTER\\_Pos](#) 24
- #define [SCB\\_CPUID\\_IMPLEMENTER\\_Msk](#) (0xFFUL << SCB\_CPUID\_IMPLEMENTER\_Pos)
- #define [SCB\\_CPUID\\_VARIANT\\_Pos](#) 20
- #define [SCB\\_CPUID\\_VARIANT\\_Msk](#) (0xFUL << SCB\_CPUID\_VARIANT\_Pos)
- #define [SCB\\_CPUID\\_ARCHITECTURE\\_Pos](#) 16
- #define [SCB\\_CPUID\\_ARCHITECTURE\\_Msk](#) (0xFUL << SCB\_CPUID\_ARCHITECTURE\_Pos)
- #define [SCB\\_CPUID\\_PARTNO\\_Pos](#) 4
- #define [SCB\\_CPUID\\_PARTNO\\_Msk](#) (0xFFFUL << SCB\_CPUID\_PARTNO\_Pos)
- #define [SCB\\_CPUID\\_REVISION\\_Pos](#) 0
- #define [SCB\\_CPUID\\_REVISION\\_Msk](#) (0xFUL << SCB\_CPUID\_REVISION\_Pos)
- #define [SCB\\_ICSR\\_NMIPENDSET\\_Pos](#) 31
- #define [SCB\\_ICSR\\_NMIPENDSET\\_Msk](#) (1UL << SCB\_ICSR\_NMIPENDSET\_Pos)
- #define [SCB\\_ICSR\\_PENDSVSET\\_Pos](#) 28
- #define [SCB\\_ICSR\\_PENDSVSET\\_Msk](#) (1UL << SCB\_ICSR\_PENDSVSET\_Pos)
- #define [SCB\\_ICSR\\_PENDSVCLR\\_Pos](#) 27
- #define [SCB\\_ICSR\\_PENDSVCLR\\_Msk](#) (1UL << SCB\_ICSR\_PENDSVCLR\_Pos)
- #define [SCB\\_ICSR\\_PENDSTSET\\_Pos](#) 26
- #define [SCB\\_ICSR\\_PENDSTSET\\_Msk](#) (1UL << SCB\_ICSR\_PENDSTSET\_Pos)
- #define [SCB\\_ICSR\\_PENDSTCLR\\_Pos](#) 25
- #define [SCB\\_ICSR\\_PENDSTCLR\\_Msk](#) (1UL << SCB\_ICSR\_PENDSTCLR\_Pos)
- #define [SCB\\_ICSR\\_ISRPREEMPT\\_Pos](#) 23
- #define [SCB\\_ICSR\\_ISRPREEMPT\\_Msk](#) (1UL << SCB\_ICSR\_ISRPREEMPT\_Pos)
- #define [SCB\\_ICSR\\_ISRPENDING\\_Pos](#) 22
- #define [SCB\\_ICSR\\_ISRPENDING\\_Msk](#) (1UL << SCB\_ICSR\_ISRPENDING\_Pos)
- #define [SCB\\_ICSR\\_VECTPENDING\\_Pos](#) 12
- #define [SCB\\_ICSR\\_VECTPENDING\\_Msk](#) (0x1FFUL << SCB\_ICSR\_VECTPENDING\_Pos)
- #define [SCB\\_ICSR\\_VECTACTIVE\\_Pos](#) 0
- #define [SCB\\_ICSR\\_VECTACTIVE\\_Msk](#) (0x1FFUL << SCB\_ICSR\_VECTACTIVE\_Pos)
- #define [SCB\\_AIRCR\\_VECTKEY\\_Pos](#) 16
- #define [SCB\\_AIRCR\\_VECTKEY\\_Msk](#) (0xFFFFUL << SCB\_AIRCR\_VECTKEY\_Pos)
- #define [SCB\\_AIRCR\\_VECTKEYSTAT\\_Pos](#) 16
- #define [SCB\\_AIRCR\\_VECTKEYSTAT\\_Msk](#) (0xFFFFUL << SCB\_AIRCR\_VECTKEYSTAT\_Pos)
- #define [SCB\\_AIRCR\\_ENDIANESS\\_Pos](#) 15
- #define [SCB\\_AIRCR\\_ENDIANESS\\_Msk](#) (1UL << SCB\_AIRCR\_ENDIANESS\_Pos)
- #define [SCB\\_AIRCR\\_SYSRESETREQ\\_Pos](#) 2
- #define [SCB\\_AIRCR\\_SYSRESETREQ\\_Msk](#) (1UL << SCB\_AIRCR\_SYSRESETREQ\_Pos)
- #define [SCB\\_AIRCR\\_VECTCLRACTIVE\\_Pos](#) 1
- #define [SCB\\_AIRCR\\_VECTCLRACTIVE\\_Msk](#) (1UL << SCB\_AIRCR\_VECTCLRACTIVE\_Pos)
- #define [SCB\\_SCR\\_SEVONPEND\\_Pos](#) 4
- #define [SCB\\_SCR\\_SEVONPEND\\_Msk](#) (1UL << SCB\_SCR\_SEVONPEND\_Pos)

- `#define SCB_SCR_SLEEPDEEP_Pos 2`
- `#define SCB_SCR_SLEEPDEEP_Msk (1UL << SCB_SCR_SLEEPDEEP_Pos)`
- `#define SCB_SCR_SLEEPONEXIT_Pos 1`
- `#define SCB_SCR_SLEEPONEXIT_Msk (1UL << SCB_SCR_SLEEPONEXIT_Pos)`
- `#define SCB_CCR_STKALIGN_Pos 9`
- `#define SCB_CCR_STKALIGN_Msk (1UL << SCB_CCR_STKALIGN_Pos)`
- `#define SCB_CCR_UNALIGN_TRP_Pos 3`
- `#define SCB_CCR_UNALIGN_TRP_Msk (1UL << SCB_CCR_UNALIGN_TRP_Pos)`
- `#define SCB_SHCSR_SVCALLPENDED_Pos 15`
- `#define SCB_SHCSR_SVCALLPENDED_Msk (1UL << SCB_SHCSR_SVCALLPENDED_Pos)`
- `#define SCB_CPUID_IMPLEMENTER_Pos 24`
- `#define SCB_CPUID_IMPLEMENTER_Msk (0xFFUL << SCB_CPUID_IMPLEMENTER_Pos)`
- `#define SCB_CPUID_VARIANT_Pos 20`
- `#define SCB_CPUID_VARIANT_Msk (0xFUL << SCB_CPUID_VARIANT_Pos)`
- `#define SCB_CPUID_ARCHITECTURE_Pos 16`
- `#define SCB_CPUID_ARCHITECTURE_Msk (0xFUL << SCB_CPUID_ARCHITECTURE_Pos)`
- `#define SCB_CPUID_PARTNO_Pos 4`
- `#define SCB_CPUID_PARTNO_Msk (0xFFFUL << SCB_CPUID_PARTNO_Pos)`
- `#define SCB_CPUID_REVISION_Pos 0`
- `#define SCB_CPUID_REVISION_Msk (0xFUL << SCB_CPUID_REVISION_Pos)`
- `#define SCB_ICSR_NMIPENDSET_Pos 31`
- `#define SCB_ICSR_NMIPENDSET_Msk (1UL << SCB_ICSR_NMIPENDSET_Pos)`
- `#define SCB_ICSR_PENDSVSET_Pos 28`
- `#define SCB_ICSR_PENDSVSET_Msk (1UL << SCB_ICSR_PENDSVSET_Pos)`
- `#define SCB_ICSR_PENDSVCLR_Pos 27`
- `#define SCB_ICSR_PENDSVCLR_Msk (1UL << SCB_ICSR_PENDSVCLR_Pos)`
- `#define SCB_ICSR_PENDSTSET_Pos 26`
- `#define SCB_ICSR_PENDSTSET_Msk (1UL << SCB_ICSR_PENDSTSET_Pos)`
- `#define SCB_ICSR_PENDSTCLR_Pos 25`
- `#define SCB_ICSR_PENDSTCLR_Msk (1UL << SCB_ICSR_PENDSTCLR_Pos)`
- `#define SCB_ICSR_ISRPREEMPT_Pos 23`
- `#define SCB_ICSR_ISRPREEMPT_Msk (1UL << SCB_ICSR_ISRPREEMPT_Pos)`
- `#define SCB_ICSR_ISRPENDING_Pos 22`
- `#define SCB_ICSR_ISRPENDING_Msk (1UL << SCB_ICSR_ISRPENDING_Pos)`
- `#define SCB_ICSR_VECTPENDING_Pos 12`
- `#define SCB_ICSR_VECTPENDING_Msk (0x1FFUL << SCB_ICSR_VECTPENDING_Pos)`
- `#define SCB_ICSR_RETTOBASE_Pos 11`
- `#define SCB_ICSR_RETTOBASE_Msk (1UL << SCB_ICSR_RETTOBASE_Pos)`
- `#define SCB_ICSR_VECTACTIVE_Pos 0`
- `#define SCB_ICSR_VECTACTIVE_Msk (0x1FFUL << SCB_ICSR_VECTACTIVE_Pos)`
- `#define SCB_VTOR_TBLOFF_Pos 7`
- `#define SCB_VTOR_TBLOFF_Msk (0x1FFFFFFUL << SCB_VTOR_TBLOFF_Pos)`
- `#define SCB_AIRCR_VECTKEY_Pos 16`
- `#define SCB_AIRCR_VECTKEY_Msk (0xFFFFUL << SCB_AIRCR_VECTKEY_Pos)`
- `#define SCB_AIRCR_VECTKEYSTAT_Pos 16`
- `#define SCB_AIRCR_VECTKEYSTAT_Msk (0xFFFFUL << SCB_AIRCR_VECTKEYSTAT_Pos)`
- `#define SCB_AIRCR_ENDIANESS_Pos 15`
- `#define SCB_AIRCR_ENDIANESS_Msk (1UL << SCB_AIRCR_ENDIANESS_Pos)`
- `#define SCB_AIRCR_PRIGROUP_Pos 8`
- `#define SCB_AIRCR_PRIGROUP_Msk (7UL << SCB_AIRCR_PRIGROUP_Pos)`
- `#define SCB_AIRCR_SYSRESETREQ_Pos 2`
- `#define SCB_AIRCR_SYSRESETREQ_Msk (1UL << SCB_AIRCR_SYSRESETREQ_Pos)`
- `#define SCB_AIRCR_VECTCLRACTIVE_Pos 1`
- `#define SCB_AIRCR_VECTCLRACTIVE_Msk (1UL << SCB_AIRCR_VECTCLRACTIVE_Pos)`
- `#define SCB_AIRCR_VECTRESET_Pos 0`



- #define [SCB\\_AIRCR\\_VECTRESET\\_Msk](#) (1UL << SCB\_AIRCR\_VECTRESET\_Pos)
- #define [SCB\\_SCR\\_SEVONPEND\\_Pos](#) 4
- #define [SCB\\_SCR\\_SEVONPEND\\_Msk](#) (1UL << SCB\_SCR\_SEVONPEND\_Pos)
- #define [SCB\\_SCR\\_SLEEPDEEP\\_Pos](#) 2
- #define [SCB\\_SCR\\_SLEEPDEEP\\_Msk](#) (1UL << SCB\_SCR\_SLEEPDEEP\_Pos)
- #define [SCB\\_SCR\\_SLEEPONEXIT\\_Pos](#) 1
- #define [SCB\\_SCR\\_SLEEPONEXIT\\_Msk](#) (1UL << SCB\_SCR\_SLEEPONEXIT\_Pos)
- #define [SCB\\_CCR\\_STKALIGN\\_Pos](#) 9
- #define [SCB\\_CCR\\_STKALIGN\\_Msk](#) (1UL << SCB\_CCR\_STKALIGN\_Pos)
- #define [SCB\\_CCR\\_BFHFNMIGN\\_Pos](#) 8
- #define [SCB\\_CCR\\_BFHFNMIGN\\_Msk](#) (1UL << SCB\_CCR\_BFHFNMIGN\_Pos)
- #define [SCB\\_CCR\\_DIV\\_0\\_TRP\\_Pos](#) 4
- #define [SCB\\_CCR\\_DIV\\_0\\_TRP\\_Msk](#) (1UL << SCB\_CCR\_DIV\_0\_TRP\_Pos)
- #define [SCB\\_CCR\\_UNALIGN\\_TRP\\_Pos](#) 3
- #define [SCB\\_CCR\\_UNALIGN\\_TRP\\_Msk](#) (1UL << SCB\_CCR\_UNALIGN\_TRP\_Pos)
- #define [SCB\\_CCR\\_USERSETMPEND\\_Pos](#) 1
- #define [SCB\\_CCR\\_USERSETMPEND\\_Msk](#) (1UL << SCB\_CCR\_USERSETMPEND\_Pos)
- #define [SCB\\_CCR\\_NONBASETHRDENA\\_Pos](#) 0
- #define [SCB\\_CCR\\_NONBASETHRDENA\\_Msk](#) (1UL << SCB\_CCR\_NONBASETHRDENA\_Pos)
- #define [SCB\\_SHCSR\\_USGFAULTENA\\_Pos](#) 18
- #define [SCB\\_SHCSR\\_USGFAULTENA\\_Msk](#) (1UL << SCB\_SHCSR\_USGFAULTENA\_Pos)
- #define [SCB\\_SHCSR\\_BUSFAULTENA\\_Pos](#) 17
- #define [SCB\\_SHCSR\\_BUSFAULTENA\\_Msk](#) (1UL << SCB\_SHCSR\_BUSFAULTENA\_Pos)
- #define [SCB\\_SHCSR\\_MEMFAULTENA\\_Pos](#) 16
- #define [SCB\\_SHCSR\\_MEMFAULTENA\\_Msk](#) (1UL << SCB\_SHCSR\_MEMFAULTENA\_Pos)
- #define [SCB\\_SHCSR\\_SVCALLPENDED\\_Pos](#) 15
- #define [SCB\\_SHCSR\\_SVCALLPENDED\\_Msk](#) (1UL << SCB\_SHCSR\_SVCALLPENDED\_Pos)
- #define [SCB\\_SHCSR\\_BUSFAULTPENDED\\_Pos](#) 14
- #define [SCB\\_SHCSR\\_BUSFAULTPENDED\\_Msk](#) (1UL << SCB\_SHCSR\_BUSFAULTPENDED\_Pos)
- #define [SCB\\_SHCSR\\_MEMFAULTPENDED\\_Pos](#) 13
- #define [SCB\\_SHCSR\\_MEMFAULTPENDED\\_Msk](#) (1UL << SCB\_SHCSR\_MEMFAULTPENDED\_Pos)
- #define [SCB\\_SHCSR\\_USGFAULTPENDED\\_Pos](#) 12
- #define [SCB\\_SHCSR\\_USGFAULTPENDED\\_Msk](#) (1UL << SCB\_SHCSR\_USGFAULTPENDED\_Pos)
- #define [SCB\\_SHCSR\\_SYSTICKACT\\_Pos](#) 11
- #define [SCB\\_SHCSR\\_SYSTICKACT\\_Msk](#) (1UL << SCB\_SHCSR\_SYSTICKACT\_Pos)
- #define [SCB\\_SHCSR\\_PENDSVACT\\_Pos](#) 10
- #define [SCB\\_SHCSR\\_PENDSVACT\\_Msk](#) (1UL << SCB\_SHCSR\_PENDSVACT\_Pos)
- #define [SCB\\_SHCSR\\_MONITORACT\\_Pos](#) 8
- #define [SCB\\_SHCSR\\_MONITORACT\\_Msk](#) (1UL << SCB\_SHCSR\_MONITORACT\_Pos)
- #define [SCB\\_SHCSR\\_SVCALLACT\\_Pos](#) 7
- #define [SCB\\_SHCSR\\_SVCALLACT\\_Msk](#) (1UL << SCB\_SHCSR\_SVCALLACT\_Pos)
- #define [SCB\\_SHCSR\\_USGFAULTACT\\_Pos](#) 3
- #define [SCB\\_SHCSR\\_USGFAULTACT\\_Msk](#) (1UL << SCB\_SHCSR\_USGFAULTACT\_Pos)
- #define [SCB\\_SHCSR\\_BUSFAULTACT\\_Pos](#) 1
- #define [SCB\\_SHCSR\\_BUSFAULTACT\\_Msk](#) (1UL << SCB\_SHCSR\_BUSFAULTACT\_Pos)
- #define [SCB\\_SHCSR\\_MEMFAULTACT\\_Pos](#) 0
- #define [SCB\\_SHCSR\\_MEMFAULTACT\\_Msk](#) (1UL << SCB\_SHCSR\_MEMFAULTACT\_Pos)
- #define [SCB\\_CFSR\\_USGFAULTSR\\_Pos](#) 16
- #define [SCB\\_CFSR\\_USGFAULTSR\\_Msk](#) (0xFFFFFUL << SCB\_CFSR\_USGFAULTSR\_Pos)
- #define [SCB\\_CFSR\\_BUSFAULTSR\\_Pos](#) 8
- #define [SCB\\_CFSR\\_BUSFAULTSR\\_Msk](#) (0xFFFUL << SCB\_CFSR\_BUSFAULTSR\_Pos)
- #define [SCB\\_CFSR\\_MEMFAULTSR\\_Pos](#) 0
- #define [SCB\\_CFSR\\_MEMFAULTSR\\_Msk](#) (0xFFFUL << SCB\_CFSR\_MEMFAULTSR\_Pos)
- #define [SCB\\_HFSR\\_DEBUGEVT\\_Pos](#) 31
- #define [SCB\\_HFSR\\_DEBUGEVT\\_Msk](#) (1UL << SCB\_HFSR\_DEBUGEVT\_Pos)

- `#define SCB_HFSR_FORCED_Pos` 30
- `#define SCB_HFSR_FORCED_Msk` (1UL << SCB\_HFSR\_FORCED\_Pos)
- `#define SCB_HFSR_VECTTBL_Pos` 1
- `#define SCB_HFSR_VECTTBL_Msk` (1UL << SCB\_HFSR\_VECTTBL\_Pos)
- `#define SCB_DFSR_EXTERNAL_Pos` 4
- `#define SCB_DFSR_EXTERNAL_Msk` (1UL << SCB\_DFSR\_EXTERNAL\_Pos)
- `#define SCB_DFSR_VCATCH_Pos` 3
- `#define SCB_DFSR_VCATCH_Msk` (1UL << SCB\_DFSR\_VCATCH\_Pos)
- `#define SCB_DFSR_DWTTRAP_Pos` 2
- `#define SCB_DFSR_DWTTRAP_Msk` (1UL << SCB\_DFSR\_DWTTRAP\_Pos)
- `#define SCB_DFSR_BKPT_Pos` 1
- `#define SCB_DFSR_BKPT_Msk` (1UL << SCB\_DFSR\_BKPT\_Pos)
- `#define SCB_DFSR_HALTED_Pos` 0
- `#define SCB_DFSR_HALTED_Msk` (1UL << SCB\_DFSR\_HALTED\_Pos)

## 6.92.2 Macro Definition Documentation

### 6.92.2.1 `#define SCB_AIRCR_ENDIANESS_Msk` (1UL << SCB\_AIRCR\_ENDIANESS\_Pos)

SCB AIRCR: ENDIANESS Mask

Definition at line 391 of file core\_cm0plus.h.

### 6.92.2.2 `#define SCB_AIRCR_ENDIANESS_Msk` (1UL << SCB\_AIRCR\_ENDIANESS\_Pos)

SCB AIRCR: ENDIANESS Mask

Definition at line 452 of file core\_cm4.h.

### 6.92.2.3 `#define SCB_AIRCR_ENDIANESS_Pos` 15

SCB AIRCR: ENDIANESS Position

Definition at line 390 of file core\_cm0plus.h.

### 6.92.2.4 `#define SCB_AIRCR_ENDIANESS_Pos` 15

SCB AIRCR: ENDIANESS Position

Definition at line 451 of file core\_cm4.h.

### 6.92.2.5 `#define SCB_AIRCR_PRIGROUP_Msk` (7UL << SCB\_AIRCR\_PRIGROUP\_Pos)

SCB AIRCR: PRIGROUP Mask

Definition at line 455 of file core\_cm4.h.

### 6.92.2.6 `#define SCB_AIRCR_PRIGROUP_Pos` 8

SCB AIRCR: PRIGROUP Position

Definition at line 454 of file core\_cm4.h.

**6.92.2.7** `#define SCB_AICR_SYSRESETREQ_Msk (1UL << SCB_AICR_SYSRESETREQ_Pos)`

SCB AICR: SYSRESETREQ Mask

Definition at line 394 of file core\_cm0plus.h.

**6.92.2.8** `#define SCB_AICR_SYSRESETREQ_Msk (1UL << SCB_AICR_SYSRESETREQ_Pos)`

SCB AICR: SYSRESETREQ Mask

Definition at line 458 of file core\_cm4.h.

**6.92.2.9** `#define SCB_AICR_SYSRESETREQ_Pos 2`

SCB AICR: SYSRESETREQ Position

Definition at line 393 of file core\_cm0plus.h.

**6.92.2.10** `#define SCB_AICR_SYSRESETREQ_Pos 2`

SCB AICR: SYSRESETREQ Position

Definition at line 457 of file core\_cm4.h.

**6.92.2.11** `#define SCB_AICR_VECTCLRACTIVE_Msk (1UL << SCB_AICR_VECTCLRACTIVE_Pos)`

SCB AICR: VECTCLRACTIVE Mask

Definition at line 397 of file core\_cm0plus.h.

**6.92.2.12** `#define SCB_AICR_VECTCLRACTIVE_Msk (1UL << SCB_AICR_VECTCLRACTIVE_Pos)`

SCB AICR: VECTCLRACTIVE Mask

Definition at line 461 of file core\_cm4.h.

**6.92.2.13** `#define SCB_AICR_VECTCLRACTIVE_Pos 1`

SCB AICR: VECTCLRACTIVE Position

Definition at line 396 of file core\_cm0plus.h.

**6.92.2.14** `#define SCB_AICR_VECTCLRACTIVE_Pos 1`

SCB AICR: VECTCLRACTIVE Position

Definition at line 460 of file core\_cm4.h.

**6.92.2.15** `#define SCB_AICR_VECTKEY_Msk (0xFFFFUL << SCB_AICR_VECTKEY_Pos)`

SCB AICR: VECTKEY Mask

Definition at line 385 of file core\_cm0plus.h.

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

SCB AIRCR: VECTKEY Mask

Definition at line 446 of file core\_cm4.h.

**6.92.2.17** `#define SCB_AIRCR_VECTKEY_Pos 16`

SCB AIRCR: VECTKEY Position

Definition at line 384 of file core\_cm0plus.h.

**6.92.2.18** `#define SCB_AIRCR_VECTKEY_Pos 16`

SCB AIRCR: VECTKEY Position

Definition at line 445 of file core\_cm4.h.

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

SCB AIRCR: VECTKEYSTAT Mask

Definition at line 388 of file core\_cm0plus.h.

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

SCB AIRCR: VECTKEYSTAT Mask

Definition at line 449 of file core\_cm4.h.

**6.92.2.21** `#define SCB_AIRCR_VECTKEYSTAT_Pos 16`

SCB AIRCR: VECTKEYSTAT Position

Definition at line 387 of file core\_cm0plus.h.

**6.92.2.22** `#define SCB_AIRCR_VECTKEYSTAT_Pos 16`

SCB AIRCR: VECTKEYSTAT Position

Definition at line 448 of file core\_cm4.h.

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

SCB AIRCR: VECTRESET Mask

Definition at line 464 of file core\_cm4.h.

**6.92.2.24** `#define SCB_AIRCR_VECTRESET_Pos 0`

SCB AIRCR: VECTRESET Position

Definition at line 463 of file core\_cm4.h.

6.92.2.25 `#define SCB_CCR_BFHFNMIGN_Msk (1UL << SCB_CCR_BFHFNMIGN_Pos)`

SCB CCR: BFHFNMIGN Mask

Definition at line 481 of file core\_cm4.h.

6.92.2.26 `#define SCB_CCR_BFHFNMIGN_Pos 8`

SCB CCR: BFHFNMIGN Position

Definition at line 480 of file core\_cm4.h.

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

SCB CCR: DIV\_0\_TRP Mask

Definition at line 484 of file core\_cm4.h.

6.92.2.28 `#define SCB_CCR_DIV_0_TRP_Pos 4`

SCB CCR: DIV\_0\_TRP Position

Definition at line 483 of file core\_cm4.h.

6.92.2.29 `#define SCB_CCR_NONBASETHRDENA_Msk (1UL << SCB_CCR_NONBASETHRDENA_Pos)`

SCB CCR: NONBASETHRDENA Mask

Definition at line 493 of file core\_cm4.h.

6.92.2.30 `#define SCB_CCR_NONBASETHRDENA_Pos 0`

SCB CCR: NONBASETHRDENA Position

Definition at line 492 of file core\_cm4.h.

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

SCB CCR: STKALIGN Mask

Definition at line 411 of file core\_cm0plus.h.

6.92.2.32 `#define SCB_CCR_STKALIGN_Pos 0`

SCB CCR: STKALIGN Mask

Definition at line 478 of file core\_cm4.h.

6.92.2.33 `#define SCB_CCR_STKALIGN_Pos 9`

SCB CCR: STKALIGN Position

Definition at line 410 of file core\_cm0plus.h.

**6.92.2.34** `#define SCB_CCR_STKALIGN_Pos 9`

SCB CCR: STKALIGN Position

Definition at line 477 of file core\_cm4.h.

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

SCB CCR: UNALIGN\_TRP Mask

Definition at line 414 of file core\_cm0plus.h.

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

SCB CCR: UNALIGN\_TRP Mask

Definition at line 487 of file core\_cm4.h.

**6.92.2.37** `#define SCB_CCR_UNALIGN_TRP_Pos 3`

SCB CCR: UNALIGN\_TRP Position

Definition at line 413 of file core\_cm0plus.h.

**6.92.2.38** `#define SCB_CCR_UNALIGN_TRP_Pos 3`

SCB CCR: UNALIGN\_TRP Position

Definition at line 486 of file core\_cm4.h.

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

SCB CCR: USERSETMPEND Mask

Definition at line 490 of file core\_cm4.h.

**6.92.2.40** `#define SCB_CCR_USERSETMPEND_Pos 1`

SCB CCR: USERSETMPEND Position

Definition at line 489 of file core\_cm4.h.

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

SCB CFSR: Bus Fault Status Register Mask

Definition at line 543 of file core\_cm4.h.

**6.92.2.42** `#define SCB_CFSR_BUSFAULTSR_Pos 8`

SCB CFSR: Bus Fault Status Register Position

Definition at line 542 of file core\_cm4.h.

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

SCB CFSR: Memory Manage Fault Status Register Mask

Definition at line 546 of file core\_cm4.h.

**6.92.2.44** `#define SCB_CFSR_MEMFAULTSR_Pos 0`

SCB CFSR: Memory Manage Fault Status Register Position

Definition at line 545 of file core\_cm4.h.

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

SCB CFSR: Usage Fault Status Register Mask

Definition at line 540 of file core\_cm4.h.

**6.92.2.46** `#define SCB_CFSR_USGFAULTSR_Pos 16`

SCB CFSR: Usage Fault Status Register Position

Definition at line 539 of file core\_cm4.h.

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

SCB CPUID: ARCHITECTURE Mask

Definition at line 341 of file core\_cm0plus.h.

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

SCB CPUID: ARCHITECTURE Mask

Definition at line 401 of file core\_cm4.h.

**6.92.2.49** `#define SCB_CPUID_ARCHITECTURE_Pos 16`

SCB CPUID: ARCHITECTURE Position

Definition at line 340 of file core\_cm0plus.h.

**6.92.2.50** `#define SCB_CPUID_ARCHITECTURE_Pos 16`

SCB CPUID: ARCHITECTURE Position

Definition at line 400 of file core\_cm4.h.

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

SCB CPUID: IMPLEMENTER Mask

Definition at line 335 of file core\_cm0plus.h.

**6.92.2.52** `#define SCB_CPUID_IMPLEMENTER_Msk (0xFFFUL << SCB_CPUID_IMPLEMENTER_Pos)`

SCB CPUID: IMPLEMENTER Mask

Definition at line 395 of file core\_cm4.h.

**6.92.2.53** `#define SCB_CPUID_IMPLEMENTER_Pos 24`

SCB CPUID: IMPLEMENTER Position

Definition at line 334 of file core\_cm0plus.h.

**6.92.2.54** `#define SCB_CPUID_IMPLEMENTER_Pos 24`

SCB CPUID: IMPLEMENTER Position

Definition at line 394 of file core\_cm4.h.

**6.92.2.55** `#define SCB_CPUID_PARTNO_Msk (0xFFFFFUL << SCB_CPUID_PARTNO_Pos)`

SCB CPUID: PARTNO Mask

Definition at line 344 of file core\_cm0plus.h.

**6.92.2.56** `#define SCB_CPUID_PARTNO_Msk (0xFFFFFUL << SCB_CPUID_PARTNO_Pos)`

SCB CPUID: PARTNO Mask

Definition at line 404 of file core\_cm4.h.

**6.92.2.57** `#define SCB_CPUID_PARTNO_Pos 4`

SCB CPUID: PARTNO Position

Definition at line 343 of file core\_cm0plus.h.

**6.92.2.58** `#define SCB_CPUID_PARTNO_Pos 4`

SCB CPUID: PARTNO Position

Definition at line 403 of file core\_cm4.h.

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

SCB CPUID: REVISION Mask

Definition at line 347 of file core\_cm0plus.h.

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

SCB CPUID: REVISION Mask

Definition at line 407 of file core\_cm4.h.



6.92.2.61 `#define SCB_CPUID_REVISION_Pos 0`

SCB CPUID: REVISION Position

Definition at line 346 of file core\_cm0plus.h.

6.92.2.62 `#define SCB_CPUID_REVISION_Pos 0`

SCB CPUID: REVISION Position

Definition at line 406 of file core\_cm4.h.

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

SCB CPUID: VARIANT Mask

Definition at line 338 of file core\_cm0plus.h.

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

SCB CPUID: VARIANT Mask

Definition at line 398 of file core\_cm4.h.

6.92.2.65 `#define SCB_CPUID_VARIANT_Pos 20`

SCB CPUID: VARIANT Position

Definition at line 337 of file core\_cm0plus.h.

6.92.2.66 `#define SCB_CPUID_VARIANT_Pos 20`

SCB CPUID: VARIANT Position

Definition at line 397 of file core\_cm4.h.

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

SCB DFSR: BKPT Mask

Definition at line 569 of file core\_cm4.h.

6.92.2.68 `#define SCB_DFSR_BKPT_Pos 1`

SCB DFSR: BKPT Position

Definition at line 568 of file core\_cm4.h.

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

SCB DFSR: DWTTRAP Mask

Definition at line 566 of file core\_cm4.h.

**6.92.2.70 #define SCB\_DFSR\_DWTTRAP\_Pos 2**

SCB DFSR: DWTTRAP Position

Definition at line 565 of file core\_cm4.h.

**6.92.2.71 #define SCB\_DFSR\_EXTERNAL\_Msk (1UL << SCB\_DFSR\_EXTERNAL\_Pos)**

SCB DFSR: EXTERNAL Mask

Definition at line 560 of file core\_cm4.h.

**6.92.2.72 #define SCB\_DFSR\_EXTERNAL\_Pos 4**

SCB DFSR: EXTERNAL Position

Definition at line 559 of file core\_cm4.h.

**6.92.2.73 #define SCB\_DFSR\_HALTED\_Msk (1UL << SCB\_DFSR\_HALTED\_Pos)**

SCB DFSR: HALTED Mask

Definition at line 572 of file core\_cm4.h.

**6.92.2.74 #define SCB\_DFSR\_HALTED\_Pos 0**

SCB DFSR: HALTED Position

Definition at line 571 of file core\_cm4.h.

**6.92.2.75 #define SCB\_DFSR\_VCATCH\_Msk (1UL << SCB\_DFSR\_VCATCH\_Pos)**

SCB DFSR: VCATCH Mask

Definition at line 563 of file core\_cm4.h.

**6.92.2.76 #define SCB\_DFSR\_VCATCH\_Pos 3**

SCB DFSR: VCATCH Position

Definition at line 562 of file core\_cm4.h.

**6.92.2.77 #define SCB\_HFSR\_DEBUGEVT\_Msk (1UL << SCB\_HFSR\_DEBUGEVT\_Pos)**

SCB HFSR: DEBUGEVT Mask

Definition at line 550 of file core\_cm4.h.

**6.92.2.78 #define SCB\_HFSR\_DEBUGEVT\_Pos 31**

SCB HFSR: DEBUGEVT Position

Definition at line 549 of file core\_cm4.h.

6.92.2.79 **#define SCB\_HFSR\_FORCED\_Msk** (1UL << SCB\_HFSR\_FORCED\_Pos)

SCB HFSR: FORCED Mask

Definition at line 553 of file core\_cm4.h.

6.92.2.80 **#define SCB\_HFSR\_FORCED\_Pos** 30

SCB HFSR: FORCED Position

Definition at line 552 of file core\_cm4.h.

6.92.2.81 **#define SCB\_HFSR\_VECTTBL\_Msk** (1UL << SCB\_HFSR\_VECTTBL\_Pos)

SCB HFSR: VECTTBL Mask

Definition at line 556 of file core\_cm4.h.

6.92.2.82 **#define SCB\_HFSR\_VECTTBL\_Pos** 1

SCB HFSR: VECTTBL Position

Definition at line 555 of file core\_cm4.h.

6.92.2.83 **#define SCB\_ICSR\_ISRPENDING\_Msk** (1UL << SCB\_ICSR\_ISRPENDING\_Pos)

SCB ICSR: ISRPENDING Mask

Definition at line 369 of file core\_cm0plus.h.

6.92.2.84 **#define SCB\_ICSR\_ISRPENDING\_Msk** (1UL << SCB\_ICSR\_ISRPENDING\_Pos)

SCB ICSR: ISRPENDING Mask

Definition at line 429 of file core\_cm4.h.

6.92.2.85 **#define SCB\_ICSR\_ISRPENDING\_Pos** 22

SCB ICSR: ISRPENDING Position

Definition at line 368 of file core\_cm0plus.h.

6.92.2.86 **#define SCB\_ICSR\_ISRPENDING\_Pos** 22

SCB ICSR: ISRPENDING Position

Definition at line 428 of file core\_cm4.h.

6.92.2.87 **#define SCB\_ICSR\_ISRPREEMPT\_Msk** (1UL << SCB\_ICSR\_ISRPREEMPT\_Pos)

SCB ICSR: ISRPREEMPT Mask

Definition at line 366 of file core\_cm0plus.h.

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

SCB ICSR: ISRPREEMPT Mask

Definition at line 426 of file core\_cm4.h.

**6.92.2.89** `#define SCB_ICSR_ISRPREEMPT_Pos 23`

SCB ICSR: ISRPREEMPT Position

Definition at line 365 of file core\_cm0plus.h.

**6.92.2.90** `#define SCB_ICSR_ISRPREEMPT_Pos 23`

SCB ICSR: ISRPREEMPT Position

Definition at line 425 of file core\_cm4.h.

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

SCB ICSR: NMIPENDSET Mask

Definition at line 351 of file core\_cm0plus.h.

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

SCB ICSR: NMIPENDSET Mask

Definition at line 411 of file core\_cm4.h.

**6.92.2.93** `#define SCB_ICSR_NMIPENDSET_Pos 31`

SCB ICSR: NMIPENDSET Position

Definition at line 350 of file core\_cm0plus.h.

**6.92.2.94** `#define SCB_ICSR_NMIPENDSET_Pos 31`

SCB ICSR: NMIPENDSET Position

Definition at line 410 of file core\_cm4.h.

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

SCB ICSR: PENDSTCLR Mask

Definition at line 363 of file core\_cm0plus.h.

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

SCB ICSR: PENDSTCLR Mask

Definition at line 423 of file core\_cm4.h.

6.92.2.97 `#define SCB_ICSR_PENDSTCLR_Pos 25`

SCB ICSR: PENDSTCLR Position

Definition at line 362 of file core\_cm0plus.h.

6.92.2.98 `#define SCB_ICSR_PENDSTCLR_Pos 25`

SCB ICSR: PENDSTCLR Position

Definition at line 422 of file core\_cm4.h.

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

SCB ICSR: PENDSTSET Mask

Definition at line 360 of file core\_cm0plus.h.

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

SCB ICSR: PENDSTSET Mask

Definition at line 420 of file core\_cm4.h.

6.92.2.101 `#define SCB_ICSR_PENDSTSET_Pos 26`

SCB ICSR: PENDSTSET Position

Definition at line 359 of file core\_cm0plus.h.

6.92.2.102 `#define SCB_ICSR_PENDSTSET_Pos 26`

SCB ICSR: PENDSTSET Position

Definition at line 419 of file core\_cm4.h.

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

SCB ICSR: PENDSVCLR Mask

Definition at line 357 of file core\_cm0plus.h.

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

SCB ICSR: PENDSVCLR Mask

Definition at line 417 of file core\_cm4.h.

6.92.2.105 `#define SCB_ICSR_PENDSVCLR_Pos 27`

SCB ICSR: PENDSVCLR Position

Definition at line 356 of file core\_cm0plus.h.

6.92.2.106 `#define SCB_ICSR_PENDSVCLR_Pos 27`

SCB ICSR: PENDSVCLR Position

Definition at line 416 of file core\_cm4.h.

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

SCB ICSR: PENDSVSET Mask

Definition at line 354 of file core\_cm0plus.h.

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

SCB ICSR: PENDSVSET Mask

Definition at line 414 of file core\_cm4.h.

6.92.2.109 `#define SCB_ICSR_PENDSVSET_Pos 28`

SCB ICSR: PENDSVSET Position

Definition at line 353 of file core\_cm0plus.h.

6.92.2.110 `#define SCB_ICSR_PENDSVSET_Pos 28`

SCB ICSR: PENDSVSET Position

Definition at line 413 of file core\_cm4.h.

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

SCB ICSR: RETTOBASE Mask

Definition at line 435 of file core\_cm4.h.

6.92.2.112 `#define SCB_ICSR_RETTOBASE_Pos 11`

SCB ICSR: RETTOBASE Position

Definition at line 434 of file core\_cm4.h.

6.92.2.113 `#define SCB_ICSR_VECTACTIVE_Msk (0x1FFUL << SCB_ICSR_VECTACTIVE_Pos)`

SCB ICSR: VECTACTIVE Mask

Definition at line 375 of file core\_cm0plus.h.

6.92.2.114 `#define SCB_ICSR_VECTACTIVE_Msk (0x1FFUL << SCB_ICSR_VECTACTIVE_Pos)`

SCB ICSR: VECTACTIVE Mask

Definition at line 438 of file core\_cm4.h.

6.92.2.115 `#define SCB_ICSR_VECTACTIVE_Pos 0`

SCB ICSR: VECTACTIVE Position

Definition at line 374 of file core\_cm0plus.h.

6.92.2.116 `#define SCB_ICSR_VECTACTIVE_Pos 0`

SCB ICSR: VECTACTIVE Position

Definition at line 437 of file core\_cm4.h.

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

SCB ICSR: VECTPENDING Mask

Definition at line 372 of file core\_cm0plus.h.

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

SCB ICSR: VECTPENDING Mask

Definition at line 432 of file core\_cm4.h.

6.92.2.119 `#define SCB_ICSR_VECTPENDING_Pos 12`

SCB ICSR: VECTPENDING Position

Definition at line 371 of file core\_cm0plus.h.

6.92.2.120 `#define SCB_ICSR_VECTPENDING_Pos 12`

SCB ICSR: VECTPENDING Position

Definition at line 431 of file core\_cm4.h.

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

SCB SCR: SEVONPEND Mask

Definition at line 401 of file core\_cm0plus.h.

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

SCB SCR: SEVONPEND Mask

Definition at line 468 of file core\_cm4.h.

6.92.2.123 `#define SCB_SCR_SEVONPEND_Pos 4`

SCB SCR: SEVONPEND Position

Definition at line 400 of file core\_cm0plus.h.

**6.92.2.124** `#define SCB_SCR_SEVONPEND_Pos 4`

SCB SCR: SEVONPEND Position

Definition at line 467 of file core\_cm4.h.

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

SCB SCR: SLEEPDEEP Mask

Definition at line 404 of file core\_cm0plus.h.

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

SCB SCR: SLEEPDEEP Mask

Definition at line 471 of file core\_cm4.h.

**6.92.2.127** `#define SCB_SCR_SLEEPDEEP_Pos 2`

SCB SCR: SLEEPDEEP Position

Definition at line 403 of file core\_cm0plus.h.

**6.92.2.128** `#define SCB_SCR_SLEEPDEEP_Pos 2`

SCB SCR: SLEEPDEEP Position

Definition at line 470 of file core\_cm4.h.

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

SCB SCR: SLEEPONEXIT Mask

Definition at line 407 of file core\_cm0plus.h.

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

SCB SCR: SLEEPONEXIT Mask

Definition at line 474 of file core\_cm4.h.

**6.92.2.131** `#define SCB_SCR_SLEEPONEXIT_Pos 1`

SCB SCR: SLEEPONEXIT Position

Definition at line 406 of file core\_cm0plus.h.

**6.92.2.132** `#define SCB_SCR_SLEEPONEXIT_Pos 1`

SCB SCR: SLEEPONEXIT Position

Definition at line 473 of file core\_cm4.h.



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

SCB SHCSR: BUSFAULTACT Mask

Definition at line 533 of file core\_cm4.h.

6.92.2.134 `#define SCB_SHCSR_BUSFAULTACT_Pos 1`

SCB SHCSR: BUSFAULTACT Position

Definition at line 532 of file core\_cm4.h.

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

SCB SHCSR: BUSFAULTENA Mask

Definition at line 500 of file core\_cm4.h.

6.92.2.136 `#define SCB_SHCSR_BUSFAULTENA_Pos 17`

SCB SHCSR: BUSFAULTENA Position

Definition at line 499 of file core\_cm4.h.

6.92.2.137 `#define SCB_SHCSR_BUSFAULTPENDEDED_Msk (1UL << SCB_SHCSR_BUSFAULTPENDEDED_Pos)`

SCB SHCSR: BUSFAULTPENDEDED Mask

Definition at line 509 of file core\_cm4.h.

6.92.2.138 `#define SCB_SHCSR_BUSFAULTPENDEDED_Pos 14`

SCB SHCSR: BUSFAULTPENDEDED Position

Definition at line 508 of file core\_cm4.h.

6.92.2.139 `#define SCB_SHCSR_MEMFAULTACT_Msk (1UL << SCB_SHCSR_MEMFAULTACT_Pos)`

SCB SHCSR: MEMFAULTACT Mask

Definition at line 536 of file core\_cm4.h.

6.92.2.140 `#define SCB_SHCSR_MEMFAULTACT_Pos 0`

SCB SHCSR: MEMFAULTACT Position

Definition at line 535 of file core\_cm4.h.

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

SCB SHCSR: MEMFAULTENA Mask

Definition at line 503 of file core\_cm4.h.

6.92.2.142 `#define SCB_SHCSR_MEMFAULTENA_Pos 16`

SCB SHCSR: MEMFAULTENA Position

Definition at line 502 of file core\_cm4.h.

6.92.2.143 `#define SCB_SHCSR_MEMFAULTPENDEDED_Msk (1UL << SCB_SHCSR_MEMFAULTPENDEDED_Pos)`

SCB SHCSR: MEMFAULTPENDEDED Mask

Definition at line 512 of file core\_cm4.h.

6.92.2.144 `#define SCB_SHCSR_MEMFAULTPENDEDED_Pos 13`

SCB SHCSR: MEMFAULTPENDEDED Position

Definition at line 511 of file core\_cm4.h.

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

SCB SHCSR: MONITORACT Mask

Definition at line 524 of file core\_cm4.h.

6.92.2.146 `#define SCB_SHCSR_MONITORACT_Pos 8`

SCB SHCSR: MONITORACT Position

Definition at line 523 of file core\_cm4.h.

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

SCB SHCSR: PENDSVACT Mask

Definition at line 521 of file core\_cm4.h.

6.92.2.148 `#define SCB_SHCSR_PENDSVACT_Pos 10`

SCB SHCSR: PENDSVACT Position

Definition at line 520 of file core\_cm4.h.

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

SCB SHCSR: SVCALLACT Mask

Definition at line 527 of file core\_cm4.h.

6.92.2.150 `#define SCB_SHCSR_SVCALLACT_Pos 7`

SCB SHCSR: SVCALLACT Position

Definition at line 526 of file core\_cm4.h.

6.92.2.151 `#define SCB_SHCSR_SVCALLPENDEDED_Msk (1UL << SCB_SHCSR_SVCALLPENDEDED_Pos)`

SCB SHCSR: SVCALLPENDEDED Mask

Definition at line 418 of file core\_cm0plus.h.

6.92.2.152 `#define SCB_SHCSR_SVCALLPENDEDED_Msk (1UL << SCB_SHCSR_SVCALLPENDEDED_Pos)`

SCB SHCSR: SVCALLPENDEDED Mask

Definition at line 506 of file core\_cm4.h.

6.92.2.153 `#define SCB_SHCSR_SVCALLPENDEDED_Pos 15`

SCB SHCSR: SVCALLPENDEDED Position

Definition at line 417 of file core\_cm0plus.h.

6.92.2.154 `#define SCB_SHCSR_SVCALLPENDEDED_Pos 15`

SCB SHCSR: SVCALLPENDEDED Position

Definition at line 505 of file core\_cm4.h.

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

SCB SHCSR: SYSTICKACT Mask

Definition at line 518 of file core\_cm4.h.

6.92.2.156 `#define SCB_SHCSR_SYSTICKACT_Pos 11`

SCB SHCSR: SYSTICKACT Position

Definition at line 517 of file core\_cm4.h.

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

SCB SHCSR: USGFAULTACT Mask

Definition at line 530 of file core\_cm4.h.

6.92.2.158 `#define SCB_SHCSR_USGFAULTACT_Pos 3`

SCB SHCSR: USGFAULTACT Position

Definition at line 529 of file core\_cm4.h.

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

SCB SHCSR: USGFAULTENA Mask

Definition at line 497 of file core\_cm4.h.

6.92.2.160 `#define SCB_SHCSR_USGFAULTENA_Pos 18`

SCB SHCSR: USGFAULTENA Position

Definition at line 496 of file core\_cm4.h.

6.92.2.161 `#define SCB_SHCSR_USGFAULTPENDEDED_Msk (1UL << SCB_SHCSR_USGFAULTPENDEDED_Pos)`

SCB SHCSR: USGFAULTPENDEDED Mask

Definition at line 515 of file core\_cm4.h.

6.92.2.162 `#define SCB_SHCSR_USGFAULTPENDEDED_Pos 12`

SCB SHCSR: USGFAULTPENDEDED Position

Definition at line 514 of file core\_cm4.h.

6.92.2.163 `#define SCB_VTOR_TBLOFF_Msk (0x1FFFFFFUL << SCB_VTOR_TBLOFF_Pos)`

SCB VTOR: TBLOFF Mask

Definition at line 442 of file core\_cm4.h.

6.92.2.164 `#define SCB_VTOR_TBLOFF_Pos 7`

SCB VTOR: TBLOFF Position

Definition at line 441 of file core\_cm4.h.

## 6.93 System Controls not in SCB (SCnSCB)

### 6.93.1 Detailed Description

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

#### Data Structures

- struct [SCnSCB\\_Type](#)

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

#### Macros

- #define [SCnSCB\\_ICTR\\_INTLINESNUM\\_Pos](#) 0
- #define [SCnSCB\\_ICTR\\_INTLINESNUM\\_Msk](#) (0xFUL << SCnSCB\_ICTR\_INTLINESNUM\_Pos)
- #define [SCnSCB\\_ACTLR\\_DISOOF\\_Pos](#) 9
- #define [SCnSCB\\_ACTLR\\_DISOOF\\_Msk](#) (1UL << SCnSCB\_ACTLR\_DISOOF\_Pos)
- #define [SCnSCB\\_ACTLR\\_DISFPCA\\_Pos](#) 8
- #define [SCnSCB\\_ACTLR\\_DISFPCA\\_Msk](#) (1UL << SCnSCB\_ACTLR\_DISFPCA\_Pos)
- #define [SCnSCB\\_ACTLR\\_DISFOLD\\_Pos](#) 2
- #define [SCnSCB\\_ACTLR\\_DISFOLD\\_Msk](#) (1UL << SCnSCB\_ACTLR\_DISFOLD\_Pos)
- #define [SCnSCB\\_ACTLR\\_DISDEFWBUF\\_Pos](#) 1
- #define [SCnSCB\\_ACTLR\\_DISDEFWBUF\\_Msk](#) (1UL << SCnSCB\_ACTLR\_DISDEFWBUF\_Pos)
- #define [SCnSCB\\_ACTLR\\_DISMCYCINT\\_Pos](#) 0
- #define [SCnSCB\\_ACTLR\\_DISMCYCINT\\_Msk](#) (1UL << SCnSCB\_ACTLR\_DISMCYCINT\_Pos)

### 6.93.2 Macro Definition Documentation

#### 6.93.2.1 #define SCnSCB\_ACTLR\_DISDEFWBUF\_Msk (1UL << SCnSCB\_ACTLR\_DISDEFWBUF\_Pos)

ACTLR: DISDEFWBUF Mask

Definition at line 607 of file core\_cm4.h.

#### 6.93.2.2 #define SCnSCB\_ACTLR\_DISDEFWBUF\_Pos 1

ACTLR: DISDEFWBUF Position

Definition at line 606 of file core\_cm4.h.

#### 6.93.2.3 #define SCnSCB\_ACTLR\_DISFOLD\_Msk (1UL << SCnSCB\_ACTLR\_DISFOLD\_Pos)

ACTLR: DISFOLD Mask

Definition at line 604 of file core\_cm4.h.

#### 6.93.2.4 #define SCnSCB\_ACTLR\_DISFOLD\_Pos 2

ACTLR: DISFOLD Position

Definition at line 603 of file core\_cm4.h.

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

ACTLR: DISFPCA Mask

Definition at line 601 of file core\_cm4.h.

**6.93.2.6** `#define SCnSCB_ACTLR_DISFPCA_Pos 8`

ACTLR: DISFPCA Position

Definition at line 600 of file core\_cm4.h.

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

ACTLR: DISMCYCINT Mask

Definition at line 610 of file core\_cm4.h.

**6.93.2.8** `#define SCnSCB_ACTLR_DISMCYCINT_Pos 0`

ACTLR: DISMCYCINT Position

Definition at line 609 of file core\_cm4.h.

**6.93.2.9** `#define SCnSCB_ACTLR_DISOOFM_Msk (1UL << SCnSCB_ACTLR_DISOOFM_Pos)`

ACTLR: DISOOFM Mask

Definition at line 598 of file core\_cm4.h.

**6.93.2.10** `#define SCnSCB_ACTLR_DISOOFM_Pos 9`

ACTLR: DISOOFM Position

Definition at line 597 of file core\_cm4.h.

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

ICTR: INTLINESNUM Mask

Definition at line 594 of file core\_cm4.h.

**6.93.2.12** `#define SCnSCB_ICTR_INTLINESNUM_Pos 0`

ICTR: INTLINESNUM Position

Definition at line 593 of file core\_cm4.h.

## 6.94 System Tick Timer (SysTick)

### 6.94.1 Detailed Description

Type definitions for the System Timer Registers.

#### Data Structures

- struct [SysTick\\_Type](#)

*Structure type to access the System Timer (SysTick).*

#### Macros

- #define [SysTick\\_CTRL\\_COUNTFLAG\\_Pos](#) 16
- #define [SysTick\\_CTRL\\_COUNTFLAG\\_Msk](#) (1UL << SysTick\_CTRL\_COUNTFLAG\_Pos)
- #define [SysTick\\_CTRL\\_CLKSOURCE\\_Pos](#) 2
- #define [SysTick\\_CTRL\\_CLKSOURCE\\_Msk](#) (1UL << SysTick\_CTRL\_CLKSOURCE\_Pos)
- #define [SysTick\\_CTRL\\_TICKINT\\_Pos](#) 1
- #define [SysTick\\_CTRL\\_TICKINT\\_Msk](#) (1UL << SysTick\_CTRL\_TICKINT\_Pos)
- #define [SysTick\\_CTRL\\_ENABLE\\_Pos](#) 0
- #define [SysTick\\_CTRL\\_ENABLE\\_Msk](#) (1UL << SysTick\_CTRL\_ENABLE\_Pos)
- #define [SysTick\\_LOAD\\_RELOAD\\_Pos](#) 0
- #define [SysTick\\_LOAD\\_RELOAD\\_Msk](#) (0xFFFFFUL << SysTick\_LOAD\_RELOAD\_Pos)
- #define [SysTick\\_VAL\\_CURRENT\\_Pos](#) 0
- #define [SysTick\\_VAL\\_CURRENT\\_Msk](#) (0xFFFFFUL << SysTick\_VAL\_CURRENT\_Pos)
- #define [SysTick\\_CALIB\\_NOREF\\_Pos](#) 31
- #define [SysTick\\_CALIB\\_NOREF\\_Msk](#) (1UL << SysTick\_CALIB\_NOREF\_Pos)
- #define [SysTick\\_CALIB\\_SKEW\\_Pos](#) 30
- #define [SysTick\\_CALIB\\_SKEW\\_Msk](#) (1UL << SysTick\_CALIB\_SKEW\_Pos)
- #define [SysTick\\_CALIB\\_TENMS\\_Pos](#) 0
- #define [SysTick\\_CALIB\\_TENMS\\_Msk](#) (0xFFFFFUL << SysTick\_VAL\_CURRENT\_Pos)
- #define [SysTick\\_CTRL\\_COUNTFLAG\\_Pos](#) 16
- #define [SysTick\\_CTRL\\_COUNTFLAG\\_Msk](#) (1UL << SysTick\_CTRL\_COUNTFLAG\_Pos)
- #define [SysTick\\_CTRL\\_CLKSOURCE\\_Pos](#) 2
- #define [SysTick\\_CTRL\\_CLKSOURCE\\_Msk](#) (1UL << SysTick\_CTRL\_CLKSOURCE\_Pos)
- #define [SysTick\\_CTRL\\_TICKINT\\_Pos](#) 1
- #define [SysTick\\_CTRL\\_TICKINT\\_Msk](#) (1UL << SysTick\_CTRL\_TICKINT\_Pos)
- #define [SysTick\\_CTRL\\_ENABLE\\_Pos](#) 0
- #define [SysTick\\_CTRL\\_ENABLE\\_Msk](#) (1UL << SysTick\_CTRL\_ENABLE\_Pos)
- #define [SysTick\\_LOAD\\_RELOAD\\_Pos](#) 0
- #define [SysTick\\_LOAD\\_RELOAD\\_Msk](#) (0xFFFFFUL << SysTick\_LOAD\_RELOAD\_Pos)
- #define [SysTick\\_VAL\\_CURRENT\\_Pos](#) 0
- #define [SysTick\\_VAL\\_CURRENT\\_Msk](#) (0xFFFFFUL << SysTick\_VAL\_CURRENT\_Pos)
- #define [SysTick\\_CALIB\\_NOREF\\_Pos](#) 31
- #define [SysTick\\_CALIB\\_NOREF\\_Msk](#) (1UL << SysTick\_CALIB\_NOREF\_Pos)
- #define [SysTick\\_CALIB\\_SKEW\\_Pos](#) 30
- #define [SysTick\\_CALIB\\_SKEW\\_Msk](#) (1UL << SysTick\_CALIB\_SKEW\_Pos)
- #define [SysTick\\_CALIB\\_TENMS\\_Pos](#) 0
- #define [SysTick\\_CALIB\\_TENMS\\_Msk](#) (0xFFFFFUL << SysTick\_VAL\_CURRENT\_Pos)

## 6.94.2 Macro Definition Documentation

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

SysTick CALIB: NOREF Mask

Definition at line 462 of file core\_cm0plus.h.

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

SysTick CALIB: NOREF Mask

Definition at line 654 of file core\_cm4.h.

### 6.94.2.3 `#define SysTick_CALIB_NOREF_Pos 31`

SysTick CALIB: NOREF Position

Definition at line 461 of file core\_cm0plus.h.

### 6.94.2.4 `#define SysTick_CALIB_NOREF_Pos 31`

SysTick CALIB: NOREF Position

Definition at line 653 of file core\_cm4.h.

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

SysTick CALIB: SKEW Mask

Definition at line 465 of file core\_cm0plus.h.

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

SysTick CALIB: SKEW Mask

Definition at line 657 of file core\_cm4.h.

### 6.94.2.7 `#define SysTick_CALIB_SKEW_Pos 30`

SysTick CALIB: SKEW Position

Definition at line 464 of file core\_cm0plus.h.

### 6.94.2.8 `#define SysTick_CALIB_SKEW_Pos 30`

SysTick CALIB: SKEW Position

Definition at line 656 of file core\_cm4.h.

### 6.94.2.9 `#define SysTick_CALIB_TENMS_Msk (0xFFFFFUL << SysTick_VAL_CURRENT_Pos)`

SysTick CALIB: TENMS Mask

Definition at line 468 of file core\_cm0plus.h.



6.94.2.10 `#define SysTick_CALIB_TENMS_Msk (0xFFFFFUL << SysTick_VAL_CURRENT_Pos)`

SysTick CALIB: TENMS Mask

Definition at line 660 of file core\_cm4.h.

6.94.2.11 `#define SysTick_CALIB_TENMS_Pos 0`

SysTick CALIB: TENMS Position

Definition at line 467 of file core\_cm0plus.h.

6.94.2.12 `#define SysTick_CALIB_TENMS_Pos 0`

SysTick CALIB: TENMS Position

Definition at line 659 of file core\_cm4.h.

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

SysTick CTRL: CLKSOURCE Mask

Definition at line 444 of file core\_cm0plus.h.

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

SysTick CTRL: CLKSOURCE Mask

Definition at line 636 of file core\_cm4.h.

6.94.2.15 `#define SysTick_CTRL_CLKSOURCE_Pos 2`

SysTick CTRL: CLKSOURCE Position

Definition at line 443 of file core\_cm0plus.h.

6.94.2.16 `#define SysTick_CTRL_CLKSOURCE_Pos 2`

SysTick CTRL: CLKSOURCE Position

Definition at line 635 of file core\_cm4.h.

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

SysTick CTRL: COUNTFLAG Mask

Definition at line 441 of file core\_cm0plus.h.

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

SysTick CTRL: COUNTFLAG Mask

Definition at line 633 of file core\_cm4.h.

**6.94.2.19 #define SysTick\_CTRL\_COUNTFLAG\_Pos 16**

SysTick CTRL: COUNTFLAG Position

Definition at line 440 of file core\_cm0plus.h.

**6.94.2.20 #define SysTick\_CTRL\_COUNTFLAG\_Pos 16**

SysTick CTRL: COUNTFLAG Position

Definition at line 632 of file core\_cm4.h.

**6.94.2.21 #define SysTick\_CTRL\_ENABLE\_Msk (1UL << SysTick\_CTRL\_ENABLE\_Pos)**

SysTick CTRL: ENABLE Mask

Definition at line 450 of file core\_cm0plus.h.

**6.94.2.22 #define SysTick\_CTRL\_ENABLE\_Msk (1UL << SysTick\_CTRL\_ENABLE\_Pos)**

SysTick CTRL: ENABLE Mask

Definition at line 642 of file core\_cm4.h.

**6.94.2.23 #define SysTick\_CTRL\_ENABLE\_Pos 0**

SysTick CTRL: ENABLE Position

Definition at line 449 of file core\_cm0plus.h.

**6.94.2.24 #define SysTick\_CTRL\_ENABLE\_Pos 0**

SysTick CTRL: ENABLE Position

Definition at line 641 of file core\_cm4.h.

**6.94.2.25 #define SysTick\_CTRL\_TICKINT\_Msk (1UL << SysTick\_CTRL\_TICKINT\_Pos)**

SysTick CTRL: TICKINT Mask

Definition at line 447 of file core\_cm0plus.h.

**6.94.2.26 #define SysTick\_CTRL\_TICKINT\_Msk (1UL << SysTick\_CTRL\_TICKINT\_Pos)**

SysTick CTRL: TICKINT Mask

Definition at line 639 of file core\_cm4.h.

**6.94.2.27 #define SysTick\_CTRL\_TICKINT\_Pos 1**

SysTick CTRL: TICKINT Position

Definition at line 446 of file core\_cm0plus.h.

**6.94.2.28** `#define SysTick_CTRL_TICKINT_Pos 1`

SysTick CTRL: TICKINT Position

Definition at line 638 of file core\_cm4.h.

**6.94.2.29** `#define SysTick_LOAD_RELOAD_Msk (0xFFFFFUL << SysTick_LOAD_RELOAD_Pos)`

SysTick LOAD: RELOAD Mask

Definition at line 454 of file core\_cm0plus.h.

**6.94.2.30** `#define SysTick_LOAD_RELOAD_Msk (0xFFFFFUL << SysTick_LOAD_RELOAD_Pos)`

SysTick LOAD: RELOAD Mask

Definition at line 646 of file core\_cm4.h.

**6.94.2.31** `#define SysTick_LOAD_RELOAD_Pos 0`

SysTick LOAD: RELOAD Position

Definition at line 453 of file core\_cm0plus.h.

**6.94.2.32** `#define SysTick_LOAD_RELOAD_Pos 0`

SysTick LOAD: RELOAD Position

Definition at line 645 of file core\_cm4.h.

**6.94.2.33** `#define SysTick_VAL_CURRENT_Msk (0xFFFFFUL << SysTick_VAL_CURRENT_Pos)`

SysTick VAL: CURRENT Mask

Definition at line 458 of file core\_cm0plus.h.

**6.94.2.34** `#define SysTick_VAL_CURRENT_Msk (0xFFFFFUL << SysTick_VAL_CURRENT_Pos)`

SysTick VAL: CURRENT Mask

Definition at line 650 of file core\_cm4.h.

**6.94.2.35** `#define SysTick_VAL_CURRENT_Pos 0`

SysTick VAL: CURRENT Position

Definition at line 457 of file core\_cm0plus.h.

**6.94.2.36** `#define SysTick_VAL_CURRENT_Pos 0`

SysTick VAL: CURRENT Position

Definition at line 649 of file core\_cm4.h.

## 6.95 Trace Port Interface (TPI)

### 6.95.1 Detailed Description

Type definitions for the Trace Port Interface (TPI)

#### Data Structures

- struct [TPI\\_Type](#)

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

#### Macros

- `#define TPI_ACPR_PRESCALER_Pos 0`
- `#define TPI_ACPR_PRESCALER_Msk (0x1FFFUL << TPI_ACPR_PRESCALER_Pos)`
- `#define TPI_SPPR_TXMODE_Pos 0`
- `#define TPI_SPPR_TXMODE_Msk (0x3UL << TPI_SPPR_TXMODE_Pos)`
- `#define TPI_FFSR_FtNonStop_Pos 3`
- `#define TPI_FFSR_FtNonStop_Msk (0x1UL << TPI_FFSR_FtNonStop_Pos)`
- `#define TPI_FFSR_TCPresent_Pos 2`
- `#define TPI_FFSR_TCPresent_Msk (0x1UL << TPI_FFSR_TCPresent_Pos)`
- `#define TPI_FFSR_FtStopped_Pos 1`
- `#define TPI_FFSR_FtStopped_Msk (0x1UL << TPI_FFSR_FtStopped_Pos)`
- `#define TPI_FFSR_FlInProg_Pos 0`
- `#define TPI_FFSR_FlInProg_Msk (0x1UL << TPI_FFSR_FlInProg_Pos)`
- `#define TPI_FFCR_TrigIn_Pos 8`
- `#define TPI_FFCR_TrigIn_Msk (0x1UL << TPI_FFCR_TrigIn_Pos)`
- `#define TPI_FFCR_EnFCont_Pos 1`
- `#define TPI_FFCR_EnFCont_Msk (0x1UL << TPI_FFCR_EnFCont_Pos)`
- `#define TPI_TRIGGER_TRIGGER_Pos 0`
- `#define TPI_TRIGGER_TRIGGER_Msk (0x1UL << TPI_TRIGGER_TRIGGER_Pos)`
- `#define TPI_FIFO0_ITM_ATVALID_Pos 29`
- `#define TPI_FIFO0_ITM_ATVALID_Msk (0x3UL << TPI_FIFO0_ITM_ATVALID_Pos)`
- `#define TPI_FIFO0_ITM_bytecount_Pos 27`
- `#define TPI_FIFO0_ITM_bytecount_Msk (0x3UL << TPI_FIFO0_ITM_bytecount_Pos)`
- `#define TPI_FIFO0_ETM_ATVALID_Pos 26`
- `#define TPI_FIFO0_ETM_ATVALID_Msk (0x3UL << TPI_FIFO0_ETM_ATVALID_Pos)`
- `#define TPI_FIFO0_ETM_bytecount_Pos 24`
- `#define TPI_FIFO0_ETM_bytecount_Msk (0x3UL << TPI_FIFO0_ETM_bytecount_Pos)`
- `#define TPI_FIFO0_ETM2_Pos 16`
- `#define TPI_FIFO0_ETM2_Msk (0xFFUL << TPI_FIFO0_ETM2_Pos)`
- `#define TPI_FIFO0_ETM1_Pos 8`
- `#define TPI_FIFO0_ETM1_Msk (0xFFUL << TPI_FIFO0_ETM1_Pos)`
- `#define TPI_FIFO0_ETM0_Pos 0`
- `#define TPI_FIFO0_ETM0_Msk (0xFFUL << TPI_FIFO0_ETM0_Pos)`
- `#define TPI_ITATBCTR2_ATREADY_Pos 0`
- `#define TPI_ITATBCTR2_ATREADY_Msk (0x1UL << TPI_ITATBCTR2_ATREADY_Pos)`
- `#define TPI_FIFO1_ITM_ATVALID_Pos 29`
- `#define TPI_FIFO1_ITM_ATVALID_Msk (0x3UL << TPI_FIFO1_ITM_ATVALID_Pos)`
- `#define TPI_FIFO1_ITM_bytecount_Pos 27`
- `#define TPI_FIFO1_ITM_bytecount_Msk (0x3UL << TPI_FIFO1_ITM_bytecount_Pos)`
- `#define TPI_FIFO1_ETM_ATVALID_Pos 26`
- `#define TPI_FIFO1_ETM_ATVALID_Msk (0x3UL << TPI_FIFO1_ETM_ATVALID_Pos)`

- `#define TPI_FIFO1_ETM_bytecount_Pos 24`
- `#define TPI_FIFO1_ETM_bytecount_Msk (0x3UL << TPI_FIFO1_ETM_bytecount_Pos)`
- `#define TPI_FIFO1_ITM2_Pos 16`
- `#define TPI_FIFO1_ITM2_Msk (0xFFUL << TPI_FIFO1_ITM2_Pos)`
- `#define TPI_FIFO1_ITM1_Pos 8`
- `#define TPI_FIFO1_ITM1_Msk (0xFFUL << TPI_FIFO1_ITM1_Pos)`
- `#define TPI_FIFO1_ITM0_Pos 0`
- `#define TPI_FIFO1_ITM0_Msk (0xFFUL << TPI_FIFO1_ITM0_Pos)`
- `#define TPI_ITATBCTR0_ATREADY_Pos 0`
- `#define TPI_ITATBCTR0_ATREADY_Msk (0x1UL << TPI_ITATBCTR0_ATREADY_Pos)`
- `#define TPI_ITCTRL_Mode_Pos 0`
- `#define TPI_ITCTRL_Mode_Msk (0x1UL << TPI_ITCTRL_Mode_Pos)`
- `#define TPI_DEVID_NRZVALID_Pos 11`
- `#define TPI_DEVID_NRZVALID_Msk (0x1UL << TPI_DEVID_NRZVALID_Pos)`
- `#define TPI_DEVID_MANCVALID_Pos 10`
- `#define TPI_DEVID_MANCVALID_Msk (0x1UL << TPI_DEVID_MANCVALID_Pos)`
- `#define TPI_DEVID_PTINVALID_Pos 9`
- `#define TPI_DEVID_PTINVALID_Msk (0x1UL << TPI_DEVID_PTINVALID_Pos)`
- `#define TPI_DEVID_MinBufSz_Pos 6`
- `#define TPI_DEVID_MinBufSz_Msk (0x7UL << TPI_DEVID_MinBufSz_Pos)`
- `#define TPI_DEVID_AsynClkIn_Pos 5`
- `#define TPI_DEVID_AsynClkIn_Msk (0x1UL << TPI_DEVID_AsynClkIn_Pos)`
- `#define TPI_DEVID_NrTraceInput_Pos 0`
- `#define TPI_DEVID_NrTraceInput_Msk (0x1FUL << TPI_DEVID_NrTraceInput_Pos)`
- `#define TPI_DEVTYPE_SubType_Pos 0`
- `#define TPI_DEVTYPE_SubType_Msk (0xFUL << TPI_DEVTYPE_SubType_Pos)`
- `#define TPI_DEVTYPE_MajorType_Pos 4`
- `#define TPI_DEVTYPE_MajorType_Msk (0xFUL << TPI_DEVTYPE_MajorType_Pos)`

## 6.95.2 Macro Definition Documentation

### 6.95.2.1 `#define TPI_ACPR_PRESCALER_Msk (0x1FFFUL << TPI_ACPR_PRESCALER_Pos)`

TPI ACPR: PRESCALER Mask

Definition at line 949 of file core\_cm4.h.

### 6.95.2.2 `#define TPI_ACPR_PRESCALER_Pos 0`

TPI ACPR: PRESCALER Position

Definition at line 948 of file core\_cm4.h.

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

TPI DEVID: AsynClkIn Mask

Definition at line 1049 of file core\_cm4.h.

### 6.95.2.4 `#define TPI_DEVID_AsynClkIn_Pos 5`

TPI DEVID: AsynClkIn Position

Definition at line 1048 of file core\_cm4.h.

**6.95.2.5   #define TPI\_DEVID\_MANCVALID\_Msk (0x1UL << TPI\_DEVID\_MANCVALID\_Pos)**

TPI DEVID: MANCVALID Mask

Definition at line 1040 of file core\_cm4.h.

**6.95.2.6   #define TPI\_DEVID\_MANCVALID\_Pos 10**

TPI DEVID: MANCVALID Position

Definition at line 1039 of file core\_cm4.h.

**6.95.2.7   #define TPI\_DEVID\_MinBufSz\_Msk (0x7UL << TPI\_DEVID\_MinBufSz\_Pos)**

TPI DEVID: MinBufSz Mask

Definition at line 1046 of file core\_cm4.h.

**6.95.2.8   #define TPI\_DEVID\_MinBufSz\_Pos 6**

TPI DEVID: MinBufSz Position

Definition at line 1045 of file core\_cm4.h.

**6.95.2.9   #define TPI\_DEVID\_NrTraceInput\_Msk (0x1FUL << TPI\_DEVID\_NrTraceInput\_Pos)**

TPI DEVID: NrTraceInput Mask

Definition at line 1052 of file core\_cm4.h.

**6.95.2.10   #define TPI\_DEVID\_NrTraceInput\_Pos 0**

TPI DEVID: NrTraceInput Position

Definition at line 1051 of file core\_cm4.h.

**6.95.2.11   #define TPI\_DEVID\_NRZVALID\_Msk (0x1UL << TPI\_DEVID\_NRZVALID\_Pos)**

TPI DEVID: NRZVALID Mask

Definition at line 1037 of file core\_cm4.h.

**6.95.2.12   #define TPI\_DEVID\_NRZVALID\_Pos 11**

TPI DEVID: NRZVALID Position

Definition at line 1036 of file core\_cm4.h.

**6.95.2.13   #define TPI\_DEVID\_PTINVALID\_Msk (0x1UL << TPI\_DEVID\_PTINVALID\_Pos)**

TPI DEVID: PTINVALID Mask

Definition at line 1043 of file core\_cm4.h.

**6.95.2.14 #define TPI\_DEVID\_PTINVALID\_Pos 9**

TPI DEVID: PTINVALID Position

Definition at line 1042 of file core\_cm4.h.

**6.95.2.15 #define TPI\_DEVTYPE\_MajorType\_Msk (0xFUL << TPI\_DEVTYPE\_MajorType\_Pos)**

TPI DEVTYPE: MajorType Mask

Definition at line 1059 of file core\_cm4.h.

**6.95.2.16 #define TPI\_DEVTYPE\_MajorType\_Pos 4**

TPI DEVTYPE: MajorType Position

Definition at line 1058 of file core\_cm4.h.

**6.95.2.17 #define TPI\_DEVTYPE\_SubType\_Msk (0xFUL << TPI\_DEVTYPE\_SubType\_Pos)**

TPI DEVTYPE: SubType Mask

Definition at line 1056 of file core\_cm4.h.

**6.95.2.18 #define TPI\_DEVTYPE\_SubType\_Pos 0**

TPI DEVTYPE: SubType Position

Definition at line 1055 of file core\_cm4.h.

**6.95.2.19 #define TPI\_FFCR\_EnFCont\_Msk (0x1UL << TPI\_FFCR\_EnFCont\_Pos)**

TPI FFCR: EnFCont Mask

Definition at line 973 of file core\_cm4.h.

**6.95.2.20 #define TPI\_FFCR\_EnFCont\_Pos 1**

TPI FFCR: EnFCont Position

Definition at line 972 of file core\_cm4.h.

**6.95.2.21 #define TPI\_FFCR\_TrigIn\_Msk (0x1UL << TPI\_FFCR\_TrigIn\_Pos)**

TPI FFCR: TrigIn Mask

Definition at line 970 of file core\_cm4.h.

**6.95.2.22 #define TPI\_FFCR\_TrigIn\_Pos 8**

TPI FFCR: TrigIn Position

Definition at line 969 of file core\_cm4.h.

6.95.2.23 `#define TPI_FFSR_FInProg_Msk (0x1UL << TPI_FFSR_FInProg_Pos)`

TPI FFSR: FInProg Mask

Definition at line 966 of file core\_cm4.h.

6.95.2.24 `#define TPI_FFSR_FInProg_Pos 0`

TPI FFSR: FInProg Position

Definition at line 965 of file core\_cm4.h.

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

TPI FFSR: FtNonStop Mask

Definition at line 957 of file core\_cm4.h.

6.95.2.26 `#define TPI_FFSR_FtNonStop_Pos 3`

TPI FFSR: FtNonStop Position

Definition at line 956 of file core\_cm4.h.

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

TPI FFSR: FtStopped Mask

Definition at line 963 of file core\_cm4.h.

6.95.2.28 `#define TPI_FFSR_FtStopped_Pos 1`

TPI FFSR: FtStopped Position

Definition at line 962 of file core\_cm4.h.

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

TPI FFSR: TCPresent Mask

Definition at line 960 of file core\_cm4.h.

6.95.2.30 `#define TPI_FFSR_TCPresent_Pos 2`

TPI FFSR: TCPresent Position

Definition at line 959 of file core\_cm4.h.

6.95.2.31 `#define TPI_FIFO0_ETM0_Msk (0xFFUL << TPI_FIFO0_ETM0_Pos)`

TPI FIFO0: ETM0 Mask

Definition at line 999 of file core\_cm4.h.



6.95.2.32 `#define TPI_FIFO0_ETM0_Pos 0`

TPI FIFO0: ETM0 Position

Definition at line 998 of file core\_cm4.h.

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

TPI FIFO0: ETM1 Mask

Definition at line 996 of file core\_cm4.h.

6.95.2.34 `#define TPI_FIFO0_ETM1_Pos 8`

TPI FIFO0: ETM1 Position

Definition at line 995 of file core\_cm4.h.

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

TPI FIFO0: ETM2 Mask

Definition at line 993 of file core\_cm4.h.

6.95.2.36 `#define TPI_FIFO0_ETM2_Pos 16`

TPI FIFO0: ETM2 Position

Definition at line 992 of file core\_cm4.h.

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

TPI FIFO0: ETM\_ATVALID Mask

Definition at line 987 of file core\_cm4.h.

6.95.2.38 `#define TPI_FIFO0_ETM_ATVALID_Pos 26`

TPI FIFO0: ETM\_ATVALID Position

Definition at line 986 of file core\_cm4.h.

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

TPI FIFO0: ETM\_bytecount Mask

Definition at line 990 of file core\_cm4.h.

6.95.2.40 `#define TPI_FIFO0_ETM_bytecount_Pos 24`

TPI FIFO0: ETM\_bytecount Position

Definition at line 989 of file core\_cm4.h.

6.95.2.41 **#define TPI\_FIFO0\_ITM\_ATVALID\_Msk** (0x3UL << TPI\_FIFO0\_ITM\_ATVALID\_Pos)

TPI FIFO0: ITM\_ATVALID Mask

Definition at line 981 of file core\_cm4.h.

6.95.2.42 **#define TPI\_FIFO0\_ITM\_ATVALID\_Pos** 29

TPI FIFO0: ITM\_ATVALID Position

Definition at line 980 of file core\_cm4.h.

6.95.2.43 **#define TPI\_FIFO0\_ITM\_bytecount\_Msk** (0x3UL << TPI\_FIFO0\_ITM\_bytecount\_Pos)

TPI FIFO0: ITM\_bytecount Mask

Definition at line 984 of file core\_cm4.h.

6.95.2.44 **#define TPI\_FIFO0\_ITM\_bytecount\_Pos** 27

TPI FIFO0: ITM\_bytecount Position

Definition at line 983 of file core\_cm4.h.

6.95.2.45 **#define TPI\_FIFO1\_ETM\_ATVALID\_Msk** (0x3UL << TPI\_FIFO1\_ETM\_ATVALID\_Pos)

TPI FIFO1: ETM\_ATVALID Mask

Definition at line 1013 of file core\_cm4.h.

6.95.2.46 **#define TPI\_FIFO1\_ETM\_ATVALID\_Pos** 26

TPI FIFO1: ETM\_ATVALID Position

Definition at line 1012 of file core\_cm4.h.

6.95.2.47 **#define TPI\_FIFO1\_ETM\_bytecount\_Msk** (0x3UL << TPI\_FIFO1\_ETM\_bytecount\_Pos)

TPI FIFO1: ETM\_bytecount Mask

Definition at line 1016 of file core\_cm4.h.

6.95.2.48 **#define TPI\_FIFO1\_ETM\_bytecount\_Pos** 24

TPI FIFO1: ETM\_bytecount Position

Definition at line 1015 of file core\_cm4.h.

6.95.2.49 **#define TPI\_FIFO1\_ITM0\_Msk** (0xFFUL << TPI\_FIFO1\_ITM0\_Pos)

TPI FIFO1: ITM0 Mask

Definition at line 1025 of file core\_cm4.h.

6.95.2.50 `#define TPI_FIFO1_ITM0_Pos 0`

TPI FIFO1: ITM0 Position

Definition at line 1024 of file core\_cm4.h.

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

TPI FIFO1: ITM1 Mask

Definition at line 1022 of file core\_cm4.h.

6.95.2.52 `#define TPI_FIFO1_ITM1_Pos 8`

TPI FIFO1: ITM1 Position

Definition at line 1021 of file core\_cm4.h.

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

TPI FIFO1: ITM2 Mask

Definition at line 1019 of file core\_cm4.h.

6.95.2.54 `#define TPI_FIFO1_ITM2_Pos 16`

TPI FIFO1: ITM2 Position

Definition at line 1018 of file core\_cm4.h.

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

TPI FIFO1: ITM\_ATVALID Mask

Definition at line 1007 of file core\_cm4.h.

6.95.2.56 `#define TPI_FIFO1_ITM_ATVALID_Pos 29`

TPI FIFO1: ITM\_ATVALID Position

Definition at line 1006 of file core\_cm4.h.

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

TPI FIFO1: ITM\_bytecount Mask

Definition at line 1010 of file core\_cm4.h.

6.95.2.58 `#define TPI_FIFO1_ITM_bytecount_Pos 27`

TPI FIFO1: ITM\_bytecount Position

Definition at line 1009 of file core\_cm4.h.

6.95.2.59 `#define TPI_ITATBCTR0_ATREADY_Msk (0x1UL << TPI_ITATBCTR0_ATREADY_Pos)`

TPI ITATBCTR0: ATREADY Mask

Definition at line 1029 of file core\_cm4.h.

6.95.2.60 `#define TPI_ITATBCTR0_ATREADY_Pos 0`

TPI ITATBCTR0: ATREADY Position

Definition at line 1028 of file core\_cm4.h.

6.95.2.61 `#define TPI_ITATBCTR2_ATREADY_Msk (0x1UL << TPI_ITATBCTR2_ATREADY_Pos)`

TPI ITATBCTR2: ATREADY Mask

Definition at line 1003 of file core\_cm4.h.

6.95.2.62 `#define TPI_ITATBCTR2_ATREADY_Pos 0`

TPI ITATBCTR2: ATREADY Position

Definition at line 1002 of file core\_cm4.h.

6.95.2.63 `#define TPI_ITCTRL_Mode_Msk (0x1UL << TPI_ITCTRL_Mode_Pos)`

TPI ITCTRL: Mode Mask

Definition at line 1033 of file core\_cm4.h.

6.95.2.64 `#define TPI_ITCTRL_Mode_Pos 0`

TPI ITCTRL: Mode Position

Definition at line 1032 of file core\_cm4.h.

6.95.2.65 `#define TPI_SPPR_TXMODE_Msk (0x3UL << TPI_SPPR_TXMODE_Pos)`

TPI SPPR: TXMODE Mask

Definition at line 953 of file core\_cm4.h.

6.95.2.66 `#define TPI_SPPR_TXMODE_Pos 0`

TPI SPPR: TXMODE Position

Definition at line 952 of file core\_cm4.h.

6.95.2.67 `#define TPI_TRIGGER_TRIGGER_Msk (0x1UL << TPI_TRIGGER_TRIGGER_Pos)`

TPI TRIGGER: TRIGGER Mask

Definition at line 977 of file core\_cm4.h.

6.95.2.68 `#define TPI_TRIGGER_TRIGGER_Pos 0`

TPI TRIGGER: TRIGGER Position

Definition at line 976 of file core\_cm4.h.

## 6.96 USBD\_Core

### 6.96.1 Detailed Description

#### Data Structures

- struct [WB\\_T](#)
- union [WORD\\_BYTE](#)
- struct [BM\\_T](#)
- union [REQUEST\\_TYPE](#)
- struct [USB\\_SETUP\\_PACKET](#)
- struct [USB\\_DEVICE\\_DESCRIPTOR](#)
- struct [USB\\_DEVICE\\_QUALIFIER\\_DESCRIPTOR](#)
- struct [USB\\_CONFIGURATION\\_DESCRIPTOR](#)
- struct [USB\\_IAD\\_DESCRIPTOR](#)
- struct [USB\\_INTERFACE\\_DESCRIPTOR](#)
- struct [USB\\_ENDPOINT\\_DESCRIPTOR](#)
- struct [USB\\_STRING\\_DESCRIPTOR](#)
- struct [USB\\_COMMON\\_DESCRIPTOR](#)
- struct [USB\\_OTHER\\_SPEED\\_CONFIGURATION](#)

#### Macros

- `#define USB_CONFIG_POWER_MA(mA) ((mA)/2)`
- `#define USB_ENDPOINT_0_HS_MAXP 64`
- `#define USB_ENDPOINT_0_LS_MAXP 8`
- `#define USB_ENDPOINT_BULK_HS_MAXP 512`
- `#define WBVAL(x) ((x) & 0xFF),(((x) >> 8) & 0xFF)`
- `#define B3VAL(x) ((x) & 0xFF),(((x) >> 8) & 0xFF),(((x) >> 16) & 0xFF)`
- `#define USB_DEVICE_DESC_SIZE (sizeof(USB_DEVICE_DESCRIPTOR))`
- `#define USB_CONFIGURATION_DESC_SIZE (sizeof(USB_CONFIGURATION_DESCRIPTOR))`
- `#define USB_INTERFACE_DESC_SIZE (sizeof(USB_INTERFACE_DESCRIPTOR))`
- `#define USB_INTERFACE_ASSOC_DESC_SIZE (sizeof(USB_IAD_DESCRIPTOR))`
- `#define USB_ENDPOINT_DESC_SIZE (sizeof(USB_ENDPOINT_DESCRIPTOR))`
- `#define USB_DEVICE_QUALI_SIZE (sizeof(USB_DEVICE_QUALIFIER_DESCRIPTOR))`
- `#define USB_OTHER_SPEED_CONF_SIZE (sizeof(USB_OTHER_SPEED_CONFIGURATION))`

#### Typedefs

- typedef void \* [USB\\_HANDLE\\_T](#)

### 6.96.2 Macro Definition Documentation

#### 6.96.2.1 `#define B3VAL( x ) ((x) & 0xFF),(((x) >> 8) & 0xFF),(((x) >> 16) & 0xFF)`

Definition at line 693 of file usbd.h.

#### 6.96.2.2 `#define REQUEST_CLASS 1`

Class Request

Definition at line 110 of file usbd.h.

**6.96.2.3 #define REQUEST\_DEVICE\_TO\_HOST 1**

Request from device to host

Definition at line 101 of file usbd.h.

**6.96.2.4 #define REQUEST\_HOST\_TO\_DEVICE 0**

bmRequestType.Dir defines Request from host to device

Definition at line 99 of file usbd.h.

**6.96.2.5 #define REQUEST\_RESERVED 3**

Reserved Request

Definition at line 114 of file usbd.h.

**6.96.2.6 #define REQUEST\_STANDARD 0**

bmRequestType.Type defines Standard Request

Definition at line 108 of file usbd.h.

**6.96.2.7 #define REQUEST\_TO\_DEVICE 0**

bmRequestType.Recipient defines Request to device

Definition at line 121 of file usbd.h.

**6.96.2.8 #define REQUEST\_TO\_ENDPOINT 2**

Request to endpoint

Definition at line 125 of file usbd.h.

**6.96.2.9 #define REQUEST\_TO\_INTERFACE 1**

Request to interface

Definition at line 123 of file usbd.h.

**6.96.2.10 #define REQUEST\_TO\_OTHER 3**

Request to other

Definition at line 127 of file usbd.h.

**6.96.2.11 #define REQUEST\_VENDOR 2**

Vendor Request

Definition at line 112 of file usbd.h.

**6.96.2.12** `#define USB_CONFIG_BUS_POWERED 0x80`

Bus powered

Definition at line 288 of file usbd.h.

**6.96.2.13** `#define USB_CONFIG_POWER_MA( mA ) ((mA)/2)`

bMaxPower in Configuration Descriptor

Definition at line 296 of file usbd.h.

**6.96.2.14** `#define USB_CONFIG_POWERED_MASK 0x40`

bmAttributes in Configuration Descriptor Power field mask

Definition at line 286 of file usbd.h.

**6.96.2.15** `#define USB_CONFIG_REMOTE_WAKEUP 0x20`

remote wakeup

Definition at line 292 of file usbd.h.

**6.96.2.16** `#define USB_CONFIG_SELF_POWERED 0xC0`

Self powered

Definition at line 290 of file usbd.h.

**6.96.2.17** `#define USB_CONFIGURATION_DESC_SIZE (sizeof(USB_CONFIGURATION_DESCRIPTOR))`

Definition at line 696 of file usbd.h.

**6.96.2.18** `#define USB_CONFIGURATION_DESCRIPTOR_TYPE 2`

Configuration descriptor type

Definition at line 230 of file usbd.h.

**6.96.2.19** `#define USB_DEBUG_DESCRIPTOR_TYPE 10`

Debug descriptor type

Definition at line 246 of file usbd.h.

**6.96.2.20** `#define USB_DEVICE_CLASS_APP 0xFE`

Application device class

Definition at line 277 of file usbd.h.



**6.96.2.21 #define USB\_DEVICE\_CLASS\_AUDIO 0x01**

Audio device class

Definition at line 257 of file usbd.h.

**6.96.2.22 #define USB\_DEVICE\_CLASS\_COMMUNICATIONS 0x02**

Communications device class

Definition at line 259 of file usbd.h.

**6.96.2.23 #define USB\_DEVICE\_CLASS\_HUB 0x09**

Hub device class

Definition at line 273 of file usbd.h.

**6.96.2.24 #define USB\_DEVICE\_CLASS\_HUMAN\_INTERFACE 0x03**

Human interface device class

Definition at line 261 of file usbd.h.

**6.96.2.25 #define USB\_DEVICE\_CLASS\_MISCELLANEOUS 0xEF**

miscellaneous device class

Definition at line 275 of file usbd.h.

**6.96.2.26 #define USB\_DEVICE\_CLASS\_MONITOR 0x04**

monitor device class

Definition at line 263 of file usbd.h.

**6.96.2.27 #define USB\_DEVICE\_CLASS\_PHYSICAL\_INTERFACE 0x05**

physical interface device class

Definition at line 265 of file usbd.h.

**6.96.2.28 #define USB\_DEVICE\_CLASS\_POWER 0x06**

power device class

Definition at line 267 of file usbd.h.

**6.96.2.29 #define USB\_DEVICE\_CLASS\_PRINTER 0x07**

Printer device class

Definition at line 269 of file usbd.h.

**6.96.2.30 #define USB\_DEVICE\_CLASS\_RESERVED 0x00**

USB Device Classes Reserved device class

Definition at line 255 of file usbd.h.

**6.96.2.31 #define USB\_DEVICE\_CLASS\_STORAGE 0x08**

Storage device class

Definition at line 271 of file usbd.h.

**6.96.2.32 #define USB\_DEVICE\_CLASS\_VENDOR\_SPECIFIC 0xFF**

Vendor specific device class

Definition at line 279 of file usbd.h.

**6.96.2.33 #define USB\_DEVICE\_DESC\_SIZE (sizeof(USB\_DEVICE\_DESCRIPTOR))**

Definition at line 695 of file usbd.h.

**6.96.2.34 #define USB\_DEVICE\_DESCRIPTOR\_TYPE 1**

USB Descriptor Types Device descriptor type

Definition at line 228 of file usbd.h.

**6.96.2.35 #define USB\_DEVICE\_QUALI\_SIZE (sizeof(USB\_DEVICE\_QUALIFIER\_DESCRIPTOR))**

Definition at line 700 of file usbd.h.

**6.96.2.36 #define USB\_DEVICE\_QUALIFIER\_DESCRIPTOR\_TYPE 6**

Device qualifier descriptor type

Definition at line 238 of file usbd.h.

**6.96.2.37 #define USB\_ENDPOINT\_0\_HS\_MAXP 64**

Control endpoint EP0's maximum packet size in high-speed mode.

Definition at line 345 of file usbd.h.

**6.96.2.38 #define USB\_ENDPOINT\_0\_LS\_MAXP 8**

Control endpoint EP0's maximum packet size in low-speed mode.

Definition at line 347 of file usbd.h.

**6.96.2.39 #define USB\_ENDPOINT\_BULK\_HS\_MAXP 512**

Bulk endpoint's maximum packet size in high-speed mode.

Definition at line 349 of file usbd.h.

**6.96.2.40   #define USB\_ENDPOINT\_DESC\_SIZE (sizeof(USB\_ENDPOINT\_DESCRIPTOR))**

Definition at line 699 of file usbd.h.

**6.96.2.41   #define USB\_ENDPOINT\_DESCRIPTOR\_TYPE 5**

Endpoint descriptor type

Definition at line 236 of file usbd.h.

**6.96.2.42   #define USB\_ENDPOINT\_DIRECTION\_MASK 0x80**

bEndpointAddress in Endpoint Descriptor Endopint address mask

Definition at line 302 of file usbd.h.

**6.96.2.43   #define USB\_ENDPOINT\_IN( addr ) ((addr) | 0x80)**

Macro to convert IN endopint number to endpoint address value.

Definition at line 306 of file usbd.h.

**6.96.2.44   #define USB\_ENDPOINT\_OUT( addr ) ((addr) | 0x00)**

Macro to convert OUT endopint number to endpoint address value.

Definition at line 304 of file usbd.h.

**6.96.2.45   #define USB\_ENDPOINT\_SYNC\_ADAPTIVE 0x08**

Adaptive sync Endopint

Definition at line 329 of file usbd.h.

**6.96.2.46   #define USB\_ENDPOINT\_SYNC\_ASYNCHRONOUS 0x04**

Asynchronous sync Endopint

Definition at line 327 of file usbd.h.

**6.96.2.47   #define USB\_ENDPOINT\_SYNC\_MASK 0x0C**

Endopint sync type mask

Definition at line 323 of file usbd.h.

**6.96.2.48   #define USB\_ENDPOINT\_SYNC\_NO\_SYNCHRONIZATION 0x00**

no synchronization Endopint

Definition at line 325 of file usbd.h.

**6.96.2.49** `#define USB_ENDPOINT_SYNC_SYNCHRONOUS 0x0C`

Synchronous sync Endopint

Definition at line 331 of file usbd.h.

**6.96.2.50** `#define USB_ENDPOINT_TYPE_BULK 0x02`

bulk Endopint type

Definition at line 319 of file usbd.h.

**6.96.2.51** `#define USB_ENDPOINT_TYPE_CONTROL 0x00`

Control Endopint type

Definition at line 315 of file usbd.h.

**6.96.2.52** `#define USB_ENDPOINT_TYPE_INTERRUPT 0x03`

interrupt Endopint type

Definition at line 321 of file usbd.h.

**6.96.2.53** `#define USB_ENDPOINT_TYPE_ISOCHRONOUS 0x01`

isochronous Endopint type

Definition at line 317 of file usbd.h.

**6.96.2.54** `#define USB_ENDPOINT_TYPE_MASK 0x03`

bmAttributes in Endpoint Descriptor Endopint type mask

Definition at line 313 of file usbd.h.

**6.96.2.55** `#define USB_ENDPOINT_USAGE_DATA 0x00`

Endopint data usage type

Definition at line 335 of file usbd.h.

**6.96.2.56** `#define USB_ENDPOINT_USAGE_FEEDBACK 0x10`

Endopint feedback usage type

Definition at line 337 of file usbd.h.

**6.96.2.57** `#define USB_ENDPOINT_USAGE_IMPLICIT_FEEDBACK 0x20`

Endopint implicit feedback usage type

Definition at line 339 of file usbd.h.

**6.96.2.58 #define USB\_ENDPOINT\_USAGE\_MASK 0x30**

Endpoint usage type mask

Definition at line 333 of file usbd.h.

**6.96.2.59 #define USB\_ENDPOINT\_USAGE\_RESERVED 0x30**

Endpoint reserved usage type

Definition at line 341 of file usbd.h.

**6.96.2.60 #define USB\_FEATURE\_ENDPOINT\_STALL 0**

USB Standard Feature selectors ENDPOINT\_STALL feature

Definition at line 191 of file usbd.h.

**6.96.2.61 #define USB\_FEATURE\_REMOTE\_WAKEUP 1**

REMOTE\_WAKEUP feature

Definition at line 193 of file usbd.h.

**6.96.2.62 #define USB\_FEATURE\_TEST\_MODE 2**

TEST\_MODE feature

Definition at line 195 of file usbd.h.

**6.96.2.63 #define USB\_GETSTATUS\_ENDPOINT\_STALL 0x01**

ENDPOINT\_STALL status

Definition at line 184 of file usbd.h.

**6.96.2.64 #define USB\_GETSTATUS\_REMOTE\_WAKEUP 0x02**

REMOTE\_WAKEUP capable status

Definition at line 182 of file usbd.h.

**6.96.2.65 #define USB\_GETSTATUS\_SELF\_POWERED 0x01**

USB GET\_STATUS Bit Values SELF\_POWERED status

Definition at line 180 of file usbd.h.

**6.96.2.66 #define USB\_INTERFACE\_ASSOC\_DESC\_SIZE (sizeof(USB\_IAD\_DESCRIPTOR))**

Definition at line 698 of file usbd.h.

**6.96.2.67    #define USB\_INTERFACE\_ASSOCIATION\_DESCRIPTOR\_TYPE 11**

Interface association descriptor type

Definition at line 248 of file usbd.h.

**6.96.2.68    #define USB\_INTERFACE\_DESC\_SIZE (sizeof(USB\_INTERFACE\_DESCRIPTOR))**

Definition at line 697 of file usbd.h.

**6.96.2.69    #define USB\_INTERFACE\_DESCRIPTOR\_TYPE 4**

Interface descriptor type

Definition at line 234 of file usbd.h.

**6.96.2.70    #define USB\_INTERFACE\_POWER\_DESCRIPTOR\_TYPE 8**

Interface power descriptor type

Definition at line 242 of file usbd.h.

**6.96.2.71    #define USB\_OTG\_DESCRIPTOR\_TYPE 9**

OTG descriptor type

Definition at line 244 of file usbd.h.

**6.96.2.72    #define USB\_OTHER\_SPEED\_CONF\_SIZE (sizeof(USB\_OTHER\_SPEED\_CONFIGURATION))**

Definition at line 701 of file usbd.h.

**6.96.2.73    #define USB\_OTHER\_SPEED\_CONFIG\_DESCRIPTOR\_TYPE 7**

Other speed configuration descriptor type

Definition at line 240 of file usbd.h.

**6.96.2.74    #define USB\_REQUEST\_CLEAR\_FEATURE 1**

CLEAR\_FEATURE request

Definition at line 155 of file usbd.h.

**6.96.2.75    #define USB\_REQUEST\_GET\_CONFIGURATION 8**

GET\_CONFIGURATION request

Definition at line 165 of file usbd.h.

**6.96.2.76    #define USB\_REQUEST\_GET\_DESCRIPTOR 6**

GET\_DESCRIPTOR request

Definition at line 161 of file usbd.h.

6.96.2.77 **#define USB\_REQUEST\_GET\_INTERFACE 10**

GET\_INTERFACE request

Definition at line 169 of file usbd.h.

6.96.2.78 **#define USB\_REQUEST\_GET\_STATUS 0**

USB Standard Request Codes GET\_STATUS request

Definition at line 153 of file usbd.h.

6.96.2.79 **#define USB\_REQUEST\_SET\_ADDRESS 5**

SET\_ADDRESS request

Definition at line 159 of file usbd.h.

6.96.2.80 **#define USB\_REQUEST\_SET\_CONFIGURATION 9**

SET\_CONFIGURATION request

Definition at line 167 of file usbd.h.

6.96.2.81 **#define USB\_REQUEST\_SET\_DESCRIPTOR 7**

SET\_DESCRIPTOR request

Definition at line 163 of file usbd.h.

6.96.2.82 **#define USB\_REQUEST\_SET\_FEATURE 3**

SET\_FEATURE request

Definition at line 157 of file usbd.h.

6.96.2.83 **#define USB\_REQUEST\_SET\_INTERFACE 11**

SET\_INTERFACE request

Definition at line 171 of file usbd.h.

6.96.2.84 **#define USB\_REQUEST\_SYNC\_FRAME 12**

SYNC\_FRAME request

Definition at line 173 of file usbd.h.

6.96.2.85 **#define USB\_STRING\_DESCRIPTOR\_TYPE 3**

String descriptor type

Definition at line 232 of file usbd.h.

6.96.2.86 `#define WBVAL( x ) ((x) & 0xFF),(((x) >> 8) & 0xFF)`

Definition at line 692 of file usbd.h.

### 6.96.3 Typedef Documentation

6.96.3.1 `typedef void* USBD_HANDLE_T`

USB device stack/module handle.

Definition at line 690 of file usbd.h.



## 6.97 emWin download and installation

The emWin library is not included with LPCOpen. It is available as a separate ZIP file download and installs right into the LPCOpen directory tree for Keil and IAR projects or into the LPCXpresso workspace for LPCXpresso projects. To download emWin, go to the emWin project page at:

[emWin download page on LPCware.com](#)

Once you have downloaded this files, look at the installation.txt file in the "lpcopen/software/emWin" directory for information on how to install emWin into LPCOpen.



## Chapter 7

# Data Structure Documentation

### 7.1 APSR\_Type Union Reference

#### 7.1.1 Detailed Description

Union type to access the Application Program Status Register (APSR).

Definition at line 206 of file core\_cm0plus.h.

```
#include "core_cm0plus.h"
```

#### Data Fields

- struct {  
    uint32\_t [\\_reserved0](#):27  
    uint32\_t [Q](#):1  
    uint32\_t [V](#):1  
    uint32\_t [C](#):1  
    uint32\_t [Z](#):1  
    uint32\_t [N](#):1  
} [b](#)
- uint32\_t [w](#)
- struct {  
    uint32\_t [\\_reserved0](#):16  
    uint32\_t [GE](#):4  
    uint32\_t [\\_reserved1](#):7  
    uint32\_t [Q](#):1  
    uint32\_t [V](#):1  
    uint32\_t [C](#):1  
    uint32\_t [Z](#):1  
    uint32\_t [N](#):1  
} [b](#)

#### 7.1.2 Field Documentation

##### 7.1.2.1 uint32\_t \_reserved0

bit: 0..26 Reserved

bit: 0..15 Reserved

Definition at line 211 of file core\_cm0plus.h.

#### 7.1.2.2 uint32\_t\_reserved1

bit: 20..26 Reserved

Definition at line 260 of file core\_cm4.h.

#### 7.1.2.3 struct { ... } b

Structure used for bit access

#### 7.1.2.4 struct { ... } b

Structure used for bit access

#### 7.1.2.5 uint32\_t C

bit: 29 Carry condition code flag

Definition at line 219 of file core\_cm0plus.h.

#### 7.1.2.6 uint32\_t GE

bit: 16..19 Greater than or Equal flags

Definition at line 259 of file core\_cm4.h.

#### 7.1.2.7 uint32\_t N

bit: 31 Negative condition code flag

Definition at line 221 of file core\_cm0plus.h.

#### 7.1.2.8 uint32\_t Q

bit: 27 Saturation condition flag

Definition at line 217 of file core\_cm0plus.h.

#### 7.1.2.9 uint32\_t V

bit: 28 Overflow condition code flag

Definition at line 218 of file core\_cm0plus.h.

#### 7.1.2.10 uint32\_t w

Type used for word access

Definition at line 223 of file core\_cm0plus.h.

### 7.1.2.11 uint32\_t Z

bit: 30 Zero condition code flag

Definition at line 220 of file `core_cm0plus.h`.

The documentation for this union was generated from the following files:

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm0plus.h](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm4.h](#)

## 7.2 BM\_T Struct Reference

### 7.2.1 Detailed Description

Structure to define 8 bit USB request.

Definition at line 131 of file `usbd.h`.

```
#include "usbd.h"
```

#### Data Fields

- uint8\_t [Recipient](#): 5
- uint8\_t [Type](#): 2
- uint8\_t [Dir](#): 1

### 7.2.2 Field Documentation

#### 7.2.2.1 uint8\_t Dir

Direction type.

Definition at line 135 of file `usbd.h`.

#### 7.2.2.2 uint8\_t Recipient

Recipient type.

Definition at line 133 of file `usbd.h`.

#### 7.2.2.3 uint8\_t Type

Request type.

Definition at line 134 of file `usbd.h`.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.3 CONTROL\_Type Union Reference

### 7.3.1 Detailed Description

Union type to access the Control Registers (CONTROL).

Definition at line 268 of file core\_cm0plus.h.

```
#include "core_cm0plus.h"
```

#### Data Fields

- struct {
  - uint32\_t nPRIV:1
  - uint32\_t SPSEL:1
  - uint32\_t FPCA:1
  - uint32\_t \_reserved0:29
 } b
- uint32\_t w
- struct {
  - uint32\_t nPRIV:1
  - uint32\_t SPSEL:1
  - uint32\_t FPCA:1
  - uint32\_t \_reserved0:29
 } b

### 7.3.2 Field Documentation

#### 7.3.2.1 uint32\_t \_reserved0

bit: 3..31 Reserved

Definition at line 275 of file core\_cm0plus.h.

#### 7.3.2.2 struct { ... } b

Structure used for bit access

#### 7.3.2.3 struct { ... } b

Structure used for bit access

#### 7.3.2.4 uint32\_t FPCA

bit: 2 FP extension active flag

Definition at line 274 of file core\_cm0plus.h.

#### 7.3.2.5 uint32\_t nPRIV

bit: 0 Execution privilege in Thread mode

Definition at line 272 of file core\_cm0plus.h.

### 7.3.2.6 uint32\_t SPSEL

bit: 1 Stack to be used

Definition at line 273 of file core\_cm0plus.h.

### 7.3.2.7 uint32\_t w

Type used for word access

Definition at line 277 of file core\_cm0plus.h.

The documentation for this union was generated from the following files:

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm0plus.h
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm4.h

## 7.4 CoreDebug\_Type Struct Reference

### 7.4.1 Detailed Description

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

Definition at line 1271 of file core\_cm4.h.

```
#include "core_cm4.h"
```

#### Data Fields

- [\\_\\_IO uint32\\_t DHCSR](#)
- [\\_\\_O uint32\\_t DCRSR](#)
- [\\_\\_IO uint32\\_t DCRDR](#)
- [\\_\\_IO uint32\\_t DEMCR](#)

### 7.4.2 Field Documentation

#### 7.4.2.1 \_\_IO uint32\_t DCRDR

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

Definition at line 1275 of file core\_cm4.h.

#### 7.4.2.2 \_\_O uint32\_t DCRSR

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

Definition at line 1274 of file core\_cm4.h.

#### 7.4.2.3 \_\_IO uint32\_t DEMCR

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

Definition at line 1276 of file core\_cm4.h.

#### 7.4.2.4 `__IO uint32_t DHCSR`

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

Definition at line 1273 of file `core_cm4.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/core_cm4.h`

## 7.5 DMA\_CHDESC\_T Struct Reference

### 7.5.1 Detailed Description

Definition at line 523 of file `dma_5411x.h`.

```
#include "dma_5411x.h"
```

#### Data Fields

- `uint32_t xfercfg`
- `uint32_t source`
- `uint32_t dest`
- `uint32_t next`

### 7.5.2 Field Documentation

#### 7.5.2.1 `uint32_t dest`

DMA transfer destination end address

Definition at line 526 of file `dma_5411x.h`.

#### 7.5.2.2 `uint32_t next`

Link to next DMA descriptor, must be 16 byte aligned

Definition at line 527 of file `dma_5411x.h`.

#### 7.5.2.3 `uint32_t source`

DMA transfer source end address

Definition at line 525 of file `dma_5411x.h`.

#### 7.5.2.4 `uint32_t xfercfg`

Transfer configuration (only used in linked lists and ping-pong configs)

Definition at line 524 of file `dma_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/dma_5411x.h`



## 7.6 DMA\_DUAL\_DESCRIPTOR\_T Struct Reference

### 7.6.1 Detailed Description

Definition at line 55 of file dma\_service\_5411x.h.

```
#include "dma_service_5411x.h"
```

#### Data Fields

- [DMA\\_CHDESC\\_T descr](#) [2]

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[dma\\_service\\_5411x.h](#)

## 7.7 DMA\_PERIPHERAL\_CONTEXT\_T Struct Reference

### 7.7.1 Detailed Description

Definition at line 46 of file dma\_service\_5411x.h.

```
#include "dma_service_5411x.h"
```

#### Data Fields

- uint32\_t [channel](#)
- volatile uint32\_t \* [register\\_location](#)
- uint32\_t [width](#)
- uint32\_t [src\\_increment](#)
- uint32\_t [dst\\_increment](#)
- bool [write](#)

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[dma\\_service\\_5411x.h](#)

## 7.8 DMIC\_CHANNEL\_CONFIG\_T Struct Reference

### 7.8.1 Detailed Description

Definition at line 192 of file dmic\_5411x.h.

```
#include "dmic_5411x.h"
```

#### Data Fields

- [STEREO\\_SIDE\\_T side](#)
- [PDM\\_DIV\\_T divhclk](#)
- uint32\_t [osr](#)
- int32\_t [gainshft](#)
- [COMPENSATION\\_T preac2coef](#)

- [COMPENSATION\\_T preac4coef](#)

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[dmic\\_5411x.h](#)

## 7.9 DMIC\_STATISTICS\_T Struct Reference

### 7.9.1 Detailed Description

DMIC statistics structure.

#### Note

Maintains current DMIC statistics.

Definition at line 87 of file `dmic_5411x.h`.

```
#include "dmic_5411x.h"
```

#### Data Fields

- `uint32_t` [fifo\\_ints](#)
- `uint32_t` [fifo\\_overrun](#)
- `uint32_t` [fifo\\_underrun](#)

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[dmic\\_5411x.h](#)

## 7.10 DWT\_Type Struct Reference

### 7.10.1 Detailed Description

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

Definition at line 774 of file `core_cm4.h`.

```
#include "core_cm4.h"
```

#### Data Fields

- `__IO uint32_t` [CTRL](#)
- `__IO uint32_t` [CYCCNT](#)
- `__IO uint32_t` [CPICNT](#)
- `__IO uint32_t` [EXCCNT](#)
- `__IO uint32_t` [SLEEPCNT](#)
- `__IO uint32_t` [LSUCNT](#)
- `__IO uint32_t` [FOLDCNT](#)
- `__I uint32_t` [PCSR](#)
- `__IO uint32_t` [COMP0](#)
- `__IO uint32_t` [MASK0](#)
- `__IO uint32_t` [FUNCTION0](#)

- uint32\_t [RESERVED0](#) [1]
- [\\_\\_IO](#) uint32\_t [COMP1](#)
- [\\_\\_IO](#) uint32\_t [MASK1](#)
- [\\_\\_IO](#) uint32\_t [FUNCTION1](#)
- uint32\_t [RESERVED1](#) [1]
- [\\_\\_IO](#) uint32\_t [COMP2](#)
- [\\_\\_IO](#) uint32\_t [MASK2](#)
- [\\_\\_IO](#) uint32\_t [FUNCTION2](#)
- uint32\_t [RESERVED2](#) [1]
- [\\_\\_IO](#) uint32\_t [COMP3](#)
- [\\_\\_IO](#) uint32\_t [MASK3](#)
- [\\_\\_IO](#) uint32\_t [FUNCTION3](#)

## 7.10.2 Field Documentation

### 7.10.2.1 [\\_\\_IO](#) uint32\_t [COMP0](#)

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

Definition at line 784 of file core\_cm4.h.

### 7.10.2.2 [\\_\\_IO](#) uint32\_t [COMP1](#)

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

Definition at line 788 of file core\_cm4.h.

### 7.10.2.3 [\\_\\_IO](#) uint32\_t [COMP2](#)

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

Definition at line 792 of file core\_cm4.h.

### 7.10.2.4 [\\_\\_IO](#) uint32\_t [COMP3](#)

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

Definition at line 796 of file core\_cm4.h.

### 7.10.2.5 [\\_\\_IO](#) uint32\_t [CPICNT](#)

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

Definition at line 778 of file core\_cm4.h.

### 7.10.2.6 [\\_\\_IO](#) uint32\_t [CTRL](#)

Offset: 0x000 (R/W) Control Register

Definition at line 776 of file core\_cm4.h.

### 7.10.2.7 [\\_\\_IO](#) uint32\_t [CYCCNT](#)

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

Definition at line 777 of file core\_cm4.h.

#### 7.10.2.8 `__IO uint32_t EXCCNT`

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

Definition at line 779 of file core\_cm4.h.

#### 7.10.2.9 `__IO uint32_t FOLDCNT`

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

Definition at line 782 of file core\_cm4.h.

#### 7.10.2.10 `__IO uint32_t FUNCTION0`

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

Definition at line 786 of file core\_cm4.h.

#### 7.10.2.11 `__IO uint32_t FUNCTION1`

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

Definition at line 790 of file core\_cm4.h.

#### 7.10.2.12 `__IO uint32_t FUNCTION2`

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

Definition at line 794 of file core\_cm4.h.

#### 7.10.2.13 `__IO uint32_t FUNCTION3`

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

Definition at line 798 of file core\_cm4.h.

#### 7.10.2.14 `__IO uint32_t LSUCNT`

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

Definition at line 781 of file core\_cm4.h.

#### 7.10.2.15 `__IO uint32_t MASK0`

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

Definition at line 785 of file core\_cm4.h.

#### 7.10.2.16 `__IO uint32_t MASK1`

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

Definition at line 789 of file core\_cm4.h.

**7.10.2.17** `__IO uint32_t MASK2`

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

Definition at line 793 of file `core_cm4.h`.

**7.10.2.18** `__IO uint32_t MASK3`

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

Definition at line 797 of file `core_cm4.h`.

**7.10.2.19** `__I uint32_t PCSR`

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

Definition at line 783 of file `core_cm4.h`.

**7.10.2.20** `uint32_t RESERVED0[1]`

Definition at line 787 of file `core_cm4.h`.

**7.10.2.21** `uint32_t RESERVED1[1]`

Definition at line 791 of file `core_cm4.h`.

**7.10.2.22** `uint32_t RESERVED2[1]`

Definition at line 795 of file `core_cm4.h`.

**7.10.2.23** `__IO uint32_t SLEEP_CNT`

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

Definition at line 780 of file `core_cm4.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/core_cm4.h`

## 7.11 I2CM\_XFER\_T Struct Reference

### 7.11.1 Detailed Description

Master transfer data structure definitions.

Definition at line 74 of file `i2cm_5411x.h`.

```
#include "i2cm_5411x.h"
```

#### Data Fields

- `const uint8_t * txBuff`
- `uint8_t * rxBuff`

- [uint16\\_t txSz](#)
- [uint16\\_t rxSz](#)
- [uint16\\_t status](#)
- [uint8\\_t slaveAddr](#)

## 7.11.2 Field Documentation

### 7.11.2.1 `uint8_t* rxBuff`

Pointer memory where bytes received from I2C be stored

Definition at line 76 of file `i2cm_5411x.h`.

### 7.11.2.2 `uint16_t rxSz`

Number of bytes to received, if 0 only transmission we be carried on

Definition at line 79 of file `i2cm_5411x.h`.

### 7.11.2.3 `uint8_t slaveAddr`

7-bit I2C Slave address

Definition at line 82 of file `i2cm_5411x.h`.

### 7.11.2.4 `uint16_t status`

Status of the current I2C transfer

Definition at line 81 of file `i2cm_5411x.h`.

### 7.11.2.5 `const uint8_t* txBuff`

Pointer to array of bytes to be transmitted

Definition at line 75 of file `i2cm_5411x.h`.

### 7.11.2.6 `uint16_t txSz`

Number of bytes in transmit array, if 0 only receive transfer will be carried on

Definition at line 77 of file `i2cm_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/i2cm\_5411x.h`

## 7.12 I2CS\_XFER\_T Struct Reference

### 7.12.1 Detailed Description

Slave transfer are performed using 4 callbacks. These 3 callbacks handle most I2C slave transfer cases. When the slave is setup and a slave interrupt is receive and processed with the [Chip\\_I2CS\\_XferHandler\(\)](#) function in the I2C interrupt handler, one of these 4 callbacks is called. The callbacks can be used for unsized transfers from the master.

When an address is received, the `SlaveXferAddr()` callback is called with the received address. Only addresses enabled in the slave controller will be handled. The slave controller can support up to 4 slave addresses.

If the master is going to perform a read operation, the `SlaveXferSend()` callback is called. Place the data byte to send in `*data` and return a value of 0 or `I2C_SLVCTL_SLVCONTINUE` to the caller, or return a value of `I2C_SLVCTL_SLVNACK` to NACK the master. If you are using DMA and have setup a DMA descriptor in the callback, return `I2C_SLVCTL_SLVDMA`.

If the master performs a write operation, the `SlaveXferRecv()` callback is called with the received data. Return a value of 0 or `I2C_SLVCTL_SLVCONTINUE` to the caller to continue data transfer. Return `I2C_SLVCTL_SLVNACK` to NACK the master. If you are using DMA and have setup a DMA descriptor in the callback, return `I2C_SLVCTL_SLVDMA`.

Once the transfer completes, the `SlaveXferDone()` callback will be called.

Definition at line 103 of file `i2cs_5411x.h`.

```
#include "i2cs_5411x.h"
```

## Data Fields

- [I2CSlaveXferStart slaveStart](#)
- [I2CSlaveXferSend slaveSend](#)
- [I2CSlaveXferRecv slaveRecv](#)
- [I2CSlaveXferDone slaveDone](#)

## 7.12.2 Field Documentation

### 7.12.2.1 I2CSlaveXferDone slaveDone

Called when a slave transfer is complete

Definition at line 107 of file `i2cs_5411x.h`.

### 7.12.2.2 I2CSlaveXferRecv slaveRecv

Called when a byte is received from master

Definition at line 106 of file `i2cs_5411x.h`.

### 7.12.2.3 I2CSlaveXferSend slaveSend

Called when a byte is needed to send to master

Definition at line 105 of file `i2cs_5411x.h`.

### 7.12.2.4 I2CSlaveXferStart slaveStart

Called when an matching I2C slave address is received

Definition at line 104 of file `i2cs_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/i2cs_5411x.h`

## 7.13 I2S\_AUDIO\_FORMAT\_T Struct Reference

### 7.13.1 Detailed Description

Definition at line 182 of file i2s\_5411x.h.

```
#include "i2s_5411x.h"
```

#### Data Fields

- [I2S\\_DIR\\_T](#) Direction
- [I2S\\_MSTSLVCFG\\_T](#) MSCfg
- [I2S\\_MODE\\_T](#) Mode
- bool [RightLow](#)
- bool [LeftJust](#)
- bool [PDMDData](#)
- bool [SCKPol](#)
- bool [WSPol](#)
- uint32\_t [Divider](#)
- uint8\_t [ChannelNumber](#)
- uint16\_t [WordWidth](#)
- uint16\_t [FrameWidth](#)
- uint16\_t [DataPos](#)
- uint16\_t [FIFOdepth](#)

### 7.13.2 Field Documentation

#### 7.13.2.1 uint8\_t ChannelNumber

Channel Number - 1 is mono, 2 is stereo

Definition at line 192 of file i2s\_5411x.h.

#### 7.13.2.2 uint16\_t DataPos

Data position in the frame

Definition at line 195 of file i2s\_5411x.h.

#### 7.13.2.3 I2S\_DIR\_T Direction

Data direction: tx or rx

Definition at line 183 of file i2s\_5411x.h.

#### 7.13.2.4 uint32\_t Divider

Flexcomm function clock divider

Definition at line 191 of file i2s\_5411x.h.

#### 7.13.2.5 uint16\_t FIFOdepth

FIFO depth (fifo config)

Definition at line 196 of file i2s\_5411x.h.



#### 7.13.2.6 uint16\_t FrameWidth

Frame Width

Definition at line 194 of file i2s\_5411x.h.

#### 7.13.2.7 bool LeftJust

left justify data in FIFO

Definition at line 187 of file i2s\_5411x.h.

#### 7.13.2.8 I2S\_MODE\_T Mode

I2S mode

Definition at line 185 of file i2s\_5411x.h.

#### 7.13.2.9 I2S\_MSTSLVCFG\_T MSCfg

Master / Slave configuration

Definition at line 184 of file i2s\_5411x.h.

#### 7.13.2.10 bool PDMDData

data source is the D-Mic subsystem

Definition at line 188 of file i2s\_5411x.h.

#### 7.13.2.11 bool RightLow

right channel data in low portion of FIFO

Definition at line 186 of file i2s\_5411x.h.

#### 7.13.2.12 bool SCKPol

SCK polarity

Definition at line 189 of file i2s\_5411x.h.

#### 7.13.2.13 uint16\_t WordWidth

Word Width

Definition at line 193 of file i2s\_5411x.h.

#### 7.13.2.14 bool WSPol

WS polarity

Definition at line 190 of file i2s\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[i2s\\_5411x.h](#)

## 7.14 I2S\_STATISTICS\_T Struct Reference

### 7.14.1 Detailed Description

I2S statistics structure.

#### Note

Maintains current I2S statistics.

Definition at line 204 of file i2s\_5411x.h.

```
#include "i2s_5411x.h"
```

### Data Fields

- uint32\_t [interrupts](#)
- uint32\_t [lvl\\_tx](#)
- uint32\_t [lvl\\_rx](#)
- uint32\_t [fifo\\_err\\_tx](#)
- uint32\_t [fifo\\_err\\_rx](#)
- uint32\_t [i2s\\_busy](#)
- uint32\_t [i2s\\_slvfrmerr](#)
- uint32\_t [i2s\\_data\\_paused](#)

### 7.14.2 Field Documentation

#### 7.14.2.1 uint32\_t fifo\_err\_rx

count: FIFO receive errors

Definition at line 209 of file i2s\_5411x.h.

#### 7.14.2.2 uint32\_t fifo\_err\_tx

count: FIFO transmit errors

Definition at line 208 of file i2s\_5411x.h.

#### 7.14.2.3 uint32\_t i2s\_busy

count: I2S channel pair is busy

Definition at line 210 of file i2s\_5411x.h.

#### 7.14.2.4 uint32\_t i2s\_data\_paused

count: I2S data paused

Definition at line 212 of file i2s\_5411x.h.

#### 7.14.2.5 uint32\_t i2s\_slvfrmerr

count: I2S slave frame error

Definition at line 211 of file i2s\_5411x.h.

#### 7.14.2.6 uint32\_t interrupts

count: interrupts

Definition at line 205 of file i2s\_5411x.h.

#### 7.14.2.7 uint32\_t lvl\_rx

count: receive interrupts

Definition at line 207 of file i2s\_5411x.h.

#### 7.14.2.8 uint32\_t lvl\_tx

count: transmit interrupts

Definition at line 206 of file i2s\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[i2s\\_5411x.h](#)

## 7.15 IPSR\_Type Union Reference

### 7.15.1 Detailed Description

Union type to access the Interrupt Program Status Register (IPSR).

Definition at line 229 of file core\_cm0plus.h.

```
#include "core_cm0plus.h"
```

#### Data Fields

- struct {
  - uint32\_t [ISR](#):9
  - uint32\_t [\\_reserved0](#):23
- } [b](#)
- uint32\_t [w](#)
- struct {
  - uint32\_t [ISR](#):9
  - uint32\_t [\\_reserved0](#):23
- } [b](#)

### 7.15.2 Field Documentation

#### 7.15.2.1 uint32\_t \_reserved0

bit: 9..31 Reserved

Definition at line 234 of file core\_cm0plus.h.

### 7.15.2.2 struct { ... } b

Structure used for bit access

### 7.15.2.3 struct { ... } b

Structure used for bit access

### 7.15.2.4 uint32\_t ISR

bit: 0.. 8 Exception number

Definition at line 233 of file core\_cm0plus.h.

### 7.15.2.5 uint32\_t w

Type used for word access

Definition at line 236 of file core\_cm0plus.h.

The documentation for this union was generated from the following files:

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm0plus.h
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm4.h

## 7.16 ITM\_Type Struct Reference

### 7.16.1 Detailed Description

Structure type to access the Instrumentation Trace Macrocell Register (ITM).

Definition at line 673 of file core\_cm4.h.

```
#include "core_cm4.h"
```

### Data Fields

- union {
  - [\\_\\_O uint8\\_t u8](#)
  - [\\_\\_O uint16\\_t u16](#)
  - [\\_\\_O uint32\\_t u32](#)
- [PORT](#) [32]
- uint32\_t [RESERVED0](#) [864]
- [\\_\\_IO uint32\\_t TER](#)
- uint32\_t [RESERVED1](#) [15]
- [\\_\\_IO uint32\\_t TPR](#)
- uint32\_t [RESERVED2](#) [15]
- [\\_\\_IO uint32\\_t TCR](#)
- uint32\_t [RESERVED3](#) [29]
- [\\_\\_O uint32\\_t IWR](#)
- [\\_\\_I uint32\\_t IRR](#)
- [\\_\\_IO uint32\\_t IMCR](#)
- uint32\_t [RESERVED4](#) [43]
- [\\_\\_O uint32\\_t LAR](#)

- [\\_\\_I uint32\\_t LSR](#)
- [uint32\\_t RESERVED5](#) [6]
- [\\_\\_I uint32\\_t PID4](#)
- [\\_\\_I uint32\\_t PID5](#)
- [\\_\\_I uint32\\_t PID6](#)
- [\\_\\_I uint32\\_t PID7](#)
- [\\_\\_I uint32\\_t PID0](#)
- [\\_\\_I uint32\\_t PID1](#)
- [\\_\\_I uint32\\_t PID2](#)
- [\\_\\_I uint32\\_t PID3](#)
- [\\_\\_I uint32\\_t CID0](#)
- [\\_\\_I uint32\\_t CID1](#)
- [\\_\\_I uint32\\_t CID2](#)
- [\\_\\_I uint32\\_t CID3](#)

## 7.16.2 Field Documentation

### 7.16.2.1 [\\_\\_I uint32\\_t CID0](#)

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

Definition at line 703 of file core\_cm4.h.

### 7.16.2.2 [\\_\\_I uint32\\_t CID1](#)

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

Definition at line 704 of file core\_cm4.h.

### 7.16.2.3 [\\_\\_I uint32\\_t CID2](#)

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

Definition at line 705 of file core\_cm4.h.

### 7.16.2.4 [\\_\\_I uint32\\_t CID3](#)

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

Definition at line 706 of file core\_cm4.h.

### 7.16.2.5 [\\_\\_IO uint32\\_t IMCR](#)

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

Definition at line 690 of file core\_cm4.h.

### 7.16.2.6 [\\_\\_I uint32\\_t IRR](#)

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

Definition at line 689 of file core\_cm4.h.

#### 7.16.2.7 `__O uint32_t IWR`

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

Definition at line 688 of file core\_cm4.h.

#### 7.16.2.8 `__O uint32_t LAR`

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

Definition at line 692 of file core\_cm4.h.

#### 7.16.2.9 `__I uint32_t LSR`

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

Definition at line 693 of file core\_cm4.h.

#### 7.16.2.10 `__I uint32_t PID0`

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

Definition at line 699 of file core\_cm4.h.

#### 7.16.2.11 `__I uint32_t PID1`

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

Definition at line 700 of file core\_cm4.h.

#### 7.16.2.12 `__I uint32_t PID2`

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

Definition at line 701 of file core\_cm4.h.

#### 7.16.2.13 `__I uint32_t PID3`

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

Definition at line 702 of file core\_cm4.h.

#### 7.16.2.14 `__I uint32_t PID4`

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

Definition at line 695 of file core\_cm4.h.

#### 7.16.2.15 `__I uint32_t PID5`

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

Definition at line 696 of file core\_cm4.h.

**7.16.2.16** `__I uint32_t PID6`

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

Definition at line 697 of file core\_cm4.h.

**7.16.2.17** `__I uint32_t PID7`

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

Definition at line 698 of file core\_cm4.h.

**7.16.2.18** `__O { ... } PORT[32]`

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

**7.16.2.19** `uint32_t RESERVED0[864]`

Definition at line 681 of file core\_cm4.h.

**7.16.2.20** `uint32_t RESERVED1[15]`

Definition at line 683 of file core\_cm4.h.

**7.16.2.21** `uint32_t RESERVED2[15]`

Definition at line 685 of file core\_cm4.h.

**7.16.2.22** `uint32_t RESERVED3[29]`

Definition at line 687 of file core\_cm4.h.

**7.16.2.23** `uint32_t RESERVED4[43]`

Definition at line 691 of file core\_cm4.h.

**7.16.2.24** `uint32_t RESERVED5[6]`

Definition at line 694 of file core\_cm4.h.

**7.16.2.25** `__IO uint32_t TCR`

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

Definition at line 686 of file core\_cm4.h.

**7.16.2.26** `__IO uint32_t TER`

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

Definition at line 682 of file core\_cm4.h.

#### 7.16.2.27 `__IO uint32_t TPR`

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

Definition at line 684 of file `core_cm4.h`.

#### 7.16.2.28 `__O uint16_t u16`

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

Definition at line 678 of file `core_cm4.h`.

#### 7.16.2.29 `__O uint32_t u32`

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

Definition at line 679 of file `core_cm4.h`.

#### 7.16.2.30 `__O uint8_t u8`

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

Definition at line 677 of file `core_cm4.h`.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm4.h](#)

## 7.17 LPC\_ADC\_T Struct Reference

### 7.17.1 Detailed Description

ADC register block structure.

Definition at line 54 of file `adc_5411x.h`.

```
#include "adc_5411x.h"
```

### Data Fields

- `__IO uint32_t CTRL`
- `__IO uint32_t INSEL`
- union {
  - `__IO uint32_t SEQ_CTRL [2]`
  - struct {
    - `__IO uint32_t SEQA_CTRL`
    - `__IO uint32_t SEQB_CTRL`
- };
- union {
  - `__IO uint32_t SEQ_GDAT [2]`
  - struct {
    - `__IO uint32_t SEQA_GDAT`
    - `__IO uint32_t SEQB_GDAT`
- }



```
};
```

- [\\_\\_IO uint32\\_t RESERVED0](#) [2]
- [\\_\\_IO uint32\\_t DAT](#) [12]
- [\\_\\_IO uint32\\_t THR\\_LOW](#) [2]
- [\\_\\_IO uint32\\_t THR\\_HIGH](#) [2]
- [\\_\\_IO uint32\\_t CHAN\\_THRSEL](#)
- [\\_\\_IO uint32\\_t INTEN](#)
- [\\_\\_IO uint32\\_t FLAGS](#)
- [\\_\\_IO uint32\\_t STARTUP](#)
- [\\_\\_IO uint32\\_t CALIBR](#)

## 7.17.2 Field Documentation

### 7.17.2.1 union { ... }

### 7.17.2.2 union { ... }

### 7.17.2.3 \_\_IO uint32\_t CALIBR

Definition at line 83 of file adc\_5411x.h.

### 7.17.2.4 \_\_IO uint32\_t CHAN\_THRSEL

Definition at line 79 of file adc\_5411x.h.

### 7.17.2.5 \_\_IO uint32\_t CTRL

< ADCn Structure

Definition at line 55 of file adc\_5411x.h.

### 7.17.2.6 \_\_IO uint32\_t DAT[12]

Definition at line 76 of file adc\_5411x.h.

### 7.17.2.7 \_\_IO uint32\_t FLAGS

Definition at line 81 of file adc\_5411x.h.

### 7.17.2.8 \_\_IO uint32\_t INSEL

Definition at line 56 of file adc\_5411x.h.

### 7.17.2.9 \_\_IO uint32\_t INTEN

Definition at line 80 of file adc\_5411x.h.

### 7.17.2.10 \_\_IO uint32\_t RESERVED0[2]

Definition at line 75 of file adc\_5411x.h.

#### 7.17.2.11 `__IO uint32_t SEQ_CTRL[2]`

Definition at line 58 of file `adc_5411x.h`.

#### 7.17.2.12 `__IO uint32_t SEQ_GDAT[2]`

Definition at line 67 of file `adc_5411x.h`.

#### 7.17.2.13 `__IO uint32_t SEQA_CTRL`

Definition at line 60 of file `adc_5411x.h`.

#### 7.17.2.14 `__IO uint32_t SEQA_GDAT`

Definition at line 69 of file `adc_5411x.h`.

#### 7.17.2.15 `__IO uint32_t SEQB_CTRL`

Definition at line 61 of file `adc_5411x.h`.

#### 7.17.2.16 `__IO uint32_t SEQB_GDAT`

Definition at line 70 of file `adc_5411x.h`.

#### 7.17.2.17 `__IO uint32_t STARTUP`

Definition at line 82 of file `adc_5411x.h`.

#### 7.17.2.18 `__IO uint32_t THR_HIGH[2]`

Definition at line 78 of file `adc_5411x.h`.

#### 7.17.2.19 `__IO uint32_t THR_LOW[2]`

Definition at line 77 of file `adc_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/adc\_5411x.h`

## 7.18 LPC\_ASYNC\_SYSCON\_T Struct Reference

### 7.18.1 Detailed Description

LPC5411X Asynchronous system configuration register block structure.

Definition at line 152 of file `syscon_5411x.h`.

```
#include "syscon_5411x.h"
```

## Data Fields

- [\\_\\_IO uint32\\_t](#) [AYSNCPRESETCTRL](#)
- [\\_\\_O uint32\\_t](#) [ASYNCPRESETCTRLSET](#)
- [\\_\\_O uint32\\_t](#) [ASYNCPRESETCTRLCLR](#)
- [\\_\\_I uint32\\_t](#) [RESERVED0](#)
- [\\_\\_IO uint32\\_t](#) [ASYNCAPBCLKCTRL](#)
- [\\_\\_IO uint32\\_t](#) [ASYNCAPBCLKCTRLSET](#)
- [\\_\\_IO uint32\\_t](#) [ASYNCAPBCLKCTRLCLR](#)
- [\\_\\_I uint32\\_t](#) [RESERVED1](#)
- [\\_\\_IO uint32\\_t](#) [ASYNCAPBCLKSELA](#)

## 7.18.2 Field Documentation

### 7.18.2.1 [\\_\\_IO uint32\\_t](#) [ASYNCAPBCLKCTRL](#)

clk enable register

Definition at line 157 of file syscon\_5411x.h.

### 7.18.2.2 [\\_\\_IO uint32\\_t](#) [ASYNCAPBCLKCTRLCLR](#)

clk enable Clr register

Definition at line 159 of file syscon\_5411x.h.

### 7.18.2.3 [\\_\\_IO uint32\\_t](#) [ASYNCAPBCLKCTRLSET](#)

clk enable Set register

Definition at line 158 of file syscon\_5411x.h.

### 7.18.2.4 [\\_\\_IO uint32\\_t](#) [ASYNCAPBCLKSELA](#)

clk source mux A register

Definition at line 161 of file syscon\_5411x.h.

### 7.18.2.5 [\\_\\_O uint32\\_t](#) [ASYNCPRESETCTRLCLR](#)

peripheral reset Clr register

Definition at line 155 of file syscon\_5411x.h.

### 7.18.2.6 [\\_\\_O uint32\\_t](#) [ASYNCPRESETCTRLSET](#)

peripheral reset Set register

Definition at line 154 of file syscon\_5411x.h.

### 7.18.2.7 [\\_\\_IO uint32\\_t](#) [AYSNCPRESETCTRL](#)

peripheral reset register

Definition at line 153 of file syscon\_5411x.h.

### 7.18.2.8 `__IO uint32_t RESERVED0`

Definition at line 156 of file `syscon_5411x.h`.

### 7.18.2.9 `__IO uint32_t RESERVED1`

Definition at line 160 of file `syscon_5411x.h`.

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

- [C:/Jenkins/LPC5411x/lpc5411x/chip\\_5411x/inc/syscon\\_5411x.h](#)

## 7.19 LPC\_CRC\_T Struct Reference

### 7.19.1 Detailed Description

CRC register block structure.

Definition at line 47 of file `crc_5411x.h`.

```
#include "crc_5411x.h"
```

### Data Fields

- `__IO uint32_t MODE`
  - `__IO uint32_t SEED`
  - union {
    - `__IO uint32_t SUM`
    - `__IO uint32_t WRDATA32`
    - `__IO uint16_t WRDATA16`
    - `__IO uint8_t WRDATA8`
- ```
};
```

### 7.19.2 Field Documentation

#### 7.19.2.1 union { ... }

#### 7.19.2.2 `__IO uint32_t MODE`

< CRC Structure CRC Mode Register

Definition at line 48 of file `crc_5411x.h`.

#### 7.19.2.3 `__IO uint32_t SEED`

CRC SEED Register

Definition at line 49 of file `crc_5411x.h`.

#### 7.19.2.4 `__IO uint32_t SUM`

CRC Checksum Register.

Definition at line 51 of file `crc_5411x.h`.

7.19.2.5 `__O uint16_t WRDATA16`

CRC Data Register: write size 16-bit

Definition at line 53 of file `crc_5411x.h`.

7.19.2.6 `__O uint32_t WRDATA32`

CRC Data Register: write size 32-bit

Definition at line 52 of file `crc_5411x.h`.

7.19.2.7 `__O uint8_t WRDATA8`

CRC Data Register: write size 8-bit

Definition at line 54 of file `crc_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/crc_5411x.h`

## 7.20 LPC\_DMA\_CHANNEL\_T Struct Reference

### 7.20.1 Detailed Description

DMA Controller shared registers structure.

Definition at line 75 of file `dma_5411x.h`.

```
#include "dma_5411x.h"
```

#### Data Fields

- `__IO uint32_t CFG`
- `__IO uint32_t CTLSTAT`
- `__IO uint32_t XFRCFG`
- `__I uint32_t RESERVED`

### 7.20.2 Field Documentation

7.20.2.1 `__IO uint32_t CFG`

< DMA channel register structure DMA Configuration register

Definition at line 76 of file `dma_5411x.h`.

7.20.2.2 `__IO uint32_t CTLSTAT`

DMA Control and status register

Definition at line 77 of file `dma_5411x.h`.

7.20.2.3 `__I uint32_t RESERVED`

Definition at line 79 of file `dma_5411x.h`.

#### 7.20.2.4 `__IO uint32_t XFERCFG`

DMA Transfer configuration register

Definition at line 78 of file `dma_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/dma_5411x.h`

## 7.21 `LPC_DMA_COMMON_T` Struct Reference

### 7.21.1 Detailed Description

DMA Controller shared registers structure.

Definition at line 46 of file `dma_5411x.h`.

```
#include "dma_5411x.h"
```

### Data Fields

- `__IO uint32_t ENABLESET`
- `__I uint32_t RESERVED0`
- `__O uint32_t ENABLECLR`
- `__I uint32_t RESERVED1`
- `__I uint32_t ACTIVE`
- `__I uint32_t RESERVED2`
- `__I uint32_t BUSY`
- `__I uint32_t RESERVED3`
- `__IO uint32_t ERRINT`
- `__I uint32_t RESERVED4`
- `__IO uint32_t INTENSET`
- `__I uint32_t RESERVED5`
- `__O uint32_t INTENCLR`
- `__I uint32_t RESERVED6`
- `__IO uint32_t INTA`
- `__I uint32_t RESERVED7`
- `__IO uint32_t INTB`
- `__I uint32_t RESERVED8`
- `__O uint32_t SETVALID`
- `__I uint32_t RESERVED9`
- `__O uint32_t SETTRIG`
- `__I uint32_t RESERVED10`
- `__O uint32_t ABORT`

### 7.21.2 Field Documentation

#### 7.21.2.1 `__O uint32_t ABORT`

DMA Channel Abort control for all DMA channels

Definition at line 69 of file `dma_5411x.h`.

#### 7.21.2.2 `__I uint32_t ACTIVE`

DMA Channel Active status for all DMA channels

Definition at line 51 of file dma\_5411x.h.

#### 7.21.2.3 `__I uint32_t BUSY`

DMA Channel Busy status for all DMA channels

Definition at line 53 of file dma\_5411x.h.

#### 7.21.2.4 `__O uint32_t ENABLECLR`

DMA Channel Enable Clear for all DMA channels

Definition at line 49 of file dma\_5411x.h.

#### 7.21.2.5 `__IO uint32_t ENABLESET`

< DMA shared registers structure DMA Channel Enable read and Set for all DMA channels

Definition at line 47 of file dma\_5411x.h.

#### 7.21.2.6 `__IO uint32_t ERRINT`

DMA Error Interrupt status for all DMA channels

Definition at line 55 of file dma\_5411x.h.

#### 7.21.2.7 `__IO uint32_t INTA`

DMA Interrupt A status for all DMA channels

Definition at line 61 of file dma\_5411x.h.

#### 7.21.2.8 `__IO uint32_t INTB`

DMA Interrupt B status for all DMA channels

Definition at line 63 of file dma\_5411x.h.

#### 7.21.2.9 `__O uint32_t INTENCLR`

DMA Interrupt Enable Clear for all DMA channels

Definition at line 59 of file dma\_5411x.h.

#### 7.21.2.10 `__IO uint32_t INTENSET`

DMA Interrupt Enable read and Set for all DMA channels

Definition at line 57 of file dma\_5411x.h.

#### 7.21.2.11 `__I uint32_t RESERVED0`

Definition at line 48 of file dma\_5411x.h.

#### 7.21.2.12 `__I uint32_t RESERVED1`

Definition at line 50 of file dma\_5411x.h.

#### 7.21.2.13 `__I uint32_t RESERVED10`

Definition at line 68 of file dma\_5411x.h.

#### 7.21.2.14 `__I uint32_t RESERVED2`

Definition at line 52 of file dma\_5411x.h.

#### 7.21.2.15 `__I uint32_t RESERVED3`

Definition at line 54 of file dma\_5411x.h.

#### 7.21.2.16 `__I uint32_t RESERVED4`

Definition at line 56 of file dma\_5411x.h.

#### 7.21.2.17 `__I uint32_t RESERVED5`

Definition at line 58 of file dma\_5411x.h.

#### 7.21.2.18 `__I uint32_t RESERVED6`

Definition at line 60 of file dma\_5411x.h.

#### 7.21.2.19 `__I uint32_t RESERVED7`

Definition at line 62 of file dma\_5411x.h.

#### 7.21.2.20 `__I uint32_t RESERVED8`

Definition at line 64 of file dma\_5411x.h.

#### 7.21.2.21 `__I uint32_t RESERVED9`

Definition at line 66 of file dma\_5411x.h.

#### 7.21.2.22 `__O uint32_t SETTRIG`

DMA Set Trigger control bits for all DMA channels

Definition at line 67 of file dma\_5411x.h.



7.21.2.23 `__O uint32_t SETVALID`

DMA Set ValidPending control bits for all DMA channels

Definition at line 65 of file `dma_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/dma_5411x.h`

## 7.22 LPC\_DMA\_T Struct Reference

### 7.22.1 Detailed Description

DMA Controller register block structure.

Definition at line 88 of file `dma_5411x.h`.

```
#include "dma_5411x.h"
```

#### Data Fields

- volatile uint32\_t `CTRL`
- volatile uint32\_t `INTSTAT`
- volatile uint32\_t `SRMBASE`
- volatile uint32\_t `RESERVED2` [5]
- `LPC_DMA_COMMON_T DMACOMMON` [1]
- volatile uint32\_t `RESERVED0` [225]
- `LPC_DMA_CHANNEL_T DMACH` [`MAX_DMA_CHANNEL`]

### 7.22.2 Field Documentation

#### 7.22.2.1 volatile uint32\_t CTRL

< DMA Structure DMA control register

Definition at line 89 of file `dma_5411x.h`.

#### 7.22.2.2 LPC\_DMA\_CHANNEL\_T DMACH[MAX\_DMA\_CHANNEL]

DMA channel registers

Definition at line 95 of file `dma_5411x.h`.

#### 7.22.2.3 LPC\_DMA\_COMMON\_T DMACOMMON[1]

DMA shared channel (common) registers

Definition at line 93 of file `dma_5411x.h`.

#### 7.22.2.4 volatile uint32\_t INTSTAT

DMA Interrupt status register

Definition at line 90 of file `dma_5411x.h`.

#### 7.22.2.5 volatile uint32\_t RESERVED0[225]

Definition at line 94 of file dma\_5411x.h.

#### 7.22.2.6 volatile uint32\_t RESERVED2[5]

Definition at line 92 of file dma\_5411x.h.

#### 7.22.2.7 volatile uint32\_t SRAMBASE

DMA SRAM address of the channel configuration table

Definition at line 91 of file dma\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/dma\_5411x.h

## 7.23 LPC\_DMIC\_Channel\_Type Struct Reference

### 7.23.1 Detailed Description

Definition at line 44 of file dmic\_5411x.h.

```
#include "dmic_5411x.h"
```

#### Data Fields

- \_\_IO uint32\_t OSR
- \_\_IO uint32\_t DIVHFCLK
- \_\_IO uint32\_t PREAC2FSCOEf
- \_\_IO uint32\_t PREAC4FSCOEf
- \_\_IO uint32\_t GAINSHFT
- \_\_IO uint32\_t TDM96EN
- \_\_IO uint32\_t TDM19EN
- \_\_IO uint32\_t reserved [25]
- \_\_IO uint32\_t FIFO\_CTRL
- \_\_IO uint32\_t FIFO\_STATUS
- \_\_IO uint32\_t FIFO\_DATA
- \_\_IO uint32\_t PHY\_CTRL
- \_\_IO uint32\_t DC\_CTRL
- \_\_IO uint32\_t reserved1 [27]

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/dmic\_5411x.h

## 7.24 LPC\_DMIC\_T Struct Reference

### 7.24.1 Detailed Description

Definition at line 62 of file dmic\_5411x.h.

```
#include "dmic_5411x.h"
```

## Data Fields

- [\\_\\_IO LPC\\_DMIC\\_Channel\\_Type CHANNEL](#) [15]
- [\\_\\_IO uint32\\_t CHANEN](#)
- [\\_\\_IO uint32\\_t reserved0](#) [2]
- [\\_\\_IO uint32\\_t IOCFG](#)
- [\\_\\_IO uint32\\_t USE2FS](#)
- [\\_\\_IO uint32\\_t reserved](#) [27]
- [\\_\\_IO uint32\\_t HWVADGAIN](#)
- [\\_\\_IO uint32\\_t HWVADHPFS](#)
- [\\_\\_IO uint32\\_t HWVADST10](#)
- [\\_\\_IO uint32\\_t HWVADRSTT](#)
- [\\_\\_IO uint32\\_t HWVADTHGN](#)
- [\\_\\_IO uint32\\_t HWVADTHGS](#)
- [\\_\\_IO uint32\\_t HWVADLOWZ](#)
- [\\_\\_IO uint32\\_t reserved1](#) [24]
- [\\_\\_O uint32\\_t ID](#)

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

- [C:/Jenkins/LPC5411x/lpc5411x/chip\\_5411x/inc/dmic\\_5411x.h](#)

## 7.25 LPC\_GPIO\_T Struct Reference

### 7.25.1 Detailed Description

GPIO port register block structure.

Definition at line 47 of file gpio\_5411x.h.

```
#include "gpio_5411x.h"
```

## Data Fields

- [\\_\\_IO uint8\\_t B](#) [128][32]
- [\\_\\_IO uint32\\_t W](#) [32][32]
- [\\_\\_IO uint32\\_t DIR](#) [32]
- [\\_\\_IO uint32\\_t MASK](#) [32]
- [\\_\\_IO uint32\\_t PIN](#) [32]
- [\\_\\_IO uint32\\_t MPIN](#) [32]
- [\\_\\_IO uint32\\_t SET](#) [32]
- [\\_\\_O uint32\\_t CLR](#) [32]
- [\\_\\_O uint32\\_t NOT](#) [32]

### 7.25.2 Field Documentation

#### 7.25.2.1 [\\_\\_IO uint8\\_t B](#)[128][32]

< GPIO\_PORT Structure Offset 0x0000: Byte pin registers ports 0 to n; pins PION\_0 to PION\_31

Definition at line 48 of file gpio\_5411x.h.

#### 7.25.2.2 `__O uint32_t CLR[32]`

Offset 0x2280: Clear port n

Definition at line 55 of file `gpio_5411x.h`.

#### 7.25.2.3 `__IO uint32_t DIR[32]`

Offset 0x2000: Direction registers port n

Definition at line 50 of file `gpio_5411x.h`.

#### 7.25.2.4 `__IO uint32_t MASK[32]`

Offset 0x2080: Mask register port n

Definition at line 51 of file `gpio_5411x.h`.

#### 7.25.2.5 `__IO uint32_t MPIN[32]`

Offset 0x2180: Masked port register port n

Definition at line 53 of file `gpio_5411x.h`.

#### 7.25.2.6 `__O uint32_t NOT[32]`

Offset 0x2300: Toggle port n

Definition at line 56 of file `gpio_5411x.h`.

#### 7.25.2.7 `__IO uint32_t PIN[32]`

Offset 0x2100: Portpin register port n

Definition at line 52 of file `gpio_5411x.h`.

#### 7.25.2.8 `__IO uint32_t SET[32]`

Offset 0x2200: Write: Set register for port n Read: output bits for port n

Definition at line 54 of file `gpio_5411x.h`.

#### 7.25.2.9 `__IO uint32_t W[32][32]`

Offset 0x1000: Word pin registers port 0 to n

Definition at line 49 of file `gpio_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/gpio_5411x.h`

## 7.26 `LPC_GPIOGROUPINT_T` Struct Reference

### 7.26.1 Detailed Description

GPIO grouped interrupt register block structure.

Definition at line 47 of file `gpiogroup_5411x.h`.

```
#include "gpiogroup_5411x.h"
```

#### Data Fields

- `__IO uint32_t CTRL`
- `__I uint32_t RESERVED0 [7]`
- `__IO uint32_t PORT_POL [8]`
- `__IO uint32_t PORT_ENA [8]`
- `uint32_t RESERVED1 [4072]`

### 7.26.2 Field Documentation

#### 7.26.2.1 `__IO uint32_t CTRL`

< GPIO\_GROUP\_INTn Structure GPIO grouped interrupt control register

Definition at line 48 of file `gpiogroup_5411x.h`.

#### 7.26.2.2 `__IO uint32_t PORT_ENA[8]`

GPIO grouped interrupt port m enable register

Definition at line 51 of file `gpiogroup_5411x.h`.

#### 7.26.2.3 `__IO uint32_t PORT_POL[8]`

GPIO grouped interrupt port polarity register

Definition at line 50 of file `gpiogroup_5411x.h`.

#### 7.26.2.4 `__I uint32_t RESERVED0[7]`

Definition at line 49 of file `gpiogroup_5411x.h`.

#### 7.26.2.5 `uint32_t RESERVED1[4072]`

Definition at line 52 of file `gpiogroup_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/gpiogroup_5411x.h`

## 7.27 LPC\_I2S\_T Struct Reference

### 7.27.1 Detailed Description

I2S register block structure.

Definition at line 42 of file `i2s_5411x.h`.

```
#include "i2s_5411x.h"
```

## Data Fields

- [\\_\\_IO uint32\\_t RESERVED0A](#) [768]
- [\\_\\_IO uint32\\_t CFG1](#)
- [\\_\\_IO uint32\\_t CFG2](#)
- [\\_\\_IO uint32\\_t STAT](#)
- [\\_\\_I uint32\\_t RESERVED00](#) [4]
- [\\_\\_IO uint32\\_t DIV](#)
- [\\_\\_I uint32\\_t RESERVED0](#) [120]
- [\\_\\_IO uint32\\_t FIFOCFG](#)
- [\\_\\_IO uint32\\_t FIFOSTAT](#)
- [\\_\\_IO uint32\\_t FIFOTRIG](#)
- [\\_\\_IO uint32\\_t FIFOINTENSET](#)
- [\\_\\_IO uint32\\_t FIFOINTENCLR](#)
- [\\_\\_IO uint32\\_t FIFOINTSTAT](#)
- [\\_\\_O uint32\\_t FIFOWR](#)
- [\\_\\_I uint32\\_t FIFORD](#)
- [\\_\\_I uint32\\_t FIFORDNOPOP](#)
- [\\_\\_I uint32\\_t FIFORD48HNOPOP](#)
- [\\_\\_I uint32\\_t RESERVED5](#) [108]
- [\\_\\_IO uint32\\_t PSELID](#)
- [\\_\\_I uint32\\_t PID](#)

## 7.27.2 Field Documentation

### 7.27.2.1 [\\_\\_IO uint32\\_t CFG1](#)

Offset: 0xC00 Configuration register #1

Definition at line 44 of file i2s\_5411x.h.

### 7.27.2.2 [\\_\\_IO uint32\\_t CFG2](#)

Offset: 0xC04 Configuration register #2

Definition at line 45 of file i2s\_5411x.h.

### 7.27.2.3 [\\_\\_IO uint32\\_t DIV](#)

Offset: 0xC1C Clock divider, used by all channel pairs.

Definition at line 48 of file i2s\_5411x.h.

### 7.27.2.4 [\\_\\_IO uint32\\_t FIFOCFG](#)

I2S FIFO Specific registers Offset: 0xE00 FIFO Configuration register

Definition at line 52 of file i2s\_5411x.h.

### 7.27.2.5 [\\_\\_IO uint32\\_t FIFOINTENCLR](#)

Offset: 0xE14 FIFO Interrupt enable CLEAR register

Definition at line 57 of file i2s\_5411x.h.

#### 7.27.2.6 `__IO uint32_t FIFOUTENSET`

Offset: 0xE10 FIFO Interrupt enable SET register

Definition at line 56 of file i2s\_5411x.h.

#### 7.27.2.7 `__IO uint32_t FIFOUTSTAT`

Offset: 0xE18 FIFO Interrupt Status register

Definition at line 58 of file i2s\_5411x.h.

#### 7.27.2.8 `__I uint32_t FIFORD`

Offset: 0xE30 FIFO Data read register

Definition at line 63 of file i2s\_5411x.h.

#### 7.27.2.9 `__I uint32_t FIFORD48HNOPOP`

Offset: 0xE44 FIFO data read for upper data bits with no FIFO pop

Definition at line 67 of file i2s\_5411x.h.

#### 7.27.2.10 `__I uint32_t FIFORDNOPOP`

Offset: 0xE40 FIFO data read with no FIFO pop

Definition at line 66 of file i2s\_5411x.h.

#### 7.27.2.11 `__IO uint32_t FIFOSTAT`

Offset: 0xE04 FIFO Status register

Definition at line 53 of file i2s\_5411x.h.

#### 7.27.2.12 `__IO uint32_t FIFOTRIG`

Offset: 0xE08 FIFO Trigger level register

Definition at line 54 of file i2s\_5411x.h.

#### 7.27.2.13 `__O uint32_t FIFOWR`

Offset: 0xE20 FIFO Data write register

Definition at line 60 of file i2s\_5411x.h.

#### 7.27.2.14 `__I uint32_t PID`

Offset: 0xFFC Module identification register

Definition at line 72 of file i2s\_5411x.h.

**7.27.2.15** `__IO uint32_t PSELID`

FLEXCOMM Interface registers Offset: 0xFF8 Peripheral select/identification register

Definition at line 71 of file `i2s_5411x.h`.

**7.27.2.16** `__I uint32_t RESERVED0[120]`

Offset: 0xC10 Reserved member

Definition at line 49 of file `i2s_5411x.h`.

**7.27.2.17** `__I uint32_t RESERVED00[4]`

Offset: 0xC0C Reserved registers

Definition at line 47 of file `i2s_5411x.h`.

**7.27.2.18** `__IO uint32_t RESERVED0A[768]`

< I2S Structure

Definition at line 43 of file `i2s_5411x.h`.

**7.27.2.19** `__I uint32_t RESERVED5[108]`

Offset: 0xE48 Reserved register

Definition at line 68 of file `i2s_5411x.h`.

**7.27.2.20** `__IO uint32_t STAT`

Offset: 0xC08 Status register

Definition at line 46 of file `i2s_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/i2s_5411x.h`

## 7.28 LPC\_INMUX\_T Struct Reference

### 7.28.1 Detailed Description

LPC5411X Input Mux Register Block Structure.

Definition at line 47 of file `inmux_5411x.h`.

```
#include "inmux_5411x.h"
```

#### Data Fields

- `__I uint32_t RESERVED1` [48]
- `__IO uint32_t PINTSEL` [8]
- `__IO uint32_t DMA_ITRIG_INMUX` [22]
- `__I uint32_t RESERVED2` [10]



- [\\_\\_IO uint32\\_t DMA\\_OTRIG\\_INMUX](#) [4]
- [\\_\\_I uint32\\_t RESERVED3](#) [4]
- [\\_\\_IO uint32\\_t FREQMEAS\\_REF](#)
- [\\_\\_IO uint32\\_t FREQMEAS\\_TARGET](#)

## 7.28.2 Field Documentation

### 7.28.2.1 [\\_\\_IO uint32\\_t DMA\\_ITRIG\\_INMUX](#)[22]

Input mux register for DMA trigger inputs

Definition at line 50 of file `inmux_5411x.h`.

### 7.28.2.2 [\\_\\_IO uint32\\_t DMA\\_OTRIG\\_INMUX](#)[4]

Input mux register for DMA trigger inputs

Definition at line 52 of file `inmux_5411x.h`.

### 7.28.2.3 [\\_\\_IO uint32\\_t FREQMEAS\\_REF](#)

Clock selection for frequency measurement ref clock

Definition at line 54 of file `inmux_5411x.h`.

### 7.28.2.4 [\\_\\_IO uint32\\_t FREQMEAS\\_TARGET](#)

Clock selection for frequency measurement target clock

Definition at line 55 of file `inmux_5411x.h`.

### 7.28.2.5 [\\_\\_IO uint32\\_t PINTSEL](#)[8]

Pin interrupt select registers

Definition at line 49 of file `inmux_5411x.h`.

### 7.28.2.6 [\\_\\_I uint32\\_t RESERVED1](#)[48]

< INMUX Structure

Definition at line 48 of file `inmux_5411x.h`.

### 7.28.2.7 [\\_\\_I uint32\\_t RESERVED2](#)[10]

Definition at line 51 of file `inmux_5411x.h`.

### 7.28.2.8 [\\_\\_I uint32\\_t RESERVED3](#)[4]

Definition at line 53 of file `inmux_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/inmux_5411x.h`

## 7.29 LPC\_IOCON\_T Struct Reference

### 7.29.1 Detailed Description

LPC5411X IO Configuration Unit register block structure.

Definition at line 47 of file iocon\_5411x.h.

```
#include "iocon_5411x.h"
```

#### Data Fields

- [\\_\\_IO uint32\\_t PIO](#) [2][32]

### 7.29.2 Field Documentation

#### 7.29.2.1 [\\_\\_IO uint32\\_t PIO](#)[2][32]

< LPC5411X IOCON Structure

Definition at line 48 of file iocon\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[iocon\\_5411x.h](#)

## 7.30 LPC\_MBOX\_T Struct Reference

### 7.30.1 Detailed Description

Mailbox register structure

Definition at line 60 of file mailbox\_5411x.h.

```
#include "mailbox_5411x.h"
```

#### Data Fields

- [LPC\\_MBOXIRQ\\_T BOX](#) [[MAILBOX\\_AVAIL](#)]
- [LPC\\_MBOXIRQ\\_T RESERVED1](#) [[15-MAILBOX\\_AVAIL](#)]
- [\\_\\_I uint32\\_t RESERVED2](#) [2]
- [\\_\\_IO uint32\\_t MUTEX](#)

### 7.30.2 Field Documentation

#### 7.30.2.1 [LPC\\_MBOXIRQ\\_T BOX](#)[[MAILBOX\\_AVAIL](#)]

< Mailbox register structure Mailbox, offset 0 = M0+, offset 1 = M4

Definition at line 61 of file mailbox\_5411x.h.

#### 7.30.2.2 [\\_\\_IO uint32\\_t MUTEX](#)

Mutex

Definition at line 64 of file mailbox\_5411x.h.

### 7.30.2.3 LPC\_MBOXIRQ\_T RESERVED1[15-MAILBOX\_AVAIL]

Definition at line 62 of file mailbox\_5411x.h.

### 7.30.2.4 \_\_I uint32\_t RESERVED2[2]

Definition at line 63 of file mailbox\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[mailbox\\_5411x.h](#)

## 7.31 LPC\_MBOXIRQ\_T Struct Reference

### 7.31.1 Detailed Description

Individual mailbox IRQ structure

Definition at line 52 of file mailbox\_5411x.h.

```
#include "mailbox_5411x.h"
```

#### Data Fields

- [\\_\\_IO uint32\\_t IRQ](#)
- [\\_\\_O uint32\\_t IRQSET](#)
- [\\_\\_O uint32\\_t IRQCLR](#)
- [\\_\\_I uint32\\_t RESERVED](#)

### 7.31.2 Field Documentation

#### 7.31.2.1 \_\_IO uint32\_t IRQ

Mailbox data

Definition at line 53 of file mailbox\_5411x.h.

#### 7.31.2.2 \_\_O uint32\_t IRQCLR

Mailbox dataclearset bits only

Definition at line 55 of file mailbox\_5411x.h.

#### 7.31.2.3 \_\_O uint32\_t IRQSET

Mailbox data set bits only

Definition at line 54 of file mailbox\_5411x.h.

#### 7.31.2.4 \_\_I uint32\_t RESERVED

Definition at line 56 of file mailbox\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[mailbox\\_5411x.h](#)

## 7.32 LPC\_MRT\_CH\_T Struct Reference

### 7.32.1 Detailed Description

MRT register block structure.

Definition at line 53 of file mrt\_5411x.h.

```
#include "mrt_5411x.h"
```

#### Data Fields

- [\\_\\_IO uint32\\_t](#) INTVAL
- [\\_\\_O uint32\\_t](#) TIMER
- [\\_\\_IO uint32\\_t](#) CTRL
- [\\_\\_IO uint32\\_t](#) STAT

### 7.32.2 Field Documentation

#### 7.32.2.1 [\\_\\_IO uint32\\_t](#) CTRL

Timer control register

Definition at line 56 of file mrt\_5411x.h.

#### 7.32.2.2 [\\_\\_IO uint32\\_t](#) INTVAL

Timer interval register

Definition at line 54 of file mrt\_5411x.h.

#### 7.32.2.3 [\\_\\_IO uint32\\_t](#) STAT

Timer status register

Definition at line 57 of file mrt\_5411x.h.

#### 7.32.2.4 [\\_\\_O uint32\\_t](#) TIMER

Timer register

Definition at line 55 of file mrt\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[mrt\\_5411x.h](#)

## 7.33 LPC\_MRT\_T Struct Reference

### 7.33.1 Detailed Description

MRT register block structure.

Definition at line 63 of file mrt\_5411x.h.

```
#include "mrt_5411x.h"
```

## Data Fields

- [LPC\\_MRT\\_CH\\_T CHANNEL](#) [[MRT\\_CHANNELS\\_NUM](#)]
- [uint32\\_t unused](#) [44]
- [\\_\\_IO uint32\\_t MODCFG](#)
- [\\_\\_O uint32\\_t IDLE\\_CH](#)
- [\\_\\_IO uint32\\_t IRQ\\_FLAG](#)

### 7.33.2 Field Documentation

#### 7.33.2.1 LPC\_MRT\_CH\_T CHANNEL[MRT\_CHANNELS\_NUM]

Definition at line 64 of file `mrt_5411x.h`.

#### 7.33.2.2 \_\_O uint32\_t IDLE\_CH

Definition at line 67 of file `mrt_5411x.h`.

#### 7.33.2.3 \_\_IO uint32\_t IRQ\_FLAG

Definition at line 68 of file `mrt_5411x.h`.

#### 7.33.2.4 \_\_IO uint32\_t MODCFG

Definition at line 66 of file `mrt_5411x.h`.

#### 7.33.2.5 uint32\_t unused[44]

Definition at line 65 of file `mrt_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/mrt_5411x.h`

## 7.34 LPC\_PIN\_INT\_T Struct Reference

### 7.34.1 Detailed Description

LPC5411X Pin Interrupt and Pattern Match register block structure.

Definition at line 47 of file `pinint_5411x.h`.

```
#include "pinint_5411x.h"
```

## Data Fields

- [\\_\\_IO uint32\\_t ISEL](#)
- [\\_\\_IO uint32\\_t IENR](#)
- [\\_\\_O uint32\\_t SIENR](#)
- [\\_\\_O uint32\\_t CIENR](#)
- [\\_\\_IO uint32\\_t IENF](#)
- [\\_\\_O uint32\\_t SIENF](#)

- [\\_\\_O uint32\\_t CIENF](#)
- [\\_\\_IO uint32\\_t RISE](#)
- [\\_\\_IO uint32\\_t FALL](#)
- [\\_\\_IO uint32\\_t IST](#)
- [\\_\\_IO uint32\\_t PMCTRL](#)
- [\\_\\_IO uint32\\_t PMSRC](#)
- [\\_\\_IO uint32\\_t PMCFG](#)

### 7.34.2 Field Documentation

#### 7.34.2.1 [\\_\\_O uint32\\_t CIENF](#)

Clear Pin Interrupt Enable Falling Edge / Active Level address

Definition at line 54 of file pinint\_5411x.h.

#### 7.34.2.2 [\\_\\_O uint32\\_t CIENR](#)

Clear Pin Interrupt Enable (Rising) register

Definition at line 51 of file pinint\_5411x.h.

#### 7.34.2.3 [\\_\\_IO uint32\\_t FALL](#)

Pin Interrupt Falling Edge register

Definition at line 56 of file pinint\_5411x.h.

#### 7.34.2.4 [\\_\\_IO uint32\\_t IENF](#)

Pin Interrupt Enable Falling Edge / Active Level register

Definition at line 52 of file pinint\_5411x.h.

#### 7.34.2.5 [\\_\\_IO uint32\\_t IENR](#)

Pin Interrupt Enable (Rising) register

Definition at line 49 of file pinint\_5411x.h.

#### 7.34.2.6 [\\_\\_IO uint32\\_t ISEL](#)

< PIN\_INT Structure Pin Interrupt Mode register

Definition at line 48 of file pinint\_5411x.h.

#### 7.34.2.7 [\\_\\_IO uint32\\_t IST](#)

Pin Interrupt Status register

Definition at line 57 of file pinint\_5411x.h.

**7.34.2.8** `__IO uint32_t PMCFG`

GPIO pattern match interrupt bit slice configuration register

Definition at line 60 of file `pinint_5411x.h`.

**7.34.2.9** `__IO uint32_t PMCTRL`

GPIO pattern match interrupt control register

Definition at line 58 of file `pinint_5411x.h`.

**7.34.2.10** `__IO uint32_t PMSRC`

GPIO pattern match interrupt bit-slice source register

Definition at line 59 of file `pinint_5411x.h`.

**7.34.2.11** `__IO uint32_t RISE`

Pin Interrupt Rising Edge register

Definition at line 55 of file `pinint_5411x.h`.

**7.34.2.12** `__O uint32_t SIENF`

Set Pin Interrupt Enable Falling Edge / Active Level register

Definition at line 53 of file `pinint_5411x.h`.

**7.34.2.13** `__O uint32_t SIENR`

Set Pin Interrupt Enable (Rising) register

Definition at line 50 of file `pinint_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pinint_5411x.h`

**7.35 LPC\_PMU\_T Struct Reference****7.35.1 Detailed Description**

PMU register block structure.

**Note**

Most of the PMU support is handled by the PMU library.

Definition at line 48 of file `pmu_5411x.h`.

```
#include "pmu_5411x.h"
```

## Data Fields

- `__I uint32_t RESERVED0` [17]
- `__IO uint32_t BODCTRL`

### 7.35.2 Field Documentation

#### 7.35.2.1 `__IO uint32_t BODCTRL`

Definition at line 50 of file `pmu_5411x.h`.

#### 7.35.2.2 `__I uint32_t RESERVED0`[17]

Definition at line 49 of file `pmu_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/pmu_5411x.h`

## 7.36 LPC\_ROM\_API\_T Struct Reference

### 7.36.1 Detailed Description

High level ROM API structure.

Definition at line 53 of file `romapi_5411x.h`.

```
#include "romapi_5411x.h"
```

## Data Fields

- `const uint32_t usbdApiBase`
- `const uint32_t reserved_clib`
- `const uint32_t reserved_power`
- `const uint32_t reserved_div`
- `const uint32_t reserved_usart`
- `const uint32_t reserved_i2cm`
- `const uint32_t reserved_i2cs`
- `const uint32_t reserved_i2cmon`
- `const uint32_t reserved_spim`
- `const uint32_t reserved_spis`
- `const uint32_t reserved_dmaaltd`
- `const uint32_t reserved_adcaltd`
- `const uint32_t reserved_uartalt`
- `const uint32_t reserved_flexcomm`

### 7.36.2 Field Documentation

#### 7.36.2.1 `const uint32_t reserved_adcaltd`

Reserved

Definition at line 65 of file `romapi_5411x.h`.



**7.36.2.2   const uint32\_t reserved\_clib**

Reserved

Definition at line 55 of file romapi\_5411x.h.

**7.36.2.3   const uint32\_t reserved\_div**

Reserved

Definition at line 57 of file romapi\_5411x.h.

**7.36.2.4   const uint32\_t reserved\_dmaaltd**

Reserved

Definition at line 64 of file romapi\_5411x.h.

**7.36.2.5   const uint32\_t reserved\_flexcomm**

Reserved

Definition at line 67 of file romapi\_5411x.h.

**7.36.2.6   const uint32\_t reserved\_i2cm**

Reserved

Definition at line 59 of file romapi\_5411x.h.

**7.36.2.7   const uint32\_t reserved\_i2cmon**

Reserved

Definition at line 61 of file romapi\_5411x.h.

**7.36.2.8   const uint32\_t reserved\_i2cs**

Reserved

Definition at line 60 of file romapi\_5411x.h.

**7.36.2.9   const uint32\_t reserved\_power**

Reserved

Definition at line 56 of file romapi\_5411x.h.

**7.36.2.10   const uint32\_t reserved\_spim**

Reserved

Definition at line 62 of file romapi\_5411x.h.

#### 7.36.2.11 `const uint32_t reserved_spis`

Reserved

Definition at line 63 of file `romapi_5411x.h`.

#### 7.36.2.12 `const uint32_t reserved_uartalt`

Reserved

Definition at line 66 of file `romapi_5411x.h`.

#### 7.36.2.13 `const uint32_t reserved_usart`

Reserved

Definition at line 58 of file `romapi_5411x.h`.

#### 7.36.2.14 `const uint32_t usbdApiBase`

USB API Base

Definition at line 54 of file `romapi_5411x.h`.

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

- [C:/Jenkins/LPC5411x/lpc5411x/chip\\_5411x/inc/romapi\\_5411x.h](#)

## 7.37 LPC\_RTC\_T Struct Reference

### 7.37.1 Detailed Description

LPC5411X Real Time clock register block structure.

Definition at line 47 of file `rtc_5411x.h`.

```
#include "rtc_5411x.h"
```

### Data Fields

- [\\_\\_IO uint32\\_t CTRL](#)
- [\\_\\_IO uint32\\_t MATCH](#)
- [\\_\\_IO uint32\\_t COUNT](#)
- [\\_\\_IO uint32\\_t WAKE](#)

### 7.37.2 Field Documentation

#### 7.37.2.1 `__IO uint32_t COUNT`

RTC counter register

Definition at line 50 of file `rtc_5411x.h`.

7.37.2.2 `__IO uint32_t CTRL`

< RTC RTC control register

Definition at line 48 of file `rtc_5411x.h`.

7.37.2.3 `__IO uint32_t MATCH`

PRTC match (alarm) register

Definition at line 49 of file `rtc_5411x.h`.

7.37.2.4 `__IO uint32_t WAKE`

RTC high-resolution/wake-up timer control register

Definition at line 51 of file `rtc_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/rtc_5411x.h`

## 7.38 LPC\_SCT\_T Struct Reference

### 7.38.1 Detailed Description

State Configurable Timer register block structure.

Definition at line 56 of file `sct_5411x.h`.

```
#include "sct_5411x.h"
```

#### Data Fields

- `__IO uint32_t CONFIG`
- union {
  - `__IO uint32_t CTRL_U`
  - struct {
    - `__IO uint16_t CTRL_L`
    - `__IO uint16_t CTRL_H`
- };
- `__IO uint16_t LIMIT_L`
- `__IO uint16_t LIMIT_H`
- `__IO uint16_t HALT_L`
- `__IO uint16_t HALT_H`
- `__IO uint16_t STOP_L`
- `__IO uint16_t STOP_H`
- `__IO uint16_t START_L`
- `__IO uint16_t START_H`
- `uint32_t RESERVED1 [10]`
- union {
  - `__IO uint32_t COUNT_U`
  - struct {
    - `__IO uint16_t COUNT_L`

```

    __IO uint16_t COUNT_H
}
};

• __IO uint16_t STATE_L
• __IO uint16_t STATE_H
• __IO uint32_t INPUT
• __IO uint16_t REGMODE_L
• __IO uint16_t REGMODE_H
• __IO uint32_t OUTPUT
• __IO uint32_t OUTPUTDIRCTRL
• __IO uint32_t RES
• __IO uint32_t DMA0REQUEST
• __IO uint32_t DMA1REQUEST
• uint32_t RESERVED2 [35]
• __IO uint32_t EVEN
• __IO uint32_t EVFLAG
• __IO uint32_t CONEN
• __IO uint32_t CONFLAG
• union {
    union {
        uint32_t U
        struct {
            uint16_t L
            uint16_t H
        }
    } MATCH [CONFIG_SCT_nRG]
    union {
        uint32_t U
        struct {
            uint16_t L
            uint16_t H
        }
    } CAP [CONFIG_SCT_nRG]
};

• uint32_t RESERVED3 [48+(16-CONFIG_SCT_nRG)]
• union {
    union {
        uint32_t U
        struct {
            uint16_t L
            uint16_t H
        }
    } MATCHREL [CONFIG_SCT_nRG]
    union {
        uint32_t U
        struct {
            uint16_t L
            uint16_t H
        }
    } CAPCTRL [CONFIG_SCT_nRG]
};

• uint32_t RESERVED6 [48+(16-CONFIG_SCT_nRG)]

```

- struct {
  - uint32\_t STATE
  - uint32\_t CTRL
 } EVENT [CONFIG\_SCT\_nEV]
- uint32\_t RESERVED9 [128-2 \*CONFIG\_SCT\_nEV]
- struct {
  - uint32\_t SET
  - uint32\_t CLR
 } OUT [CONFIG\_SCT\_nOU]
- uint32\_t RESERVED10 [191-2 \*CONFIG\_SCT\_nOU]
- \_\_IO uint32\_t MODULECONTENT

## 7.38.2 Field Documentation

7.38.2.1 union { ... }

7.38.2.2 union { ... }

7.38.2.3 union { ... }

7.38.2.4 union { ... }

7.38.2.5 \_\_I { ... } CAP[CONFIG\_SCT\_nRG]

7.38.2.6 \_\_IO { ... } CAPCTRL[CONFIG\_SCT\_nRG]

7.38.2.7 uint32\_t CLR

Output n Clear Register

Definition at line 166 of file sct\_5411x.h.

7.38.2.8 \_\_IO uint32\_t CONEN

conflict enable register

Definition at line 102 of file sct\_5411x.h.

7.38.2.9 \_\_IO uint32\_t CONFIG

Configuration Register

Definition at line 57 of file sct\_5411x.h.

7.38.2.10 \_\_IO uint32\_t CONFLAG

conflict flag register

Definition at line 103 of file sct\_5411x.h.

7.38.2.11 \_\_IO uint16\_t COUNT\_H

counter register for counter H

Definition at line 84 of file sct\_5411x.h.

#### 7.38.2.12 `__IO uint16_t COUNT_L`

counter register for counter L

Definition at line 83 of file `sct_5411x.h`.

#### 7.38.2.13 `__IO uint32_t COUNT_U`

counter register

Definition at line 80 of file `sct_5411x.h`.

#### 7.38.2.14 `uint32_t CTRL`

Definition at line 159 of file `sct_5411x.h`.

#### 7.38.2.15 `__IO uint16_t CTRL_H`

High control register

Definition at line 64 of file `sct_5411x.h`.

#### 7.38.2.16 `__IO uint16_t CTRL_L`

Low control register

Definition at line 63 of file `sct_5411x.h`.

#### 7.38.2.17 `__IO uint32_t CTRL_U`

Control Register

Definition at line 60 of file `sct_5411x.h`.

#### 7.38.2.18 `__IO uint32_t DMA0REQUEST`

DMA0 Request Register

Definition at line 97 of file `sct_5411x.h`.

#### 7.38.2.19 `__IO uint32_t DMA1REQUEST`

DMA1 Request Register

Definition at line 98 of file `sct_5411x.h`.

#### 7.38.2.20 `__IO uint32_t EVEN`

event enable register

Definition at line 100 of file `sct_5411x.h`.

7.38.2.21 `__IO { ... } EVENT[CONFIG_SCT_nEV]`

7.38.2.22 `__IO uint32_t EVFLAG`

event flag register

Definition at line 101 of file `sct_5411x.h`.

7.38.2.23 `uint16_t H`

`SCTMATCH[i].H` Access to H value

`SCTCAP[i].H` Access to H value

Definition at line 112 of file `sct_5411x.h`.

7.38.2.24 `__IO uint16_t HALT_H`

halt register for counter H

Definition at line 72 of file `sct_5411x.h`.

7.38.2.25 `__IO uint16_t HALT_L`

halt register for counter L

Definition at line 71 of file `sct_5411x.h`.

7.38.2.26 `__I uint32_t INPUT`

input register

Definition at line 91 of file `sct_5411x.h`.

7.38.2.27 `uint16_t L`

`SCTMATCH[i].L` Access to L value

`SCTCAP[i].L` Access to L value

Definition at line 111 of file `sct_5411x.h`.

7.38.2.28 `__IO uint16_t LIMIT_H`

limit register for counter H

Definition at line 70 of file `sct_5411x.h`.

7.38.2.29 `__IO uint16_t LIMIT_L`

limit register for counter L

Definition at line 69 of file `sct_5411x.h`.

7.38.2.30 `__IO { ... } MATCH[CONFIG_SCT_nRG]`

7.38.2.31 `__IO { ... } MATCHREL[CONFIG_SCT_nRG]`

7.38.2.32 `__I uint32_t MODULECONTENT`

0x7FC Module Content

Definition at line 170 of file sct\_5411x.h.

7.38.2.33 `__IO { ... } OUT[CONFIG_SCT_nOU]`

7.38.2.34 `__IO uint32_t OUTPUT`

output register

Definition at line 94 of file sct\_5411x.h.

7.38.2.35 `__IO uint32_t OUTPUTDIRCTRL`

output counter direction Control Register

Definition at line 95 of file sct\_5411x.h.

7.38.2.36 `__IO uint16_t REGMODE_H`

match - capture registers mode register H

Definition at line 93 of file sct\_5411x.h.

7.38.2.37 `__IO uint16_t REGMODE_L`

match - capture registers mode register L

Definition at line 92 of file sct\_5411x.h.

7.38.2.38 `__IO uint32_t RES`

conflict resolution register

Definition at line 96 of file sct\_5411x.h.

7.38.2.39 `uint32_t RESERVED1[10]`

0x03C reserved

Definition at line 77 of file sct\_5411x.h.

7.38.2.40 `uint32_t RESERVED10[191-2 *CONFIG_SCT_nOU]`

...-0x7F8 reserved

Definition at line 169 of file sct\_5411x.h.

7.38.2.41 `uint32_t RESERVED2[35]`

Definition at line 99 of file sct\_5411x.h.



**7.38.2.42    uint32\_t RESERVED3[48+(16-CONFIG\_SCT\_nRG)]**

Definition at line 129 of file sct\_5411x.h.

**7.38.2.43    uint32\_t RESERVED6[48+(16-CONFIG\_SCT\_nRG)]**

Definition at line 155 of file sct\_5411x.h.

**7.38.2.44    uint32\_t RESERVED9[128-2 \*CONFIG\_SCT\_nEV]**

...-0x4FC reserved

Definition at line 162 of file sct\_5411x.h.

**7.38.2.45    uint32\_t SET**

< 0x500-0x57C SCTOUT[i].SET / SCTOUT[i].CLR Output n Set Register

Definition at line 165 of file sct\_5411x.h.

**7.38.2.46    \_\_IO uint16\_t START\_H**

start register for counter H

Definition at line 76 of file sct\_5411x.h.

**7.38.2.47    \_\_IO uint16\_t START\_L**

start register for counter L

Definition at line 75 of file sct\_5411x.h.

**7.38.2.48    uint32\_t STATE**

Definition at line 158 of file sct\_5411x.h.

**7.38.2.49    \_\_IO uint16\_t STATE\_H**

state register for counter H

Definition at line 90 of file sct\_5411x.h.

**7.38.2.50    \_\_IO uint16\_t STATE\_L**

state register for counter L

Definition at line 89 of file sct\_5411x.h.

**7.38.2.51    \_\_IO uint16\_t STOP\_H**

stop register for counter H

Definition at line 74 of file sct\_5411x.h.

### 7.38.2.52 `__IO uint16_t STOP_L`

stop register for counter L

Definition at line 73 of file `sct_5411x.h`.

### 7.38.2.53 `uint32_t U`

< ... Match / Capture value SCTMATCH[i].U Unified 32-bit register

SCTCAP[i].U Unified 32-bit register

Definition at line 108 of file `sct_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/sct_5411x.h`

## 7.39 LPC\_SPI\_T Struct Reference

### 7.39.1 Detailed Description

SPI register block structure.

Definition at line 55 of file `spi_common_5411x.h`.

```
#include "spi_common_5411x.h"
```

### Data Fields

- `__I uint32_t RESERVED0` [256]
- `__IO uint32_t CFG`
- `__IO uint32_t DLY`
- `__IO uint32_t STAT`
- `__IO uint32_t INTENSET`
- `__IO uint32_t INTENCLR`
- `__I uint32_t RESERVED1` [4]
- `__IO uint32_t DIV`
- `__IO uint32_t INTSTAT`
- `__IO uint32_t FIFOCFG`
- `__IO uint32_t FIFOSTAT`
- `__IO uint32_t FIFOTRIG`
- `__IO uint32_t FIFOINTENSET`
- `__IO uint32_t FIFOINTENCLR`
- `__IO uint32_t FIFOINTSTAT`
- `__O uint32_t FIFOWR`
- `__I uint32_t FIFORD`
- `__I uint32_t FIFORDNOPOP`
- `__IO uint32_t PSELID`
- `__I uint32_t PID`

### 7.39.2 Field Documentation

#### 7.39.2.1 `__IO uint32_t CFG`

Offset: 0x400 SPI Configuration register

Definition at line 58 of file `spi_common_5411x.h`.

#### 7.39.2.2 `__IO uint32_t DIV`

Offset: 0x424 SPI clock Divider register

Definition at line 64 of file spi\_common\_5411x.h.

#### 7.39.2.3 `__IO uint32_t DLY`

Offset: 0x404 SPI Delay register

Definition at line 59 of file spi\_common\_5411x.h.

#### 7.39.2.4 `__IO uint32_t FIFOCFG`

Offset: 0xE00 FIFO Configuration register

Definition at line 69 of file spi\_common\_5411x.h.

#### 7.39.2.5 `__IO uint32_t FIFOINTENCLR`

Offset: 0xE14 FIFO Interrupt enable CLEAR register

Definition at line 74 of file spi\_common\_5411x.h.

#### 7.39.2.6 `__IO uint32_t FIFOINTENSET`

Offset: 0xE10 FIFO Interrupt enable SET register

Definition at line 73 of file spi\_common\_5411x.h.

#### 7.39.2.7 `__IO uint32_t FIFOINTSTAT`

Offset: 0xE18 FIFO Interrupt Status register

Definition at line 75 of file spi\_common\_5411x.h.

#### 7.39.2.8 `__I uint32_t FIFORD`

Offset: 0xE30 FIFO Data read register

Definition at line 79 of file spi\_common\_5411x.h.

#### 7.39.2.9 `__I uint32_t FIFORDNOPOP`

Offset: 0xE40 FIFO Data peek (read without popping out of queue) register

Definition at line 81 of file spi\_common\_5411x.h.

#### 7.39.2.10 `__IO uint32_t FIFOSTAT`

Offset: 0xE04 FIFO Status register

Definition at line 70 of file spi\_common\_5411x.h.

#### 7.39.2.11 `__IO uint32_t FIFOTRIG`

Offset: 0xE08 FIFO Trigger level register

Definition at line 71 of file spi\_common\_5411x.h.

#### 7.39.2.12 `__O uint32_t FIFOWR`

Offset: 0xE20 FIFO Data write register

Definition at line 77 of file spi\_common\_5411x.h.

#### 7.39.2.13 `__IO uint32_t INTENCLR`

Offset: 0x410 SPI Interrupt Enable Clear register

Definition at line 62 of file spi\_common\_5411x.h.

#### 7.39.2.14 `__IO uint32_t INTENSET`

Offset: 0x40c SPI Interrupt Enable Set register

Definition at line 61 of file spi\_common\_5411x.h.

#### 7.39.2.15 `__IO uint32_t INTSTAT`

Offset: 0x428 SPI Interrupt Status register

Definition at line 65 of file spi\_common\_5411x.h.

#### 7.39.2.16 `__I uint32_t PID`

Offset: 0xFFC Module identification register

Definition at line 86 of file spi\_common\_5411x.h.

#### 7.39.2.17 `__IO uint32_t PSELID`

Offset: 0xFF8 Peripheral select/identification register

Definition at line 85 of file spi\_common\_5411x.h.

#### 7.39.2.18 `__I uint32_t RESERVED0[256]`

< SPI Structure SPI registers

Definition at line 57 of file spi\_common\_5411x.h.

#### 7.39.2.19 `__I uint32_t RESERVED1[4]`

Definition at line 63 of file spi\_common\_5411x.h.

## 7.39.2.20 \_\_IO uint32\_t STAT

Offset: 0x408 SPI Status register

Definition at line 60 of file spi\_common\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/spi\_common\_5411x.h

## 7.40 LPC\_SYSCON\_T Struct Reference

### 7.40.1 Detailed Description

LPC5411X Main system configuration register block structure.

Definition at line 47 of file syscon\_5411x.h.

```
#include "syscon_5411x.h"
```

### Data Fields

- \_\_IO uint32\_t SYSMEMREMAP
- \_\_I uint32\_t RESERVED0 [3]
- \_\_IO uint32\_t AHBMATPRIO
- \_\_I uint32\_t RESERVED1 [11]
- \_\_IO uint32\_t SYSTCKCAL
- \_\_I uint32\_t RESERVED2 [1]
- \_\_IO uint32\_t NMISRC
- \_\_IO uint32\_t ASYNCAPBCTRL
- \_\_I uint32\_t RESERVED3 [28]
- \_\_I uint32\_t PIOPORCAP [2]
- \_\_I uint32\_t RESERVED4 [2]
- \_\_I uint32\_t PIORESCAP [2]
- \_\_I uint32\_t RESERVED5 [10]
- \_\_IO uint32\_t PRESETCTRL [2]
- \_\_I uint32\_t RESERVED6 [6]
- \_\_O uint32\_t PRESETCTRLSET [2]
- \_\_I uint32\_t RESERVED7 [6]
- \_\_O uint32\_t PRESETCTRLCLR [2]
- \_\_I uint32\_t RESERVED8 [42]
- \_\_IO uint32\_t SYSRSTSTAT
- \_\_I uint32\_t RESERVED9 [3]
- \_\_IO uint32\_t AHBCLKCTRL [2]
- \_\_I uint32\_t RESERVED10 [6]
- \_\_O uint32\_t AHBCLKCTRLSET [2]
- \_\_I uint32\_t RESERVED11 [6]
- \_\_O uint32\_t AHBCLKCTRLCLR [2]
- \_\_I uint32\_t RESERVED12 [14]
- \_\_IO uint32\_t MAINCLKSELA
- \_\_IO uint32\_t MAINCLKSELB
- \_\_IO uint32\_t CLKOUTSELA
- \_\_I uint32\_t RESERVED13
- \_\_IO uint32\_t SYSPLLCLKSEL
- \_\_I uint32\_t RESERVED14 [3]

- [\\_\\_IO uint32\\_t SPIFICKSEL](#)
- [\\_\\_IO uint32\\_t ADCCLKSEL](#)
- [\\_\\_IO uint32\\_t USBCLKSEL](#)
- [\\_\\_I uint32\\_t RESERVED15](#)
- [\\_\\_IO uint32\\_t FXCOMCLKSEL](#) [8]
- [\\_\\_I uint32\\_t RESERVED16](#) [4]
- [\\_\\_IO uint32\\_t MCLKCLKSEL](#)
- [\\_\\_I uint32\\_t RESERVED16A](#)
- [\\_\\_IO uint32\\_t FRGCLKSEL](#)
- [\\_\\_IO uint32\\_t DMICCLKSEL](#)
- [\\_\\_I uint32\\_t RESERVED17](#) [4]
- [\\_\\_IO uint32\\_t SYSTICKCLKDIV](#)
- [\\_\\_I uint32\\_t RESERVED18](#) [31]
- [\\_\\_IO uint32\\_t AHBCLKDIV](#)
- [\\_\\_IO uint32\\_t CLKOUTDIV](#)
- [\\_\\_I uint32\\_t RESERVED19](#) [2]
- [\\_\\_IO uint32\\_t SPIFICKDIV](#)
- [\\_\\_IO uint32\\_t ADCCLKDIV](#)
- [\\_\\_IO uint32\\_t USBCLKDIV](#)
- [\\_\\_I uint32\\_t RESERVED20](#)
- [\\_\\_IO uint32\\_t FRGCTRL](#)
- [\\_\\_I uint32\\_t RESERVED21](#)
- [\\_\\_IO uint32\\_t DMICCLKDIV](#)
- [\\_\\_IO uint32\\_t MCLKDIV](#)
- [\\_\\_I uint32\\_t RESERVED22](#) [20]
- [\\_\\_IO uint32\\_t FLASHCFG](#)
- [\\_\\_I uint32\\_t RESERVED23](#) [2]
- [\\_\\_IO uint32\\_t USBCLKCTRL](#)
- [\\_\\_IO uint32\\_t USBCLKSTAT](#)
- [\\_\\_I uint32\\_t RESERVED24](#)
- [\\_\\_IO uint32\\_t FREQMCTRL](#)
- [\\_\\_I uint32\\_t RESERVED25](#)
- [\\_\\_IO uint32\\_t MCLKIO](#)
- [\\_\\_I uint32\\_t RESERVED26](#) [55]
- [\\_\\_IO uint32\\_t FROCTRL](#)
- [\\_\\_I uint32\\_t RESERVED27](#)
- [\\_\\_IO uint32\\_t WDTOSCCTRL](#)
- [\\_\\_IO uint32\\_t RTCOSCCTRL](#)
- [\\_\\_I uint32\\_t RESERVED28](#) [28]
- [\\_\\_IO uint32\\_t SYSPLLCTRL](#)
- [\\_\\_I uint32\\_t SYSPLLSTAT](#)
- [\\_\\_IO uint32\\_t SYSPLLNDEC](#)
- [\\_\\_IO uint32\\_t SYSPLLPDEC](#)
- [\\_\\_IO uint32\\_t SYSPLLSSCTRL](#) [2]
- [\\_\\_I uint32\\_t RESERVED29](#) [30]
- [\\_\\_IO uint32\\_t PDRUNCFG](#) [2]
- [\\_\\_I uint32\\_t RESERVED30](#) [2]
- [\\_\\_O uint32\\_t PDRUNCFGSET](#) [2]
- [\\_\\_I uint32\\_t RESERVED31](#) [2]
- [\\_\\_O uint32\\_t PDRUNCFGCLR](#) [2]
- [\\_\\_I uint32\\_t RESERVED32](#) [18]
- [\\_\\_IO uint32\\_t STARTERP](#) [2]
- [\\_\\_I uint32\\_t RESERVED33](#) [6]
- [\\_\\_O uint32\\_t STARTERPSET](#) [2]
- [\\_\\_I uint32\\_t RESERVED34](#) [6]

- `__O uint32_t STARTERPCLR` [2]
- `__I uint32_t RESERVED35` [78]
- `__IO uint32_t CPCTRL`
- `__IO uint32_t CPBOOT`
- `__IO uint32_t CPSTACK`
- `__I uint32_t CPSTAT`
- `__I uint32_t RESERVED36` [381]
- `__IO uint32_t AUTOCGOR`
- `__I uint32_t RESERVED37` [123]
- `__I uint32_t JTAGIDCODE`
- `__I uint32_t DEVICE_ID` [2]

## 7.40.2 Field Documentation

### 7.40.2.1 `__IO uint32_t ADCCLKDIV`

ADC Clock divider register

Definition at line 98 of file `syscon_5411x.h`.

### 7.40.2.2 `__IO uint32_t ADCCLKSEL`

ADC Async Clk Sel register

Definition at line 82 of file `syscon_5411x.h`.

### 7.40.2.3 `__IO uint32_t AHBCLKCTRL[2]`

AHB Peripheral Clk Enable register

Definition at line 69 of file `syscon_5411x.h`.

### 7.40.2.4 `__O uint32_t AHBCLKCTRLCLR[2]`

AHB Peripheral Clk Enable Clr register

Definition at line 73 of file `syscon_5411x.h`.

### 7.40.2.5 `__O uint32_t AHBCLKCTRLSET[2]`

AHB Peripheral Clk Enable Set register

Definition at line 71 of file `syscon_5411x.h`.

### 7.40.2.6 `__IO uint32_t AHBCLKDIV`

AHB Clock divider

Definition at line 94 of file `syscon_5411x.h`.

### 7.40.2.7 `__IO uint32_t AHBMATPRIO`

AHB Martix priority register

Definition at line 50 of file `syscon_5411x.h`.

#### 7.40.2.8 `__IO uint32_t ASYNCAPBCTRL`

Asynch APB chiplet control register

Definition at line 55 of file syscon\_5411x.h.

#### 7.40.2.9 `__IO uint32_t AUTOCGOR`

Definition at line 143 of file syscon\_5411x.h.

#### 7.40.2.10 `__IO uint32_t CLKOUTDIV`

CLKOUT divider

Definition at line 95 of file syscon\_5411x.h.

#### 7.40.2.11 `__IO uint32_t CLKOUTSELA`

Clk Out Sel Source B register

Definition at line 77 of file syscon\_5411x.h.

#### 7.40.2.12 `__IO uint32_t CPBOOT`

Coprocessor boot address

Definition at line 139 of file syscon\_5411x.h.

#### 7.40.2.13 `__IO uint32_t CPCTRL`

Coprocessor control register

Definition at line 138 of file syscon\_5411x.h.

#### 7.40.2.14 `__IO uint32_t CPSTACK`

Coprocessor stack address register

Definition at line 140 of file syscon\_5411x.h.

#### 7.40.2.15 `__I uint32_t CPSTAT`

Coprocessor status register

Definition at line 141 of file syscon\_5411x.h.

#### 7.40.2.16 `__I uint32_t DEVICE_ID[2]`

Device ID Registers

Definition at line 146 of file syscon\_5411x.h.



**7.40.2.17** `__IO uint32_t DMICCLKDIV`

DMIC Clock divider register

Definition at line 103 of file syscon\_5411x.h.

**7.40.2.18** `__IO uint32_t DMICCLKSEL`

DMIC Clock select register

Definition at line 90 of file syscon\_5411x.h.

**7.40.2.19** `__IO uint32_t FLASHCFG`

Flash wait state configuration register

Definition at line 106 of file syscon\_5411x.h.

**7.40.2.20** `__IO uint32_t FREQMECTRL`

Frequency measure register

Definition at line 111 of file syscon\_5411x.h.

**7.40.2.21** `__IO uint32_t FRGCLKSEL`

FRG Clock select register

Definition at line 89 of file syscon\_5411x.h.

**7.40.2.22** `__IO uint32_t FRGCTRL`

Fraction Rate Generator Ctrl register

Definition at line 101 of file syscon\_5411x.h.

**7.40.2.23** `__IO uint32_t FROCTRL`

FRO oscillator control register

Definition at line 115 of file syscon\_5411x.h.

**7.40.2.24** `__IO uint32_t FXCOMCLKSEL[8]`

FlexCOM CLK sel register

Definition at line 85 of file syscon\_5411x.h.

**7.40.2.25** `__I uint32_t JTAGIDCODE`

JTAG ID Code register

Definition at line 145 of file syscon\_5411x.h.

#### 7.40.2.26 `__IO uint32_t MAINCLKSELA`

Main Clk sel Source Sel A register

Definition at line 75 of file syscon\_5411x.h.

#### 7.40.2.27 `__IO uint32_t MAINCLKSELB`

Main Clk sel Source Sel B register

Definition at line 76 of file syscon\_5411x.h.

#### 7.40.2.28 `__IO uint32_t MCLKCLKSEL`

MCLK Clock select register

Definition at line 87 of file syscon\_5411x.h.

#### 7.40.2.29 `__IO uint32_t MCLKDIV`

I2S MClock divider register

Definition at line 104 of file syscon\_5411x.h.

#### 7.40.2.30 `__IO uint32_t MCLKIO`

MCLK Input Output register

Definition at line 113 of file syscon\_5411x.h.

#### 7.40.2.31 `__IO uint32_t NMISRC`

NMI Source select register

Definition at line 54 of file syscon\_5411x.h.

#### 7.40.2.32 `__IO uint32_t PDRUNCFG[2]`

Power Down configuration registers

Definition at line 126 of file syscon\_5411x.h.

#### 7.40.2.33 `__O uint32_t PDRUNCFGCLR[2]`

Power down configuartion clear register

Definition at line 130 of file syscon\_5411x.h.

#### 7.40.2.34 `__O uint32_t PDRUNCFGSET[2]`

Power down configuartion set register

Definition at line 128 of file syscon\_5411x.h.

**7.40.2.35** `__I uint32_t PIOPORCAP[2]`

Power on Reset; port capture register

Definition at line 57 of file syscon\_5411x.h.

**7.40.2.36** `__I uint32_t PIORESCAP[2]`

Reset; port capture register

Definition at line 59 of file syscon\_5411x.h.

**7.40.2.37** `__IO uint32_t PRESETCTRL[2]`

Peripheral Reset control

Definition at line 61 of file syscon\_5411x.h.

**7.40.2.38** `__O uint32_t PRESETCTRLCLR[2]`

Peripheral Reset Control set

Definition at line 65 of file syscon\_5411x.h.

**7.40.2.39** `__O uint32_t PRESETCTRLSET[2]`

Peripheral Reset Control set

Definition at line 63 of file syscon\_5411x.h.

**7.40.2.40** `__I uint32_t RESERVED0[3]`

Definition at line 49 of file syscon\_5411x.h.

**7.40.2.41** `__I uint32_t RESERVED1[11]`

Definition at line 51 of file syscon\_5411x.h.

**7.40.2.42** `__I uint32_t RESERVED10[6]`

Definition at line 70 of file syscon\_5411x.h.

**7.40.2.43** `__I uint32_t RESERVED11[6]`

Definition at line 72 of file syscon\_5411x.h.

**7.40.2.44** `__I uint32_t RESERVED12[14]`

Definition at line 74 of file syscon\_5411x.h.

**7.40.2.45** `__I uint32_t RESERVED13`

Definition at line 78 of file syscon\_5411x.h.

#### 7.40.2.46 `__I uint32_t RESERVED14[3]`

Definition at line 80 of file syscon\_5411x.h.

#### 7.40.2.47 `__I uint32_t RESERVED15`

Definition at line 84 of file syscon\_5411x.h.

#### 7.40.2.48 `__I uint32_t RESERVED16[4]`

Definition at line 86 of file syscon\_5411x.h.

#### 7.40.2.49 `__I uint32_t RESERVED16A`

Definition at line 88 of file syscon\_5411x.h.

#### 7.40.2.50 `__I uint32_t RESERVED17[4]`

Definition at line 91 of file syscon\_5411x.h.

#### 7.40.2.51 `__I uint32_t RESERVED18[31]`

Definition at line 93 of file syscon\_5411x.h.

#### 7.40.2.52 `__I uint32_t RESERVED19[2]`

Definition at line 96 of file syscon\_5411x.h.

#### 7.40.2.53 `__I uint32_t RESERVED2[1]`

Definition at line 53 of file syscon\_5411x.h.

#### 7.40.2.54 `__I uint32_t RESERVED20`

Definition at line 100 of file syscon\_5411x.h.

#### 7.40.2.55 `__I uint32_t RESERVED21`

Definition at line 102 of file syscon\_5411x.h.

#### 7.40.2.56 `__I uint32_t RESERVED22[20]`

Definition at line 105 of file syscon\_5411x.h.

#### 7.40.2.57 `__I uint32_t RESERVED23[2]`

Definition at line 107 of file syscon\_5411x.h.

**7.40.2.58** `__I uint32_t RESERVED24`

Definition at line 110 of file syscon\_5411x.h.

**7.40.2.59** `__I uint32_t RESERVED25`

Definition at line 112 of file syscon\_5411x.h.

**7.40.2.60** `__I uint32_t RESERVED26[55]`

Definition at line 114 of file syscon\_5411x.h.

**7.40.2.61** `__I uint32_t RESERVED27`

Definition at line 116 of file syscon\_5411x.h.

**7.40.2.62** `__I uint32_t RESERVED28[28]`

Definition at line 119 of file syscon\_5411x.h.

**7.40.2.63** `__I uint32_t RESERVED29[30]`

Definition at line 125 of file syscon\_5411x.h.

**7.40.2.64** `__I uint32_t RESERVED3[28]`

Definition at line 56 of file syscon\_5411x.h.

**7.40.2.65** `__I uint32_t RESERVED30[2]`

Definition at line 127 of file syscon\_5411x.h.

**7.40.2.66** `__I uint32_t RESERVED31[2]`

Definition at line 129 of file syscon\_5411x.h.

**7.40.2.67** `__I uint32_t RESERVED32[18]`

Definition at line 131 of file syscon\_5411x.h.

**7.40.2.68** `__I uint32_t RESERVED33[6]`

Definition at line 133 of file syscon\_5411x.h.

**7.40.2.69** `__I uint32_t RESERVED34[6]`

Definition at line 135 of file syscon\_5411x.h.

#### 7.40.2.70 `__I uint32_t RESERVED35[78]`

Definition at line 137 of file syscon\_5411x.h.

#### 7.40.2.71 `__I uint32_t RESERVED36[381]`

Definition at line 142 of file syscon\_5411x.h.

#### 7.40.2.72 `__I uint32_t RESERVED37[123]`

Definition at line 144 of file syscon\_5411x.h.

#### 7.40.2.73 `__I uint32_t RESERVED4[2]`

Definition at line 58 of file syscon\_5411x.h.

#### 7.40.2.74 `__I uint32_t RESERVED5[10]`

Definition at line 60 of file syscon\_5411x.h.

#### 7.40.2.75 `__I uint32_t RESERVED6[6]`

Definition at line 62 of file syscon\_5411x.h.

#### 7.40.2.76 `__I uint32_t RESERVED7[6]`

Definition at line 64 of file syscon\_5411x.h.

#### 7.40.2.77 `__I uint32_t RESERVED8[42]`

Definition at line 66 of file syscon\_5411x.h.

#### 7.40.2.78 `__I uint32_t RESERVED9[3]`

Definition at line 68 of file syscon\_5411x.h.

#### 7.40.2.79 `__IO uint32_t RTCOSCCTRL`

RTC Oscillator control register

Definition at line 118 of file syscon\_5411x.h.

#### 7.40.2.80 `__IO uint32_t SPIFICKDIV`

SPIFI clock divider register

Definition at line 97 of file syscon\_5411x.h.

**7.40.2.81 \_\_IO uint32\_t SPIFICKSEL**

SPIFI clock selection register

Definition at line 81 of file syscon\_5411x.h.

**7.40.2.82 \_\_IO uint32\_t STARTERP[2]**

Start logic wakeup enable register

Definition at line 132 of file syscon\_5411x.h.

**7.40.2.83 \_\_O uint32\_t STARTERPCLR[2]**

Start logic wakeup enable clear register

Definition at line 136 of file syscon\_5411x.h.

**7.40.2.84 \_\_O uint32\_t STARTERPSET[2]**

Start logic wakeup enable set register

Definition at line 134 of file syscon\_5411x.h.

**7.40.2.85 \_\_IO uint32\_t SYSMEMREMAP**

System Remap register

Definition at line 48 of file syscon\_5411x.h.

**7.40.2.86 \_\_IO uint32\_t SYSPLLCLKSEL**

System PLL Clk Selregister

Definition at line 79 of file syscon\_5411x.h.

**7.40.2.87 \_\_IO uint32\_t SYSPLLCTRL**

System PLL control register

Definition at line 120 of file syscon\_5411x.h.

**7.40.2.88 \_\_IO uint32\_t SYSPLLNDEC**

System PLL N-DEC register

Definition at line 122 of file syscon\_5411x.h.

**7.40.2.89 \_\_IO uint32\_t SYSPLLPDEC**

System PLL P-DEC register

Definition at line 123 of file syscon\_5411x.h.

**7.40.2.90   \_\_IO uint32\_t SYSPLLSSCTRL[2]**

System PLL Spread-Spectrum control register

Definition at line 124 of file syscon\_5411x.h.

**7.40.2.91   \_\_IO uint32\_t SYSPLLSTAT**

System PLL status register

Definition at line 121 of file syscon\_5411x.h.

**7.40.2.92   \_\_IO uint32\_t SYSRSTSTAT**

System Reset Stat register

Definition at line 67 of file syscon\_5411x.h.

**7.40.2.93   \_\_IO uint32\_t SYSTCKCAL**

System Tick Calibration register

Definition at line 52 of file syscon\_5411x.h.

**7.40.2.94   \_\_IO uint32\_t SYSTICKCLKDIV**

Systick Clock divider register

Definition at line 92 of file syscon\_5411x.h.

**7.40.2.95   \_\_IO uint32\_t USBCLKCTRL**

USB Clock control register

Definition at line 108 of file syscon\_5411x.h.

**7.40.2.96   \_\_IO uint32\_t USBCLKDIV**

USB Clock divider register

Definition at line 99 of file syscon\_5411x.h.

**7.40.2.97   \_\_IO uint32\_t USBCLKSEL**

USB Async Clk Sel register

Definition at line 83 of file syscon\_5411x.h.

**7.40.2.98   \_\_IO uint32\_t USBCLKSTAT**

USB Clock Status register

Definition at line 109 of file syscon\_5411x.h.



7.40.2.99 `__IO uint32_t WDTOSCCTRL`

Watchdog Oscillator control

Definition at line 117 of file `syscon_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/syscon_5411x.h`

## 7.41 LPC\_TIMER\_T Struct Reference

### 7.41.1 Detailed Description

32-bit Standard timer register block structure

Definition at line 47 of file `timer_5411x.h`.

```
#include "timer_5411x.h"
```

#### Data Fields

- `__IO uint32_t IR`
- `__IO uint32_t TCR`
- `__IO uint32_t TC`
- `__IO uint32_t PR`
- `__IO uint32_t PC`
- `__IO uint32_t MCR`
- `__IO uint32_t MR [4]`
- `__IO uint32_t CCR`
- `__IO uint32_t CR [4]`
- `__IO uint32_t EMR`
- `__IO uint32_t RESERVED0 [12]`
- `__IO uint32_t CTCR`
- `__IO uint32_t PWMC`

### 7.41.2 Field Documentation

#### 7.41.2.1 `__IO uint32_t CCR`

Capture Control Register. The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place.

Definition at line 55 of file `timer_5411x.h`.

#### 7.41.2.2 `__IO uint32_t CR[4]`

Capture Register. CR is loaded with the value of TC when there is an event on the CAPn.0 input.

Definition at line 56 of file `timer_5411x.h`.

#### 7.41.2.3 `__IO uint32_t CTCR`

Count Control Register. The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting.

Definition at line 59 of file `timer_5411x.h`.

#### 7.41.2.4 `__IO uint32_t EMR`

External Match Register. The EMR controls the external match pins MATn.0-3 (MAT0.0-3 and MAT1.0-3 respectively).

Definition at line 57 of file timer\_5411x.h.

#### 7.41.2.5 `__IO uint32_t IR`

< TIMERN Structure Interrupt Register. The IR can be written to clear interrupts. The IR can be read to identify which of eight possible interrupt sources are pending.

Definition at line 48 of file timer\_5411x.h.

#### 7.41.2.6 `__IO uint32_t MCR`

Match Control Register. The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.

Definition at line 53 of file timer\_5411x.h.

#### 7.41.2.7 `__IO uint32_t MR[4]`

Match Register. MR can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR matches the TC.

Definition at line 54 of file timer\_5411x.h.

#### 7.41.2.8 `__IO uint32_t PC`

Prescale Counter. The 32 bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented and the PC is cleared. The PC is observable and controllable through the bus interface.

Definition at line 52 of file timer\_5411x.h.

#### 7.41.2.9 `__IO uint32_t PR`

Prescale Register. The Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC.

Definition at line 51 of file timer\_5411x.h.

#### 7.41.2.10 `__IO uint32_t PWM`

Definition at line 60 of file timer\_5411x.h.

#### 7.41.2.11 `__I uint32_t RESERVED0[12]`

Definition at line 58 of file timer\_5411x.h.

#### 7.41.2.12 `__IO uint32_t TC`

Timer Counter. The 32 bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.

Definition at line 50 of file timer\_5411x.h.

7.41.2.13 `__IO uint32_t TCR`

Timer Control Register. The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.

Definition at line 49 of file `timer_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/timer_5411x.h`

## 7.42 LPC\_USART\_T Struct Reference

### 7.42.1 Detailed Description

UART Registers.

Definition at line 53 of file `uart_5411x.h`.

```
#include "uart_5411x.h"
```

#### Data Fields

- `__IO uint32_t CFG`
- `__IO uint32_t CTL`
- `__IO uint32_t STAT`
- `__IO uint32_t INTENSET`
- `__O uint32_t INTENCLR`
- `__IO uint32_t BRG`
- `__I uint32_t INTSTAT`
- `__IO uint32_t OSR`
- `__IO uint32_t ADDR`
- `__IO uint32_t FIFOCFG`
- `__IO uint32_t FIFOSTAT`
- `__IO uint32_t FIFOTRIG`
- `__IO uint32_t FIFOINTENSET`
- `__IO uint32_t FIFOINTENCLR`
- `__IO uint32_t FIFOINTSTAT`
- `__O uint32_t FIFOWR`
- `__I uint32_t FIFORD`
- `__I uint32_t FIFORDNOPOP`
- `__IO uint32_t PSELID`
- `__I uint32_t PID`

### 7.42.2 Field Documentation

7.42.2.1 `__IO uint32_t ADDR`

Offset: 0x02C Address register (for automatic address matching)

Definition at line 63 of file `uart_5411x.h`.

7.42.2.2 `__IO uint32_t BRG`

Offset: 0x020 Baud Rate Generator register

Definition at line 60 of file `uart_5411x.h`.

#### 7.42.2.3 `__IO uint32_t CFG`

Offset: 0x000 Configuration register

Definition at line 54 of file `uart_5411x.h`.

#### 7.42.2.4 `__IO uint32_t CTL`

Offset: 0x004 Control register

Definition at line 55 of file `uart_5411x.h`.

#### 7.42.2.5 `__IO uint32_t FIFOCFG`

Offset: 0xE00 FIFO Configuration register

Definition at line 67 of file `uart_5411x.h`.

#### 7.42.2.6 `__IO uint32_t FIFOINTENCLR`

Offset: 0xE14 FIFO Interrupt enable CLEAR register

Definition at line 72 of file `uart_5411x.h`.

#### 7.42.2.7 `__IO uint32_t FIFOINTENSET`

Offset: 0xE10 FIFO Interrupt enable SET register

Definition at line 71 of file `uart_5411x.h`.

#### 7.42.2.8 `__IO uint32_t FIFOINTSTAT`

Offset: 0xE18 FIFO Interrupt Status register

Definition at line 73 of file `uart_5411x.h`.

#### 7.42.2.9 `__I uint32_t FIFORD`

Offset: 0xE30 FIFO Data read register

Definition at line 77 of file `uart_5411x.h`.

#### 7.42.2.10 `__I uint32_t FIFORDNOPOP`

Offset: 0xE40 FIFO Data peek (read without popping out of queue) register

Definition at line 79 of file `uart_5411x.h`.

#### 7.42.2.11 `__IO uint32_t FIFOSTAT`

Offset: 0xE04 FIFO Status register

Definition at line 68 of file `uart_5411x.h`.

**7.42.2.12** `__IO uint32_t FIFOTRIG`

Offset: 0xE08 FIFO Trigger level register

Definition at line 69 of file `uart_5411x.h`.

**7.42.2.13** `__O uint32_t FIFOWR`

Offset: 0xE20 FIFO Data write register

Definition at line 75 of file `uart_5411x.h`.

**7.42.2.14** `__O uint32_t INTENCLR`

Offset: 0x010 Interrupt Enable Clear register

Definition at line 58 of file `uart_5411x.h`.

**7.42.2.15** `__IO uint32_t INTENSET`

Offset: 0x00C Interrupt Enable Read and Set register

Definition at line 57 of file `uart_5411x.h`.

**7.42.2.16** `__I uint32_t INTSTAT`

Offset: 0x024 Interrupt Status register

Definition at line 61 of file `uart_5411x.h`.

**7.42.2.17** `__IO uint32_t OSR`

Offset: 0x028 Oversampling register

Definition at line 62 of file `uart_5411x.h`.

**7.42.2.18** `__I uint32_t PID`

Offset: 0xFFC Module identification register

Definition at line 84 of file `uart_5411x.h`.

**7.42.2.19** `__IO uint32_t PSELID`

Offset: 0xFF8 Peripheral select/identification register

Definition at line 83 of file `uart_5411x.h`.

**7.42.2.20** `__IO uint32_t STAT`

Offset: 0x008 Status register

Definition at line 56 of file `uart_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/uart_5411x.h`

## 7.43 LPC\_UTICK\_T Struct Reference

### 7.43.1 Detailed Description

Micro Tick register block structure.

Definition at line 47 of file utick\_5411x.h.

```
#include "utick_5411x.h"
```

#### Data Fields

- [\\_\\_IO uint32\\_t CTRL](#)
- [\\_\\_IO uint32\\_t STATUS](#)

### 7.43.2 Field Documentation

#### 7.43.2.1 [\\_\\_IO uint32\\_t CTRL](#)

UTick Control register

Definition at line 48 of file utick\_5411x.h.

#### 7.43.2.2 [\\_\\_IO uint32\\_t STATUS](#)

UTick Status register

Definition at line 49 of file utick\_5411x.h.

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

- [C:/Jenkins/LPC5411x/lpc5411x/chip\\_5411x/inc/utick\\_5411x.h](#)

## 7.44 LPC\_WWDT\_T Struct Reference

### 7.44.1 Detailed Description

Windowed Watchdog register block structure.

Definition at line 47 of file wwdt\_5411x.h.

```
#include "wwdt_5411x.h"
```

#### Data Fields

- [\\_\\_IO uint32\\_t MOD](#)
- [\\_\\_IO uint32\\_t TC](#)
- [\\_\\_O uint32\\_t FEED](#)
- [\\_\\_I uint32\\_t TV](#)
- [\\_\\_I uint32\\_t RESERVED0](#)
- [\\_\\_IO uint32\\_t WARNINT](#)
- [\\_\\_IO uint32\\_t WINDOW](#)

### 7.44.2 Field Documentation

#### 7.44.2.1 `__O uint32_t FEED`

Watchdog feed sequence register. Writing 0xAA followed by 0x55 to this register reloads the Watchdog timer with the value contained in WDTC.

Definition at line 50 of file `wwdt_5411x.h`.

#### 7.44.2.2 `__IO uint32_t MOD`

< WWDT Structure Watchdog mode register. This register contains the basic mode and status of the Watchdog Timer.

Definition at line 48 of file `wwdt_5411x.h`.

#### 7.44.2.3 `__I uint32_t RESERVED0`

Definition at line 52 of file `wwdt_5411x.h`.

#### 7.44.2.4 `__IO uint32_t TC`

Watchdog timer constant register. This register determines the time-out value.

Definition at line 49 of file `wwdt_5411x.h`.

#### 7.44.2.5 `__I uint32_t TV`

Watchdog timer value register. This register reads out the current value of the Watchdog timer.

Definition at line 51 of file `wwdt_5411x.h`.

#### 7.44.2.6 `__IO uint32_t WARNINT`

Watchdog warning interrupt register. This register contains the Watchdog warning interrupt compare value.

Definition at line 53 of file `wwdt_5411x.h`.

#### 7.44.2.7 `__IO uint32_t WINDOW`

Watchdog timer window register. This register contains the Watchdog window value.

Definition at line 54 of file `wwdt_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/wwdt_5411x.h`

## 7.45 NVIC\_Type Struct Reference

### 7.45.1 Detailed Description

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

Definition at line 291 of file `core_cm0plus.h`.

```
#include "core_cm0plus.h"
```

## Data Fields

- [\\_\\_IO uint32\\_t ISER](#) [1]
- uint32\_t [RESERVED0](#) [31]
- [\\_\\_IO uint32\\_t ICER](#) [1]
- uint32\_t [RESERVED1](#) [31]
- [\\_\\_IO uint32\\_t ISPR](#) [1]
- uint32\_t [RESERVED2](#) [31]
- [\\_\\_IO uint32\\_t ICPR](#) [1]
- uint32\_t [RESERVED3](#) [31]
- uint32\_t [RESERVED4](#) [64]
- [\\_\\_IO uint32\\_t IP](#) [8]
- [\\_\\_IO uint32\\_t IABR](#) [8]
- [\\_\\_IO uint8\\_t IP](#) [240]
- uint32\_t [RESERVED5](#) [644]
- [\\_\\_O uint32\\_t STIR](#)

## 7.45.2 Field Documentation

### 7.45.2.1 [\\_\\_IO uint32\\_t IABR](#)[8]

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

Definition at line 346 of file core\_cm4.h.

### 7.45.2.2 [\\_\\_IO uint32\\_t ICER](#)

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

Definition at line 295 of file core\_cm0plus.h.

### 7.45.2.3 [\\_\\_IO uint32\\_t ICPR](#)

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

Definition at line 299 of file core\_cm0plus.h.

### 7.45.2.4 [\\_\\_IO uint32\\_t IP](#)[8]

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

Definition at line 302 of file core\_cm0plus.h.

### 7.45.2.5 [\\_\\_IO uint8\\_t IP](#)[240]

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

Definition at line 348 of file core\_cm4.h.

### 7.45.2.6 [\\_\\_IO uint32\\_t ISER](#)

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

Definition at line 293 of file core\_cm0plus.h.



**7.45.2.7** `__IO uint32_t ISPR`

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

Definition at line 297 of file `core_cm0plus.h`.

**7.45.2.8** `uint32_t RESERVED0`

Definition at line 294 of file `core_cm0plus.h`.

**7.45.2.9** `uint32_t RESERVED2`

Definition at line 298 of file `core_cm0plus.h`.

**7.45.2.10** `uint32_t RESERVED3`

Definition at line 300 of file `core_cm0plus.h`.

**7.45.2.11** `uint32_t RESERVED4`

Definition at line 301 of file `core_cm0plus.h`.

**7.45.2.12** `uint32_t RESERVED5[644]`

Definition at line 349 of file `core_cm4.h`.

**7.45.2.13** `uint32_t RESERVED1`

Definition at line 296 of file `core_cm0plus.h`.

**7.45.2.14** `__O uint32_t STIR`

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

Definition at line 350 of file `core_cm4.h`.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm0plus.h](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm4.h](#)

## 7.46 PINMUX\_GRP\_T Struct Reference

### 7.46.1 Detailed Description

Array of IOCON pin definitions passed to [Chip\\_IOCON\\_SetPinMuxing\(\)](#) must be in this format.

Definition at line 54 of file `iocon_5411x.h`.

```
#include "iocon_5411x.h"
```

## Data Fields

- uint32\_t [port](#): 8
- uint32\_t [pin](#): 8
- uint32\_t [modefunc](#): 16

## 7.46.2 Field Documentation

### 7.46.2.1 uint32\_t modefunc

Definition at line 57 of file iocon\_5411x.h.

### 7.46.2.2 uint32\_t pin

Definition at line 56 of file iocon\_5411x.h.

### 7.46.2.3 uint32\_t port

Definition at line 55 of file iocon\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[iocon\\_5411x.h](#)

## 7.47 PINTABLE\_T Struct Reference

### 7.47.1 Detailed Description

LPC5411X Pin table structure used for enhanced boot block support.

Definition at line 84 of file pintable\_5411x.h.

```
#include "pintable_5411x.h"
```

## Data Fields

- uint32\_t [marker](#)
- uint8\_t [img\\_type](#)
- uint8\_t [ifSel](#)
- uint8\_t [hostIrqPortPin](#)
- uint8\_t [hostMisoPortPin](#)
- uint8\_t [hostMosiPortPin](#)
- uint8\_t [hostSselPortPin](#)
- uint8\_t [hostSckPortPin](#)
- uint8\_t [xorVal](#)
- uint32\_t [crc32\\_len](#)
- uint32\_t [crc32\\_val](#)
- uint32\_t [version](#)

## 7.47.2 Field Documentation

### 7.47.2.1 uint32\_t crc32\_len

Definition at line 110 of file pintable\_5411x.h.

#### 7.47.2.2 uint32\_t crc32\_val

Definition at line 111 of file `pintable_5411x.h`.

#### 7.47.2.3 uint8\_t hostIrqPortPin

Definition at line 98 of file `pintable_5411x.h`.

#### 7.47.2.4 uint8\_t hostMisoPortPin

Definition at line 100 of file `pintable_5411x.h`.

#### 7.47.2.5 uint8\_t hostMosiPortPin

Definition at line 102 of file `pintable_5411x.h`.

#### 7.47.2.6 uint8\_t hostSckPortPin

Definition at line 106 of file `pintable_5411x.h`.

#### 7.47.2.7 uint8\_t hostSselPortPin

Definition at line 104 of file `pintable_5411x.h`.

#### 7.47.2.8 uint8\_t ifSel

Definition at line 96 of file `pintable_5411x.h`.

#### 7.47.2.9 uint8\_t img\_type

Definition at line 93 of file `pintable_5411x.h`.

#### 7.47.2.10 uint32\_t marker

Definition at line 86 of file `pintable_5411x.h`.

#### 7.47.2.11 uint32\_t version

Definition at line 113 of file `pintable_5411x.h`.

#### 7.47.2.12 uint8\_t xorVal

Definition at line 108 of file `pintable_5411x.h`.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[pintable\\_5411x.h](#)

## 7.48 PLL\_CONFIG\_T Struct Reference

### 7.48.1 Detailed Description

PLL configuration structure This structure can be used to configure the settings for a PLL setup structure. Fill in the desired configuration for the PLL and call the PLL setup function to fill in a PLL setup structure.

Definition at line 226 of file pll\_5411x.h.

```
#include "pll_5411x.h"
```

### Data Fields

- uint32\_t [desiredRate](#)
- uint32\_t [InputRate](#)
- uint32\_t [flags](#)
- [SS\\_PROGMODFM\\_T](#) [ss\\_mf](#)
- [SS\\_PROGMODDP\\_T](#) [ss\\_mr](#)
- [SS\\_MODWVCTRL\\_T](#) [ss\\_mc](#)
- bool [mfDither](#)

### 7.48.2 Field Documentation

#### 7.48.2.1 uint32\_t desiredRate

Desired PLL rate in Hz

Definition at line 227 of file pll\_5411x.h.

#### 7.48.2.2 uint32\_t flags

PLL configuration flags, Or'ed value of PLL\_CONFIGFLAG\_\* definitions

Definition at line 229 of file pll\_5411x.h.

#### 7.48.2.3 uint32\_t InputRate

PLL input clock in Hz, only used if PLL\_CONFIGFLAG\_USEINRATE flag is set

Definition at line 228 of file pll\_5411x.h.

#### 7.48.2.4 bool mfDither

false for fixed modulation frequency or true for dithering, only applicable when not using PLL\_CONFIGFLAG\_FORCECENOFRACT flag

Definition at line 233 of file pll\_5411x.h.

#### 7.48.2.5 SS\_MODWVCTRL\_T ss\_mc

SS Modulation waveform control, only applicable when not using PLL\_CONFIGFLAG\_FORCECENOFRACT flag

Definition at line 232 of file pll\_5411x.h.

## 7.48.2.6 SS\_PROGMODFM\_T ss\_mf

SS Programmable modulation frequency, only applicable when not using PLL\_CONFIGFLAG\_FORCENOFRACT flag

Definition at line 230 of file pll\_5411x.h.

## 7.48.2.7 SS\_PROGMODDP\_T ss\_mr

SS Programmable frequency modulation depth, only applicable when not using PLL\_CONFIGFLAG\_FORCENOFRACT flag

Definition at line 231 of file pll\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/pll\_5411x.h

## 7.49 PLL\_SETUP\_T Struct Reference

### 7.49.1 Detailed Description

PLL setup structure This structure can be used to pre-build a PLL setup configuration at run-time and quickly set the PLL to the configuration. It can be populated with the PLL setup function. If powering up or waiting for PLL lock, the PLL input clock source should be configured prior to PLL setup.

Definition at line 250 of file pll\_5411x.h.

```
#include "pll_5411x.h"
```

### Data Fields

- uint32\_t [SYSPLLCTRL](#)
- uint32\_t [SYSPLLNDEC](#)
- uint32\_t [SYSPLLPDEC](#)
- uint32\_t [SYSPLLSSCTRL](#) [2]
- uint32\_t [pllRate](#)
- uint32\_t [flags](#)

### 7.49.2 Field Documentation

## 7.49.2.1 uint32\_t flags

PLL setup flags, Or'ed value of PLL\_SETUPFLAG\_\* definitions

Definition at line 256 of file pll\_5411x.h.

## 7.49.2.2 uint32\_t pllRate

Actual PLL rate

Definition at line 255 of file pll\_5411x.h.

#### 7.49.2.3 uint32\_t SYSPLLCTRL

PLL control register

Definition at line 251 of file pll\_5411x.h.

#### 7.49.2.4 uint32\_t SYSPLLNDEC

PLL NDEC register

Definition at line 252 of file pll\_5411x.h.

#### 7.49.2.5 uint32\_t SYSPLLPDEC

PLL PDEC register

Definition at line 253 of file pll\_5411x.h.

#### 7.49.2.6 uint32\_t SYSPLLSSCTRL[2]

PLL SSCTL registers

Definition at line 254 of file pll\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[pll\\_5411x.h](#)

## 7.50 REQUEST\_TYPE Union Reference

### 7.50.1 Detailed Description

Union of [\\_BM\\_T](#) struct and 8 bit byte.

Definition at line 141 of file usbd.h.

```
#include "usbd.h"
```

### Data Fields

- uint8\_t [B](#)
- BM\_T [BM](#)

### 7.50.2 Field Documentation

#### 7.50.2.1 uint8\_t B

byte wide access memeber

Definition at line 143 of file usbd.h.

#### 7.50.2.2 BM\_T BM

bitfield structure access memeber

Definition at line 144 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.51 RINGBUFF\_T Struct Reference

### 7.51.1 Detailed Description

Ring buffer structure.

Definition at line 46 of file `ring_buffer.h`.

```
#include "ring_buffer.h"
```

#### Data Fields

- void \* [data](#)
- int [count](#)
- int [itemSz](#)
- uint32\_t [head](#)
- uint32\_t [tail](#)
- void [\\*\(\\* copy\)\(void \\*dst, const void \\*src, uint32\\_t len\)](#)

### 7.51.2 Field Documentation

#### 7.51.2.1 void\*(\* copy)(void \*dst, const void \*src, uint32\_t len)

Definition at line 52 of file `ring_buffer.h`.

#### 7.51.2.2 int count

Definition at line 48 of file `ring_buffer.h`.

#### 7.51.2.3 void\* data

Definition at line 47 of file `ring_buffer.h`.

#### 7.51.2.4 uint32\_t head

Definition at line 50 of file `ring_buffer.h`.

#### 7.51.2.5 int itemSz

Definition at line 49 of file `ring_buffer.h`.

#### 7.51.2.6 uint32\_t tail

Definition at line 51 of file `ring_buffer.h`.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[ring\\_buffer.h](#)

## 7.52 SCB\_Type Struct Reference

### 7.52.1 Detailed Description

Structure type to access the System Control Block (SCB).

Definition at line 316 of file core\_cm0plus.h.

```
#include "core_cm0plus.h"
```

#### Data Fields

- [\\_\\_I uint32\\_t CPUID](#)
- [\\_\\_IO uint32\\_t ICSR](#)
- [uint32\\_t RESERVED0](#)
- [\\_\\_IO uint32\\_t AIRCR](#)
- [\\_\\_IO uint32\\_t SCR](#)
- [\\_\\_IO uint32\\_t CCR](#)
- [uint32\\_t RESERVED1](#)
- [\\_\\_IO uint32\\_t SHP \[2\]](#)
- [\\_\\_IO uint32\\_t SHCSR](#)
- [\\_\\_IO uint32\\_t VTOR](#)
- [\\_\\_IO uint8\\_t SHP \[12\]](#)
- [\\_\\_IO uint32\\_t CFSR](#)
- [\\_\\_IO uint32\\_t HFSR](#)
- [\\_\\_IO uint32\\_t DFSR](#)
- [\\_\\_IO uint32\\_t MMFAR](#)
- [\\_\\_IO uint32\\_t BFAR](#)
- [\\_\\_IO uint32\\_t AFSR](#)
- [\\_\\_I uint32\\_t PFR \[2\]](#)
- [\\_\\_I uint32\\_t DFR](#)
- [\\_\\_I uint32\\_t ADR](#)
- [\\_\\_I uint32\\_t MMFR \[4\]](#)
- [\\_\\_I uint32\\_t ISAR \[5\]](#)
- [\\_\\_IO uint32\\_t CPACR](#)

### 7.52.2 Field Documentation

#### 7.52.2.1 [\\_\\_I uint32\\_t ADR](#)

Offset: 0x04C (R/ ) Auxiliary Feature Register

Definition at line 386 of file core\_cm4.h.

#### 7.52.2.2 [\\_\\_IO uint32\\_t AFSR](#)

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

Definition at line 383 of file core\_cm4.h.

#### 7.52.2.3 [\\_\\_IO uint32\\_t AIRCR](#)

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

Definition at line 325 of file core\_cm0plus.h.



#### 7.52.2.4 `__IO uint32_t BFAR`

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

Definition at line 382 of file core\_cm4.h.

#### 7.52.2.5 `__IO uint32_t CCR`

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

Definition at line 327 of file core\_cm0plus.h.

#### 7.52.2.6 `__IO uint32_t CFSR`

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

Definition at line 378 of file core\_cm4.h.

#### 7.52.2.7 `__IO uint32_t CPACR`

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

Definition at line 390 of file core\_cm4.h.

#### 7.52.2.8 `__I uint32_t CPUID`

Offset: 0x000 (R/ ) CPUID Base Register

Definition at line 318 of file core\_cm0plus.h.

#### 7.52.2.9 `__I uint32_t DFR`

Offset: 0x048 (R/ ) Debug Feature Register

Definition at line 385 of file core\_cm4.h.

#### 7.52.2.10 `__IO uint32_t DFSR`

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

Definition at line 380 of file core\_cm4.h.

#### 7.52.2.11 `__IO uint32_t HFSR`

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

Definition at line 379 of file core\_cm4.h.

#### 7.52.2.12 `__IO uint32_t ICSR`

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

Definition at line 319 of file core\_cm0plus.h.

#### 7.52.2.13 `__I uint32_t ISAR[5]`

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

Definition at line 388 of file core\_cm4.h.

#### 7.52.2.14 `__IO uint32_t MMFAR`

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

Definition at line 381 of file core\_cm4.h.

#### 7.52.2.15 `__I uint32_t MMFR[4]`

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

Definition at line 387 of file core\_cm4.h.

#### 7.52.2.16 `__I uint32_t PFR[2]`

Offset: 0x040 (R/ ) Processor Feature Register

Definition at line 384 of file core\_cm4.h.

#### 7.52.2.17 `uint32_t RESERVED0`

Definition at line 323 of file core\_cm0plus.h.

#### 7.52.2.18 `uint32_t RESERVED1`

Definition at line 328 of file core\_cm0plus.h.

#### 7.52.2.19 `__IO uint32_t SCR`

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

Definition at line 326 of file core\_cm0plus.h.

#### 7.52.2.20 `__IO uint32_t SHCSR`

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

Definition at line 330 of file core\_cm0plus.h.

#### 7.52.2.21 `__IO uint32_t SHP[2]`

Offset: 0x01C (R/W) System Handlers Priority Registers. [0] is RESERVED

Definition at line 329 of file core\_cm0plus.h.

#### 7.52.2.22 `__IO uint8_t SHP[12]`

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

Definition at line 376 of file core\_cm4.h.

7.52.2.23 `__IO uint32_t VTOR`

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

Definition at line 372 of file `core_cm4.h`.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm0plus.h](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm4.h](#)

## 7.53 SCnSCB\_Type Struct Reference

### 7.53.1 Detailed Description

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

Definition at line 585 of file `core_cm4.h`.

```
#include "core_cm4.h"
```

### Data Fields

- `uint32_t` [RESERVED0](#) [1]
- `__I uint32_t` [ICTR](#)
- `__IO uint32_t` [ACTLR](#)

### 7.53.2 Field Documentation

7.53.2.1 `__IO uint32_t ACTLR`

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

Definition at line 589 of file `core_cm4.h`.

7.53.2.2 `__I uint32_t ICTR`

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

Definition at line 588 of file `core_cm4.h`.

7.53.2.3 `uint32_t RESERVED0[1]`

Definition at line 587 of file `core_cm4.h`.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm4.h](#)

## 7.54 SPI\_CFGSETUP\_T Struct Reference

### 7.54.1 Detailed Description

SPI configuration structure used for setting up master/slave mode, LSB or MSB first, and SPI mode in a single function call.

Definition at line 378 of file spi\_common\_5411x.h.

```
#include "spi_common_5411x.h"
```

### Data Fields

- uint32\_t [master](#): 8
- uint32\_t [lsbFirst](#): 8
- [SPI\\_CLOCK\\_MODE\\_T](#) [mode](#): 8
- uint32\_t [reserved](#): 8

## 7.54.2 Field Documentation

### 7.54.2.1 uint32\_t lsbFirst

Definition at line 380 of file spi\_common\_5411x.h.

### 7.54.2.2 uint32\_t master

Definition at line 379 of file spi\_common\_5411x.h.

### 7.54.2.3 SPI\_CLOCK\_MODE\_T mode

Definition at line 381 of file spi\_common\_5411x.h.

### 7.54.2.4 uint32\_t reserved

Definition at line 382 of file spi\_common\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[spi\\_common\\_5411x.h](#)

## 7.55 SPIM\_DELAY\_CONFIG\_T Struct Reference

### 7.55.1 Detailed Description

SPI Delay Configure Struct.

#### Note

All values are in SPI clocks.

Definition at line 70 of file spim\_5411x.h.

```
#include "spim_5411x.h"
```

### Data Fields

- uint8\_t [PreDelay](#)
- uint8\_t [PostDelay](#)
- uint8\_t [FrameDelay](#)
- uint8\_t [TransferDelay](#)

## 7.55.2 Field Documentation

### 7.55.2.1 uint8\_t FrameDelay

Additional Post-delay, minimum is 1 clock (SSEL asserted), 0 - 15

Definition at line 73 of file `spim_5411x.h`.

### 7.55.2.2 uint8\_t PostDelay

Additional Pre-delay, minimum is 1 clock (SSEL asserted), 0 - 15

Definition at line 72 of file `spim_5411x.h`.

### 7.55.2.3 uint8\_t PreDelay

Definition at line 71 of file `spim_5411x.h`.

### 7.55.2.4 uint8\_t TransferDelay

Delay between frames (SSEL asserted), 0 - 15

Definition at line 74 of file `spim_5411x.h`.

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

- [C:/Jenkins/LPC5411x/lpc5411x/chip\\_5411x/inc/spim\\_5411x.h](#)

## 7.56 SPIM\_XFER\_T Struct Reference

### 7.56.1 Detailed Description

SPI Master transfer data context.

#### Note

When structure member *dataWidth* > 8; *txData* and *rxData* must be of type (uint16\_t \*) otherwise it should be of type (uint8\_t \*)

Member *options* must be OR'd values from SPI\_XFER\_OPT\_\* (see #SPI\_XFER\_OPT\_EOF)

Definition at line 156 of file `spim_5411x.h`.

```
#include "spim_5411x.h"
```

#### Data Fields

- int(\* [eventCB](#) )(LPC\_SPI\_T \*pSPI, [SPIM\\_EVENT\\_T](#) evt, struct SPIM\_XFER \*xfer)
- void \* [txData](#)
- void \* [rxData](#)
- uint16\_t [txCount](#)
- uint16\_t [rxCount](#)
- uint16\_t [dataWidth](#)
- uint16\_t [sselNum](#)
- uint32\_t [option](#)
- void \* [usrData](#)

- [uint16\\_t txIndex](#)
- [uint16\\_t rxIndex](#)
- [SPIM\\_XFER\\_STATE\\_T state](#)

## 7.56.2 Field Documentation

### 7.56.2.1 `uint16_t dataWidth`

Width of the data [Valid values: 1 to 16]

Definition at line 163 of file `spim_5411x.h`.

### 7.56.2.2 `int(* eventCB)(LPC_SPI_T *pSPI, SPIM_EVENT_T evt, struct SPIM_XFER *xfer)`

Definition at line 158 of file `spim_5411x.h`.

### 7.56.2.3 `uint32_t option`

Dummy data to be transfered when `rxCount > txCount` ORed with `#SPI_XFER_OPT_EOF` (if this functionality is required)

Definition at line 165 of file `spim_5411x.h`.

### 7.56.2.4 `uint16_t rxCount`

Number of data to be received [Not in bytes]

Definition at line 162 of file `spim_5411x.h`.

### 7.56.2.5 `void* rxData`

Pointer to buffer where received data be stored

Definition at line 160 of file `spim_5411x.h`.

### 7.56.2.6 `uint16_t rxIndex`

Index of memory to which next received data be stored

Definition at line 169 of file `spim_5411x.h`.

### 7.56.2.7 `uint16_t sselNum`

Slave select number to be asserted when transferring data [Valid values: 0 to 3]

Definition at line 164 of file `spim_5411x.h`.

### 7.56.2.8 `SPIM_XFER_STATE_T state`

State of the transfer

Definition at line 170 of file `spim_5411x.h`.

**7.56.2.9 uint16\_t txCount**

Number of data to be transmitted [Not in bytes]

Definition at line 161 of file `spim_5411x.h`.

**7.56.2.10 void\* txData**

Pointer to SPI master transfer event callback functions Pointer to buffer having transmit data

Definition at line 159 of file `spim_5411x.h`.

**7.56.2.11 uint16\_t txIndex**

Index of the next item to be transfered [If this is same as txCount then tx is complete]

Definition at line 168 of file `spim_5411x.h`.

**7.56.2.12 void\* usrData**

User data associated with this Xfer structure

Definition at line 166 of file `spim_5411x.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/spim_5411x.h`

**7.57 SPIS\_XFER\_T Struct Reference****7.57.1 Detailed Description**

Slave transfer data context

Definition at line 58 of file `spis_5411x.h`.

```
#include "spis_5411x.h"
```

**Data Fields**

- `int(* eventCB)(LPC_SPI_T *pSPI, SPIS_EVENT_T evt, struct SPIS_XFER *xfer)`
- `void * txData`
- `void * rxData`
- `uint16_t rxCount`
- `uint16_t txCount`
- `uint16_t txIndex`
- `uint16_t rxIndex`
- `uint16_t thresCount`
- `uint16_t dataWidth`
- `uint8_t sselNum`
- `uint32_t ss_count`
- `bool ss_state`

## 7.57.2 Field Documentation

### 7.57.2.1 uint16\_t dataWidth

Width of the data [Valid values: 1 to 16]

Definition at line 67 of file spis\_5411x.h.

### 7.57.2.2 int(\* eventCB)(LPC\_SPI\_T \*pSPI, SPIS\_EVENT\_T evt, struct SPIS\_XFER \*xfer)

Pointer to SPI slave callback functions

Definition at line 59 of file spis\_5411x.h.

### 7.57.2.3 uint16\_t rxCount

Size of the pRXData buffer in items (not bytes), modified by driver

Definition at line 62 of file spis\_5411x.h.

### 7.57.2.4 void\* rxData

RX Data pointer

Definition at line 61 of file spis\_5411x.h.

### 7.57.2.5 uint16\_t rxIndex

Total items (not bytes) transmitted, modified by driver

Definition at line 65 of file spis\_5411x.h.

### 7.57.2.6 uint32\_t ss\_count

Count of ssel events from the data, modified by driver

Definition at line 69 of file spis\_5411x.h.

### 7.57.2.7 bool ss\_state

State of slave-select true = asserted, false = de-asserted

Definition at line 70 of file spis\_5411x.h.

### 7.57.2.8 uint8\_t sselNum

Slave number assigned to this transfer, 0 - 3, modified by driver

Definition at line 68 of file spis\_5411x.h.

### 7.57.2.9 uint16\_t thresCount

Number of data to receive to trigger [SPIS\\_EVENT\\_THRESHOLD](#) event

Definition at line 66 of file spis\_5411x.h.



#### 7.57.2.10 uint16\_t txCount

Number of items (not bytes) to send in pTXData buffer, modified by driver

Definition at line 63 of file spis\_5411x.h.

#### 7.57.2.11 void\* txData

TX Data pointer

Definition at line 60 of file spis\_5411x.h.

#### 7.57.2.12 uint16\_t txIndex

Total items (not bytes) received, modified by driver

Definition at line 64 of file spis\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[spis\\_5411x.h](#)

## 7.58 SysTick\_Type Struct Reference

### 7.58.1 Detailed Description

Structure type to access the System Timer (SysTick).

Definition at line 431 of file core\_cm0plus.h.

```
#include "core_cm0plus.h"
```

### Data Fields

- [\\_\\_IO uint32\\_t CTRL](#)
- [\\_\\_IO uint32\\_t LOAD](#)
- [\\_\\_IO uint32\\_t VAL](#)
- [\\_\\_I uint32\\_t CALIB](#)

### 7.58.2 Field Documentation

#### 7.58.2.1 \_\_I uint32\_t CALIB

Offset: 0x00C (R/ ) SysTick Calibration Register

Definition at line 436 of file core\_cm0plus.h.

#### 7.58.2.2 \_\_IO uint32\_t CTRL

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

Definition at line 433 of file core\_cm0plus.h.

### 7.58.2.3 `__IO uint32_t LOAD`

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

Definition at line 434 of file `core_cm0plus.h`.

### 7.58.2.4 `__IO uint32_t VAL`

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

Definition at line 435 of file `core_cm0plus.h`.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm0plus.h](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm4.h](#)

## 7.59 TPI\_Type Struct Reference

### 7.59.1 Detailed Description

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

Definition at line 919 of file `core_cm4.h`.

```
#include "core_cm4.h"
```

### Data Fields

- `__IO uint32_t SSPSR`
- `__IO uint32_t CSPSR`
- `uint32_t RESERVED0 [2]`
- `__IO uint32_t ACPR`
- `uint32_t RESERVED1 [55]`
- `__IO uint32_t SPPR`
- `uint32_t RESERVED2 [131]`
- `__I uint32_t FFSR`
- `__IO uint32_t FFCR`
- `__I uint32_t FSCR`
- `uint32_t RESERVED3 [759]`
- `__I uint32_t TRIGGER`
- `__I uint32_t FIFO0`
- `__I uint32_t ITATBCTR2`
- `uint32_t RESERVED4 [1]`
- `__I uint32_t ITATBCTR0`
- `__I uint32_t FIFO1`
- `__IO uint32_t ITCTRL`
- `uint32_t RESERVED5 [39]`
- `__IO uint32_t CLAIMSET`
- `__IO uint32_t CLAIMCLR`
- `uint32_t RESERVED7 [8]`
- `__I uint32_t DEVID`
- `__I uint32_t DEVTYPE`

## 7.59.2 Field Documentation

### 7.59.2.1 `__IO uint32_t ACPR`

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

Definition at line 924 of file core\_cm4.h.

### 7.59.2.2 `__IO uint32_t CLAIMCLR`

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

Definition at line 941 of file core\_cm4.h.

### 7.59.2.3 `__IO uint32_t CLAIMSET`

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

Definition at line 940 of file core\_cm4.h.

### 7.59.2.4 `__IO uint32_t CSPSR`

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

Definition at line 922 of file core\_cm4.h.

### 7.59.2.5 `__I uint32_t DEVID`

Offset: 0xFC8 (R/ ) TPIU\_DEVID

Definition at line 943 of file core\_cm4.h.

### 7.59.2.6 `__I uint32_t DEVTYPE`

Offset: 0xFCC (R/ ) TPIU\_DEVTYPE

Definition at line 944 of file core\_cm4.h.

### 7.59.2.7 `__IO uint32_t FFCR`

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

Definition at line 929 of file core\_cm4.h.

### 7.59.2.8 `__I uint32_t FFSR`

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

Definition at line 928 of file core\_cm4.h.

### 7.59.2.9 `__I uint32_t FIFO0`

Offset: 0xEEC (R/ ) Integration ETM Data

Definition at line 933 of file core\_cm4.h.

#### 7.59.2.10 `__I uint32_t FIFO1`

Offset: 0xEFC (R/ ) Integration ITM Data

Definition at line 937 of file core\_cm4.h.

#### 7.59.2.11 `__I uint32_t FSCR`

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

Definition at line 930 of file core\_cm4.h.

#### 7.59.2.12 `__I uint32_t ITATBCTR0`

Offset: 0xEF8 (R/ ) ITATBCTR0

Definition at line 936 of file core\_cm4.h.

#### 7.59.2.13 `__I uint32_t ITATBCTR2`

Offset: 0xEF0 (R/ ) ITATBCTR2

Definition at line 934 of file core\_cm4.h.

#### 7.59.2.14 `__IO uint32_t ITCTRL`

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

Definition at line 938 of file core\_cm4.h.

#### 7.59.2.15 `uint32_t RESERVED0[2]`

Definition at line 923 of file core\_cm4.h.

#### 7.59.2.16 `uint32_t RESERVED1[55]`

Definition at line 925 of file core\_cm4.h.

#### 7.59.2.17 `uint32_t RESERVED2[131]`

Definition at line 927 of file core\_cm4.h.

#### 7.59.2.18 `uint32_t RESERVED3[759]`

Definition at line 931 of file core\_cm4.h.

#### 7.59.2.19 `uint32_t RESERVED4[1]`

Definition at line 935 of file core\_cm4.h.

#### 7.59.2.20 `uint32_t RESERVED5[39]`

Definition at line 939 of file core\_cm4.h.

#### 7.59.2.21 uint32\_t RESERVED7[8]

Definition at line 942 of file core\_cm4.h.

#### 7.59.2.22 \_\_IO uint32\_t SPPR

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

Definition at line 926 of file core\_cm4.h.

#### 7.59.2.23 \_\_IO uint32\_t SSPSR

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

Definition at line 921 of file core\_cm4.h.

#### 7.59.2.24 \_\_I uint32\_t TRIGGER

Offset: 0xEE8 (R/ ) TRIGGER

Definition at line 932 of file core\_cm4.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm4.h](#)

## 7.60 UART\_BAUD\_T Struct Reference

### 7.60.1 Detailed Description

UART Baud rate calculation structure.

#### Note

Use oversampling (*ovr*) value other than 16, only if the difference between the actual baud and desired baud has an unacceptable error percentage. Smaller *ovr* values can cause the sampling position within the data-bit less accurate and may potentially cause more noise errors or incorrect data set *ovr* to < 10 only when there is no other higher values suitable.

Definition at line 211 of file uart\_5411x.h.

```
#include "uart_5411x.h"
```

#### Data Fields

- uint32\_t [clk](#)
- uint32\_t [baud](#)
- uint8\_t [ovr](#)
- uint8\_t [mul](#)
- uint16\_t [div](#)

## 7.60.2 Field Documentation

### 7.60.2.1 uint32\_t baud

IN: Required baud rate; OUT: Actual baud rate

Definition at line 213 of file uart\_5411x.h.

### 7.60.2.2 uint32\_t clk

IN: Base clock to fractional divider; OUT: "Base clock rate for UART"

Definition at line 212 of file uart\_5411x.h.

### 7.60.2.3 uint16\_t div

OUT: Integer divider to divide the "Base clock rate for UART"

Definition at line 216 of file uart\_5411x.h.

### 7.60.2.4 uint8\_t mul

IN: 0 - calculate MUL, 1 - do't calculate (*clk*) has UART base clock; OUT: MUL value to be set in FRG register

Definition at line 215 of file uart\_5411x.h.

### 7.60.2.5 uint8\_t ovr

IN: Number of desired over samples [0-auto detect or values 5 to 16]; OUT: Auto detected over samples [unchanged if IN is not 0]

Definition at line 214 of file uart\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[uart\\_5411x.h](#)

## 7.61 UART\_STATISTICS\_T Struct Reference

### 7.61.1 Detailed Description

UART statistics structure.

#### Note

Maintains current UART statistics.

Definition at line 224 of file uart\_5411x.h.

```
#include "uart_5411x.h"
```

### Data Fields

- uint32\_t [interrupts](#)
- uint32\_t [lvl\\_tx](#)
- uint32\_t [lvl\\_rx](#)

- [uint32\\_t fifo\\_err\\_tx](#)
- [uint32\\_t fifo\\_err\\_rx](#)
- [uint32\\_t uart\\_cts](#)
- [uint32\\_t uart\\_break](#)
- [uint32\\_t uart\\_start](#)
- [uint32\\_t uart\\_err\\_frame](#)
- [uint32\\_t uart\\_err\\_parity](#)
- [uint32\\_t uart\\_err\\_rx\\_noise](#)
- [uint32\\_t uart\\_err\\_auto\\_baud](#)

## 7.61.2 Field Documentation

### 7.61.2.1 `uint32_t fifo_err_rx`

count: FIFO receive errors

Definition at line 229 of file `uart_5411x.h`.

### 7.61.2.2 `uint32_t fifo_err_tx`

count: FIFO transmit errors

Definition at line 228 of file `uart_5411x.h`.

### 7.61.2.3 `uint32_t interrupts`

count: interrupts

Definition at line 225 of file `uart_5411x.h`.

### 7.61.2.4 `uint32_t lvl_rx`

count: receive interrupts

Definition at line 227 of file `uart_5411x.h`.

### 7.61.2.5 `uint32_t lvl_tx`

count: transmit interrupts

Definition at line 226 of file `uart_5411x.h`.

### 7.61.2.6 `uint32_t uart_break`

count: UART break

Definition at line 231 of file `uart_5411x.h`.

### 7.61.2.7 `uint32_t uart_cts`

count: UART CTS

Definition at line 230 of file `uart_5411x.h`.

#### 7.61.2.8 uint32\_t uart\_err\_auto\_baud

count: UART auto-baud errors

Definition at line 236 of file uart\_5411x.h.

#### 7.61.2.9 uint32\_t uart\_err\_frame

count: UART frame errors

Definition at line 233 of file uart\_5411x.h.

#### 7.61.2.10 uint32\_t uart\_err\_parity

count: UART parity errors

Definition at line 234 of file uart\_5411x.h.

#### 7.61.2.11 uint32\_t uart\_err\_rx\_noise

count: UART receive noise errors

Definition at line 235 of file uart\_5411x.h.

#### 7.61.2.12 uint32\_t uart\_start

count: UART RX starts

Definition at line 232 of file uart\_5411x.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[uart\\_5411x.h](#)

## 7.62 USB\_COMMON\_DESCRIPTOR Struct Reference

### 7.62.1 Detailed Description

USB Common Descriptor

Definition at line 664 of file usbd.h.

```
#include "usbd.h"
```

#### Data Fields

- uint8\_t [bLength](#)
- uint8\_t [bDescriptorType](#)

### 7.62.2 Field Documentation

#### 7.62.2.1 uint8\_t bDescriptorType

Descriptor Type

Definition at line 667 of file usbd.h.



#### 7.62.2.2 uint8\_t bLength

Size of this descriptor in bytes

Definition at line 666 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.63 USB\_CONFIGURATION\_DESCRIPTOR Struct Reference

### 7.63.1 Detailed Description

USB Standard Configuration Descriptor

Definition at line 437 of file usbd.h.

```
#include "usbd.h"
```

#### Data Fields

- [uint8\\_t bLength](#)
- [uint8\\_t bDescriptorType](#)
- [uint16\\_t wTotalLength](#)
- [uint8\\_t bNumInterfaces](#)
- [uint8\\_t bConfigurationValue](#)
- [uint8\\_t iConfiguration](#)
- [uint8\\_t bmAttributes](#)
- [uint8\\_t bMaxPower](#)

### 7.63.2 Field Documentation

#### 7.63.2.1 uint8\_t bConfigurationValue

Value to use as an argument to the SetConfiguration() request to select this configuration.

Definition at line 447 of file usbd.h.

#### 7.63.2.2 uint8\_t bDescriptorType

CONFIGURATION Descriptor Type

Definition at line 440 of file usbd.h.

#### 7.63.2.3 uint8\_t bLength

Size of this descriptor in bytes

Definition at line 439 of file usbd.h.

#### 7.63.2.4 uint8\_t bmAttributes

Configuration characteristics

D7: Reserved (set to one)

D6: Self-powered

D5: Remote Wakeup

D4...0: Reserved (reset to zero)

D7 is reserved and must be set to one for historical reasons.

A device configuration that uses power from the bus and a local source reports a non-zero value in bMaxPower to indicate the amount of bus power required and sets D6. The actual power source at runtime may be determined using the GetStatus(DEVICE) request (see USB 2.0 spec Section 9.4.5).

If a device configuration supports remote wakeup, D5 is set to one.

Definition at line 452 of file usbd.h.

#### 7.63.2.5 uint8\_t bMaxPower

Maximum power consumption of the USB device from the bus in this specific configuration when the device is fully operational. Expressed in 2 mA units (i.e., 50 = 100 mA).

Note: A device configuration reports whether the configuration is bus-powered or selfpowered. Device status reports whether the device is currently self-powered. If a device is disconnected from its external power source, it updates device status to indicate that it is no longer self-powered.

A device may not increase its power draw from the bus, when it loses its external power source, beyond the amount reported by its configuration.

If a device can continue to operate when disconnected from its external power source, it continues to do so. If the device cannot continue to operate, it fails operations it can no longer support. The USB System Software may determine the cause of the failure by checking the status and noting the loss of the devices power source.

Definition at line 468 of file usbd.h.

#### 7.63.2.6 uint8\_t bNumInterfaces

Number of interfaces supported by this configuration

Definition at line 446 of file usbd.h.

#### 7.63.2.7 uint8\_t iConfiguration

Index of string descriptor describing this configuration

Definition at line 450 of file usbd.h.

#### 7.63.2.8 uint16\_t wTotalLength

Total length of data returned for this configuration. Includes the combined length of all descriptors (configuration, interface, endpoint, and class- or vendor-specific) returned for this configuration.

Definition at line 441 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.64 USB\_DEVICE\_DESCRIPTOR Struct Reference

### 7.64.1 Detailed Description

USB Standard Device Descriptor

Definition at line 352 of file usbd.h.

```
#include "usbd.h"
```

## Data Fields

- uint8\_t bLength
- uint8\_t bDescriptorType
- uint16\_t bcdUSB
- uint8\_t bDeviceClass
- uint8\_t bDeviceSubClass
- uint8\_t bDeviceProtocol
- uint8\_t bMaxPacketSize0
- uint16\_t idVendor
- uint16\_t idProduct
- uint16\_t bcdDevice
- uint8\_t iManufacturer
- uint8\_t iProduct
- uint8\_t iSerialNumber
- uint8\_t bNumConfigurations

### 7.64.2 Field Documentation

#### 7.64.2.1 uint16\_t bcdDevice

Device release number in binary-coded decimal.

Definition at line 409 of file usbd.h.

#### 7.64.2.2 uint16\_t bcdUSB

BUSB Specification Release Number in Binary-Coded Decimal (i.e., 2.10 is 210H). This field identifies the release of the USB Specification with which the device and its descriptors are compliant.

Definition at line 356 of file usbd.h.

#### 7.64.2.3 uint8\_t bDescriptorType

DEVICE Descriptor Type.

Definition at line 355 of file usbd.h.

#### 7.64.2.4 uint8\_t bDeviceClass

Class code (assigned by the USB-IF). If this field is reset to zero, each interface within a configuration specifies its own class information and the various interfaces operate independently.

If this field is set to a value between 1 and FEH, the device supports different class specifications on different interfaces and the interfaces may not operate independently. This value identifies the class definition used for the aggregate interfaces.

If this field is set to FFH, the device class is vendor-specific.

Definition at line 362 of file usbd.h.

#### 7.64.2.5 uint8\_t bDeviceProtocol

Protocol code (assigned by the USB-IF). These codes are qualified by the value of the bDeviceClass and the bDeviceSubClass fields. If a device supports class-specific protocols on a device basis as opposed to an interface basis, this code identifies the protocols that the device uses as defined by the specification of the device class.

If this field is reset to zero, the device does not use class-specific protocols on a device basis. However, it may use

classspecific protocols on an interface basis.

If this field is set to FFH, the device uses a vendor-specific protocol on a device basis.

Definition at line 386 of file usbd.h.

#### 7.64.2.6 `uint8_t bDeviceSubClass`

Subclass code (assigned by the USB-IF). These codes are qualified by the value of the `bDeviceClass` field.

If the `bDeviceClass` field is reset to zero, this field must also be reset to zero.

If the `bDeviceClass` field is not set to FFH, all values are reserved for assignment by the USB-IF.

Definition at line 377 of file usbd.h.

#### 7.64.2.7 `uint8_t bLength`

Size of this descriptor in bytes.

Definition at line 354 of file usbd.h.

#### 7.64.2.8 `uint8_t bMaxPacketSize0`

Maximum packet size for endpoint zero (only 8, 16, 32, or 64 are valid). For HS devices is fixed to 64.

Definition at line 402 of file usbd.h.

#### 7.64.2.9 `uint8_t bNumConfigurations`

Number of possible configurations.

Definition at line 415 of file usbd.h.

#### 7.64.2.10 `uint16_t idProduct`

Product ID (assigned by the manufacturer).

Definition at line 408 of file usbd.h.

#### 7.64.2.11 `uint16_t idVendor`

Vendor ID (assigned by the USB-IF).

Definition at line 407 of file usbd.h.

#### 7.64.2.12 `uint8_t iManufacturer`

Index of string descriptor describing manufacturer.

Definition at line 410 of file usbd.h.

#### 7.64.2.13 `uint8_t iProduct`

Index of string descriptor describing product.

Definition at line 411 of file usbd.h.

#### 7.64.2.14 uint8\_t iSerialNumber

Index of string descriptor describing the devices serial number.

Definition at line 412 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.65 USB\_DEVICE\_QUALIFIER\_DESCRIPTOR Struct Reference

### 7.65.1 Detailed Description

USB 2.0 Device Qualifier Descriptor

Definition at line 421 of file usbd.h.

```
#include "usbd.h"
```

#### Data Fields

- uint8\_t [bLength](#)
- uint8\_t [bDescriptorType](#)
- uint16\_t [bcdUSB](#)
- uint8\_t [bDeviceClass](#)
- uint8\_t [bDeviceSubClass](#)
- uint8\_t [bDeviceProtocol](#)
- uint8\_t [bMaxPacketSize0](#)
- uint8\_t [bNumConfigurations](#)
- uint8\_t [bReserved](#)

### 7.65.2 Field Documentation

#### 7.65.2.1 uint16\_t bcdUSB

USB specification version number (e.g., 0200H for V2.00)

Definition at line 425 of file usbd.h.

#### 7.65.2.2 uint8\_t bDescriptorType

Device Qualifier Type

Definition at line 424 of file usbd.h.

#### 7.65.2.3 uint8\_t bDeviceClass

Class Code

Definition at line 426 of file usbd.h.

#### 7.65.2.4 uint8\_t bDeviceProtocol

Protocol Code

Definition at line 428 of file usbd.h.

#### 7.65.2.5 uint8\_t bDeviceSubClass

SubClass Code

Definition at line 427 of file usbd.h.

#### 7.65.2.6 uint8\_t bLength

Size of descriptor

Definition at line 423 of file usbd.h.

#### 7.65.2.7 uint8\_t bMaxPacketSize0

Maximum packet size for other speed

Definition at line 429 of file usbd.h.

#### 7.65.2.8 uint8\_t bNumConfigurations

Number of Other-speed Configurations

Definition at line 430 of file usbd.h.

#### 7.65.2.9 uint8\_t bReserved

Reserved for future use, must be zero

Definition at line 431 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.66 USB\_ENDPOINT\_DESCRIPTOR Struct Reference

### 7.66.1 Detailed Description

USB Standard Endpoint Descriptor

Definition at line 571 of file usbd.h.

```
#include "usbd.h"
```

#### Data Fields

- uint8\_t [bLength](#)
- uint8\_t [bDescriptorType](#)
- uint8\_t [bEndpointAddress](#)
- uint8\_t [bmAttributes](#)
- uint16\_t [wMaxPacketSize](#)
- uint8\_t [bInterval](#)

## 7.66.2 Field Documentation

### 7.66.2.1 uint8\_t bDescriptorType

ENDPOINT Descriptor Type

Definition at line 574 of file usbd.h.

### 7.66.2.2 uint8\_t bEndpointAddress

The address of the endpoint on the USB device described by this descriptor. The address is encoded as follows:

Bit 3...0: The endpoint number

Bit 6...4: Reserved, reset to zero

Bit 7: Direction, ignored for control endpoints 0 = OUT endpoint 1 = IN endpoint.

See also

USB\_ENDPOINT\_ADR\_Type

Definition at line 575 of file usbd.h.

### 7.66.2.3 uint8\_t bInterval

Interval for polling endpoint for data transfers. Expressed in frames or microframes depending on the device operating speed (i.e., either 1 millisecond or 125 s units).

- For full-/high-speed isochronous endpoints, this value must be in the range from 1 to 16. The bInterval value is used as the exponent for a  $2^{(bInterval - 1)}$  value; e.g., a bInterval of 4 means a period of 8 ( $2^{(4 - 1)}$ ).
- For full-/low-speed interrupt endpoints, the value of this field may be from 1 to 255.
- For high-speed interrupt endpoints, the bInterval value is used as the exponent for a  $2^{(bInterval - 1)}$  value; e.g., a bInterval of 4 means a period of 8 ( $2^{(4 - 1)}$ ). This value must be from 1 to 16.
- For high-speed bulk/control OUT endpoints, the bInterval must specify the maximum NAK rate of the endpoint. A value of 0 indicates the endpoint never NAKs. Other values indicate at most 1 NAK each bInterval number of microframes. This value must be in the range from 0 to 255.  
Refer to Chapter 5 of USB 2.0 specification for more information.

Definition at line 627 of file usbd.h.

### 7.66.2.4 uint8\_t bLength

Size of this descriptor in bytes

Definition at line 573 of file usbd.h.

### 7.66.2.5 uint8\_t bmAttributes

This field describes the endpoints attributes when it is configured using the bConfigurationValue.

Bits 1..0: Transfer Type

- 00 = Control
- 01 = Isochronous
- 10 = Bulk

- 11 = Interrupt  
If not an isochronous endpoint, bits 5..2 are reserved and must be set to zero. If isochronous, they are defined as follows:  
Bits 3..2: Synchronization Type
- 00 = No Synchronization
- 01 = Asynchronous
- 10 = Adaptive
- 11 = Synchronous  
Bits 5..4: Usage Type
- 00 = Data endpoint
- 01 = Feedback endpoint
- 10 = Implicit feedback Data endpoint
- 11 = Reserved  
Refer to Chapter 5 of USB 2.0 specification for more information.  
All other bits are reserved and must be reset to zero. Reserved bits must be ignored by the host.

See also

USBD\_EP\_ATTR\_Type

Definition at line 583 of file usbd.h.

#### 7.66.2.6 uint16\_t wMaxPacketSize

Maximum packet size this endpoint is capable of sending or receiving when this configuration is selected. For isochronous endpoints, this value is used to reserve the bus time in the schedule, required for the per-(micro)frame data payloads. The pipe may, on an ongoing basis, actually use less bandwidth than that reserved. The device reports, if necessary, the actual bandwidth used via its normal, non-USB defined mechanisms. For all endpoints, bits 10..0 specify the maximum packet size (in bytes). For high-speed isochronous and interrupt endpoints: Bits 12..11 specify the number of additional transaction opportunities per microframe:

- 00 = None (1 transaction per microframe)
- 01 = 1 additional (2 per microframe)
- 10 = 2 additional (3 per microframe)
- 11 = Reserved  
Bits 15..13 are reserved and must be set to zero.

Definition at line 607 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)



## 7.67 USB\_IAD\_DESCRIPTOR Struct Reference

### 7.67.1 Detailed Description

USB Standard Interface Association Descriptor

Definition at line 497 of file usbd.h.

```
#include "usbd.h"
```

#### Data Fields

- [uint8\\_t bLength](#)
- [uint8\\_t bDescriptorType](#)
- [uint8\\_t bFirstInterface](#)
- [uint8\\_t bInterfaceCount](#)
- [uint8\\_t bFunctionClass](#)
- [uint8\\_t bFunctionSubClass](#)
- [uint8\\_t bFunctionProtocol](#)
- [uint8\\_t iFunction](#)

### 7.67.2 Field Documentation

#### 7.67.2.1 [uint8\\_t bDescriptorType](#)

INTERFACE ASSOCIATION Descriptor Type

Definition at line 500 of file usbd.h.

#### 7.67.2.2 [uint8\\_t bFirstInterface](#)

Interface number of the first interface that is associated with this function.

Definition at line 501 of file usbd.h.

#### 7.67.2.3 [uint8\\_t bFunctionClass](#)

Class code (assigned by USB-IF).

A value of zero is not allowed in this descriptor. If this field is FFH, the function class is vendorspecific. All other values are reserved for assignment by the USB-IF.

Definition at line 505 of file usbd.h.

#### 7.67.2.4 [uint8\\_t bFunctionProtocol](#)

Protocol code (assigned by the USB).

These codes are qualified by the values of the bFunctionClass and bFunctionSubClass fields.

Definition at line 513 of file usbd.h.

#### 7.67.2.5 [uint8\\_t bFunctionSubClass](#)

Subclass code (assigned by USB-IF).

If the bFunctionClass field is not set to FFH all values are reserved for assignment by the USBIF.

Definition at line 510 of file usbd.h.

#### 7.67.2.6 `uint8_t bInterfaceCount`

Number of contiguous interfaces that are associated with this function.

Definition at line 503 of file `usbd.h`.

#### 7.67.2.7 `uint8_t bLength`

Size of this descriptor in bytes

Definition at line 499 of file `usbd.h`.

#### 7.67.2.8 `uint8_t iFunction`

Index of string descriptor describing this function.

Definition at line 516 of file `usbd.h`.

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

- `C:/Jenkins/LPC5411x/lpc5411x/chip_5411x/inc/usbd.h`

## 7.68 USB\_INTERFACE\_DESCRIPTOR Struct Reference

### 7.68.1 Detailed Description

USB Standard Interface Descriptor

Definition at line 522 of file `usbd.h`.

```
#include "usbd.h"
```

#### Data Fields

- `uint8_t bLength`
- `uint8_t bDescriptorType`
- `uint8_t bInterfaceNumber`
- `uint8_t bAlternateSetting`
- `uint8_t bNumEndpoints`
- `uint8_t bInterfaceClass`
- `uint8_t bInterfaceSubClass`
- `uint8_t bInterfaceProtocol`
- `uint8_t iInterface`

### 7.68.2 Field Documentation

#### 7.68.2.1 `uint8_t bAlternateSetting`

Value used to select this alternate setting for the interface identified in the prior field

Definition at line 530 of file `usbd.h`.

#### 7.68.2.2 `uint8_t bDescriptorType`

INTERFACE Descriptor Type

Definition at line 525 of file `usbd.h`.

#### 7.68.2.3 uint8\_t bInterfaceClass

Class code (assigned by the USB-IF).

A value of zero is reserved for future standardization.

If this field is set to FFH, the interface class is vendor-specific.

All other values are reserved for assignment by the USB-IF.

Definition at line 536 of file usbd.h.

#### 7.68.2.4 uint8\_t bInterfaceNumber

Number of this interface. Zero-based value identifying the index in the array of concurrent interfaces supported by this configuration.

Definition at line 526 of file usbd.h.

#### 7.68.2.5 uint8\_t bInterfaceProtocol

Protocol code (assigned by the USB).

These codes are qualified by the value of the bInterfaceClass and the bInterfaceSubClass fields. If an interface supports class-specific requests, this code identifies the protocols that the device uses as defined by the specification of the device class.

If this field is reset to zero, the device does not use a class-specific protocol on this interface.

If this field is set to FFH, the device uses a vendor-specific protocol for this interface.

Definition at line 551 of file usbd.h.

#### 7.68.2.6 uint8\_t bInterfaceSubClass

Subclass code (assigned by the USB-IF).

These codes are qualified by the value of the bInterfaceClass field.

If the bInterfaceClass field is reset to zero, this field must also be reset to zero.

If the bInterfaceClass field is not set to FFH, all values are reserved for assignment by the USB-IF.

Definition at line 543 of file usbd.h.

#### 7.68.2.7 uint8\_t bLength

Size of this descriptor in bytes

Definition at line 524 of file usbd.h.

#### 7.68.2.8 uint8\_t bNumEndpoints

Number of endpoints used by this interface (excluding endpoint zero). If this value is zero, this interface only uses the Default Control Pipe.

Definition at line 532 of file usbd.h.

#### 7.68.2.9 uint8\_t iInterface

Index of string descriptor describing this interface

Definition at line 565 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.69 USB\_OTHER\_SPEED\_CONFIGURATION Struct Reference

### 7.69.1 Detailed Description

USB Other Speed Configuration

Definition at line 673 of file usbd.h.

```
#include "usbd.h"
```

#### Data Fields

- [uint8\\_t bLength](#)
- [uint8\\_t bDescriptorType](#)
- [uint16\\_t wTotalLength](#)
- [uint8\\_t bNumInterfaces](#)
- [uint8\\_t bConfigurationValue](#)
- [uint8\\_t IConfiguration](#)
- [uint8\\_t bmAttributes](#)
- [uint8\\_t bMaxPower](#)

### 7.69.2 Field Documentation

#### 7.69.2.1 uint8\_t bConfigurationValue

Value to use to select configuration

Definition at line 679 of file usbd.h.

#### 7.69.2.2 uint8\_t bDescriptorType

Other\_speed\_Configuration Type

Definition at line 676 of file usbd.h.

#### 7.69.2.3 uint8\_t bLength

Size of descriptor

Definition at line 675 of file usbd.h.

#### 7.69.2.4 uint8\_t bmAttributes

Same as Configuration descriptor

Definition at line 681 of file usbd.h.

#### 7.69.2.5 uint8\_t bMaxPower

Same as Configuration descriptor

Definition at line 682 of file usbd.h.

#### 7.69.2.6 uint8\_t bNumInterfaces

Number of interfaces supported by this speed configuration

Definition at line 678 of file usbd.h.

#### 7.69.2.7 uint8\_t lConfiguration

Index of string descriptor

Definition at line 680 of file usbd.h.

#### 7.69.2.8 uint16\_t wTotalLength

Total length of data returned

Definition at line 677 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.70 USB\_SETUP\_PACKET Struct Reference

### 7.70.1 Detailed Description

USB Default Control Pipe Setup Packet

Definition at line 199 of file usbd.h.

```
#include "usbd.h"
```

#### Data Fields

- REQUEST\_TYPE [bmRequestType](#)
- uint8\_t [bRequest](#)
- WORD\_BYTE [wValue](#)
- WORD\_BYTE [wIndex](#)
- uint16\_t [wLength](#)

### 7.70.2 Field Documentation

#### 7.70.2.1 REQUEST\_TYPE bmRequestType

This bitmapped field identifies the characteristics of the specific request.

See also

[\\_BM\\_T](#).

Definition at line 201 of file usbd.h.

#### 7.70.2.2 uint8\_t bRequest

This field specifies the particular request. The Type bits in the bmRequestType field modify the meaning of this field.

See also

USBD\_REQUEST.

Definition at line 204 of file usbd.h.

#### 7.70.2.3 WORD\_BYTE wIndex

Used to pass a parameter to the device, specific to the request. The wIndex field is often used in requests to specify an endpoint or an interface.

Definition at line 211 of file usbd.h.

#### 7.70.2.4 uint16\_t wLength

This field specifies the length of the data transferred during the second phase of the control transfer.

Definition at line 215 of file usbd.h.

#### 7.70.2.5 WORD\_BYTE wValue

Used to pass a parameter to the device, specific to the request.

Definition at line 208 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.71 USB\_STRING\_DESCRIPTOR Struct Reference

### 7.71.1 Detailed Description

USB String Descriptor

Definition at line 654 of file usbd.h.

```
#include "usbd.h"
```

#### Data Fields

- uint8\_t [bLength](#)
- uint8\_t [bDescriptorType](#)
- uint16\_t [bString](#)

### 7.71.2 Field Documentation

#### 7.71.2.1 uint8\_t bDescriptorType

STRING Descriptor Type

Definition at line 657 of file usbd.h.

### 7.71.2.2 uint8\_t bLength

Size of this descriptor in bytes

Definition at line 656 of file usbd.h.

### 7.71.2.3 uint16\_t bString

UNICODE encoded string

Definition at line 658 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.72 WB\_T Struct Reference

### 7.72.1 Detailed Description

Structure to pack lower and upper byte to form 16 bit word.

Definition at line 78 of file usbd.h.

```
#include "usbd.h"
```

#### Data Fields

- [uint8\\_t L](#)
- [uint8\\_t H](#)

### 7.72.2 Field Documentation

#### 7.72.2.1 uint8\_t H

upper byte

Definition at line 81 of file usbd.h.

#### 7.72.2.2 uint8\_t L

lower byte

Definition at line 80 of file usbd.h.

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

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[usbd.h](#)

## 7.73 WORD\_BYTE Union Reference

### 7.73.1 Detailed Description

Union of [\\_WB\\_T](#) struct and 16 bit word.

Definition at line 87 of file usbd.h.

```
#include "usbd.h"
```

## Data Fields

- [uint16\\_t](#) [W](#)
- [WB\\_T](#) [WB](#)

### 7.73.2 Field Documentation

#### 7.73.2.1 [uint16\\_t](#) [W](#)

data member to do 16 bit access

Definition at line 89 of file [usbd.h](#).

#### 7.73.2.2 [WB\\_T](#) [WB](#)

data member to do 8 bit access

Definition at line 90 of file [usbd.h](#).

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

- [C:/Jenkins/LPC5411x/lpc5411x/chip\\_5411x/inc/usbd.h](#)

## 7.74 xPSR\_Type Union Reference

### 7.74.1 Detailed Description

Union type to access the Special-Purpose Program Status Registers (xPSR).

Definition at line 242 of file [core\\_cm0plus.h](#).

```
#include "core_cm0plus.h"
```

## Data Fields

- struct {
  - [uint32\\_t](#) [ISR](#):9
  - [uint32\\_t](#) [\\_reserved0](#):15
  - [uint32\\_t](#) [T](#):1
  - [uint32\\_t](#) [IT](#):2
  - [uint32\\_t](#) [Q](#):1
  - [uint32\\_t](#) [V](#):1
  - [uint32\\_t](#) [C](#):1
  - [uint32\\_t](#) [Z](#):1
  - [uint32\\_t](#) [N](#):1
- [uint32\\_t](#) [w](#)
- struct {
  - [uint32\\_t](#) [ISR](#):9
  - [uint32\\_t](#) [\\_reserved0](#):7
  - [uint32\\_t](#) [GE](#):4
  - [uint32\\_t](#) [\\_reserved1](#):4



```

uint32_t T:1
uint32_t IT:2
uint32_t Q:1
        uint32_t V:1
        uint32_t C:1
        uint32_t Z:1
        uint32_t N:1
    } b

```

## 7.74.2 Field Documentation

### 7.74.2.1 uint32\_t \_reserved0

bit: 9..23 Reserved

bit: 9..15 Reserved

Definition at line 248 of file core\_cm0plus.h.

### 7.74.2.2 uint32\_t \_reserved1

bit: 20..23 Reserved

Definition at line 297 of file core\_cm4.h.

### 7.74.2.3 struct { ... } b

Structure used for bit access

### 7.74.2.4 struct { ... } b

Structure used for bit access

### 7.74.2.5 uint32\_t C

bit: 29 Carry condition code flag

Definition at line 258 of file core\_cm0plus.h.

### 7.74.2.6 uint32\_t GE

bit: 16..19 Greater than or Equal flags

Definition at line 296 of file core\_cm4.h.

### 7.74.2.7 uint32\_t ISR

bit: 0.. 8 Exception number

Definition at line 246 of file core\_cm0plus.h.

### 7.74.2.8 uint32\_t IT

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

Definition at line 255 of file core\_cm0plus.h.

#### 7.74.2.9 uint32\_t N

bit: 31 Negative condition code flag

Definition at line 260 of file core\_cm0plus.h.

#### 7.74.2.10 uint32\_t Q

bit: 27 Saturation condition flag

Definition at line 256 of file core\_cm0plus.h.

#### 7.74.2.11 uint32\_t T

bit: 24 Thumb bit (read 0)

Definition at line 254 of file core\_cm0plus.h.

#### 7.74.2.12 uint32\_t V

bit: 28 Overflow condition code flag

Definition at line 257 of file core\_cm0plus.h.

#### 7.74.2.13 uint32\_t w

Type used for word access

Definition at line 262 of file core\_cm0plus.h.

#### 7.74.2.14 uint32\_t Z

bit: 30 Zero condition code flag

Definition at line 259 of file core\_cm0plus.h.

The documentation for this union was generated from the following files:

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm0plus.h](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/[core\\_cm4.h](#)

## Chapter 8

# File Documentation

### 8.1 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/adc\_5411x.h File Reference

#### Data Structures

- struct [LPC\\_ADC\\_T](#)  
*ADC register block structure.*

#### Macros

- #define [ADC\\_MAX\\_SAMPLE\\_RATE](#) 80000000
- #define [ADC\\_MAX\\_CHANNEL\\_NUM](#) 12
- #define [ADC\\_CR\\_CLKDIV\\_MASK](#) (0xFF << 0)  
*ADC register support bitfields and mask.*
- #define [ADC\\_CR\\_CLKDIV\\_BITPOS](#) (0)
- #define [ADC\\_CR\\_ASYNC\\_MODE](#) (1 << 8)
- #define [ADC\\_CR\\_RESOL\(n\)](#) ((n) << 9)
- #define [ADC\\_CR\\_LPWRMODEBIT](#) (1 << 10)
- #define [ADC\\_CR\\_BYPASS](#) (1 << 11)
- #define [ADC\\_CR\\_TSAMP\(n\)](#) ((n) << 12)
- #define [ADC\\_CR\\_CALMODEBIT](#) (1 << 30)
- #define [ADC\\_CR\\_BITACC\(n\)](#) (((n) & 0x1) << 9)
- #define [ADC\\_CR\\_CLKDIV\(n\)](#) (((n) & 0xFF) << 0)
- #define [ADC\\_SAMPLE\\_RATE\\_CONFIG\\_MASK](#) (ADC\_CR\_CLKDIV(0xFF) | ADC\_CR\_BITACC(0x01))
- #define [ADC\\_SEQ\\_CTRL\\_CHANNEL\\_EN\(n\)](#) (1 << n)
- #define [ADC\\_SEQ\\_CTRL\\_TRIGGER\(n\)](#) ((n & 0x3f) << 12)
- #define [ADC\\_SEQ\\_CTRL\\_HWTRIG\\_POLPOS](#) (1 << 18)
- #define [ADC\\_SEQ\\_CTRL\\_HWTRIG\\_SYNCBYPASS](#) (1 << 19)
- #define [ADC\\_SEQ\\_CTRL\\_START](#) (1 << 26)
- #define [ADC\\_SEQ\\_CTRL\\_BURST](#) (1 << 27)
- #define [ADC\\_SEQ\\_CTRL\\_SINGLESTEP](#) (1 << 28)
- #define [ADC\\_SEQ\\_CTRL\\_LOWPRIO](#) (1 << 29)
- #define [ADC\\_SEQ\\_CTRL\\_MODE\\_EOS](#) (1 << 30)
- #define [ADC\\_SEQ\\_CTRL\\_SEQ\\_ENA](#) (1UL << 31)
- #define [ADC\\_SEQ\\_GDAT\\_RESULT\\_MASK](#) (0xFFF << 4)
- #define [ADC\\_SEQ\\_GDAT\\_RESULT\\_BITPOS](#) (4)
- #define [ADC\\_SEQ\\_GDAT\\_THCMPRANGE\\_MASK](#) (0x3 << 16)
- #define [ADC\\_SEQ\\_GDAT\\_THCMPRANGE\\_BITPOS](#) (16)
- #define [ADC\\_SEQ\\_GDAT\\_THCMPCROSS\\_MASK](#) (0x3 << 18)

- `#define ADC_SEQ_GDAT_THCMPCROSS_BITPOS` (18)
- `#define ADC_SEQ_GDAT_CHAN_MASK` (0xF << 26)
- `#define ADC_SEQ_GDAT_CHAN_BITPOS` (26)
- `#define ADC_SEQ_GDAT_OVERRUN` (1 << 30)
- `#define ADC_SEQ_GDAT_DATAVALID` (1UL << 31)
- `#define ADC_DR_RESULT_BITPOS` (4)
- `#define ADC_DR_RESULT(n)` (((n) >> 4) & 0xFFF)
- `#define ADC_DR_THCMPRANGE_MASK` (0x3 << 16)
- `#define ADC_DR_THCMPRANGE_BITPOS` (16)
- `#define ADC_DR_THCMPRANGE(n)` (((n) >> ADC\_DR\_THCMPRANGE\_BITPOS) & 0x3)
- `#define ADC_DR_THCMPCROSS_MASK` (0x3 << 18)
- `#define ADC_DR_THCMPCROSS_BITPOS` (18)
- `#define ADC_DR_THCMPCROSS(n)` (((n) >> ADC\_DR\_THCMPCROSS\_BITPOS) & 0x3)
- `#define ADC_DR_CHAN_MASK` (0xF << 26)
- `#define ADC_DR_CHAN_BITPOS` (26)
- `#define ADC_DR_CHANNEL(n)` (((n) >> ADC\_DR\_CHAN\_BITPOS) & 0xF)
- `#define ADC_DR_OVERRUN` (1 << 30)
- `#define ADC_DR_DATAVALID` (1UL << 31)
- `#define ADC_DR_DONE(n)` (((n) >> 31))
- `#define ADC_THR_VAL_MASK` (0xFFF << 4)
- `#define ADC_THR_VAL_POS` (4)
- `#define ADC_THRSEL_CHAN_SEL_THR1(n)` (1 << (n))
- `#define ADC_INTEN_SEQA_ENABLE` (1 << 0)
- `#define ADC_INTEN_SEQB_ENABLE` (1 << 1)
- `#define ADC_INTEN_SEQN_ENABLE(seq)` (1 << (seq))
- `#define ADC_INTEN_OVRRUN_ENABLE` (1 << 2)
- `#define ADC_INTEN_CMP_DISBALE` (0)
- `#define ADC_INTEN_CMP_OUTSIDETH` (1)
- `#define ADC_INTEN_CMP_CROSSTH` (2)
- `#define ADC_INTEN_CMP_MASK` (3)
- `#define ADC_INTEN_CMP_ENABLE(isel, ch)` (((isel) & ADC\_INTEN\_CMP\_MASK) << ((2 \* (ch)) + 3))
- `#define ADC_FLAGS_THCMP_MASK(ch)` (1 << (ch))
- `#define ADC_FLAGS_OVRRUN_MASK(ch)` (1 << (12 + (ch)))
- `#define ADC_FLAGS_SEQA_OVRRUN_MASK` (1 << 24)
- `#define ADC_FLAGS_SEQB_OVRRUN_MASK` (1 << 25)
- `#define ADC_FLAGS_SEQN_OVRRUN_MASK(seq)` (1 << (24 + (seq)))
- `#define ADC_FLAGS_SEQA_INT_MASK` (1 << 28)
- `#define ADC_FLAGS_SEQB_INT_MASK` (1 << 29)
- `#define ADC_FLAGS_SEQN_INT_MASK(seq)` (1 << (28 + (seq)))
- `#define ADC_FLAGS_THCMP_INT_MASK` (1 << 30)
- `#define ADC_FLAGS_OVRRUN_INT_MASK` (1UL << 31)
- `#define ADC_STARTUP_ENABLE` (0x1 << 0)
- `#define ADC_STARTUP_INIT` (0x1 << 1)
- `#define ADC_CALIB` (0x1 << 0)
- `#define ADC_CALREQD` (0x1 << 1)

## Enumerations

- `enum ADC_SEQ_IDX_T` { `ADC_SEQA_IDX` = 0, `ADC_SEQB_IDX` }
- `enum ADC_TSAMP_T` {  
`ADC_TSAMP_2CLK5` = 0, `ADC_TSAMP_3CLK5`, `ADC_TSAMP_4CLK5`, `ADC_TSAMP_5CLK5`,  
`ADC_TSAMP_6CLK5`, `ADC_TSAMP_7CLK5`, `ADC_TSAMP_8CLK5`, `ADC_TSAMP_9CLK5` }

*ADC sampling time bits 12, 13 and 14.*

- enum `ADC_DR_THCMPRANGE_T` { `ADC_DR_THCMPRANGE_INRANGE`, `ADC_DR_THCMPRANGE_RESERVED`, `ADC_DR_THCMPRANGE_BELOW`, `ADC_DR_THCMPRANGE_ABOVE` }
- enum `ADC_DR_THCMPCROSS_T` { `ADC_DR_THCMPCROSS_NOCROSS`, `ADC_DR_THCMPCROSS_RESERVED`, `ADC_DR_THCMPCROSS_DOWNWARD`, `ADC_DR_THCMPCROSS_UPWARD` }
- enum `ADC_INTEN_THCMP_T` { `ADC_INTEN_THCMP_DISABLE`, `ADC_INTEN_THCMP_OUTSIDE`, `ADC_INTEN_THCMP_CROSSING` }

## Functions

- void `Chip_ADC_Init` (`LPC_ADC_T *pADC`, `uint32_t flags`)  
*Initialize the ADC peripheral.*
- void `Chip_ADC_DeInit` (`LPC_ADC_T *pADC`)  
*Shutdown ADC.*
- `__STATIC_INLINE` void `Chip_ADC_SetDivider` (`LPC_ADC_T *pADC`, `uint8_t div`)  
*Set ADC divider.*
- void `Chip_ADC_SetClockRate` (`LPC_ADC_T *pADC`, `uint32_t rate`)  
*Set ADC clock rate.*
- `__STATIC_INLINE` `uint8_t` `Chip_ADC_GetDivider` (`LPC_ADC_T *pADC`)  
*Get ADC divider.*
- `uint32_t` `Chip_ADC_Calibration` (`LPC_ADC_T *pADC`)  
*Perform ADC calibration.*
- `__STATIC_INLINE` void `Chip_ADC_SelectTempSensorInput` (`LPC_ADC_T *pADC`)  
*Selects Temperature sensor as the input for Channel 0.*
- `__STATIC_INLINE` void `Chip_ADC_SetSequencerBits` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`, `uint32_t bits`)  
*Helper function for safely setting ADC sequencer register bits.*
- `__STATIC_INLINE` void `Chip_ADC_ClearSequencerBits` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`, `uint32_t bits`)  
*Helper function for safely clearing ADC sequencer register bits.*
- `__STATIC_INLINE` void `Chip_ADC_SetupSequencer` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`, `uint32_t options`)  
*Sets up ADC conversion sequencer A or B.*
- `__STATIC_INLINE` `uint32_t` `Chip_ADC_GetSequencerCtrl` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`)  
*Get sequenceX control register value.*
- `__STATIC_INLINE` void `Chip_ADC_EnableSequencer` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`)  
*Enables a sequencer.*
- `__STATIC_INLINE` void `Chip_ADC_DisableSequencer` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`)  
*Disables a sequencer.*
- `__STATIC_INLINE` void `Chip_ADC_StartSequencer` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`)  
*Forces a sequencer trigger event (software trigger of ADC)*
- `__STATIC_INLINE` void `Chip_ADC_StartBurstSequencer` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`)  
*Starts sequencer burst mode.*
- `__STATIC_INLINE` void `Chip_ADC_StopBurstSequencer` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`)  
*Stops sequencer burst mode.*
- `__STATIC_INLINE` `uint32_t` `Chip_ADC_GetGlobalDataReg` (`LPC_ADC_T *pADC`, `ADC_SEQ_IDX_T seqIndex`)  
*Read a ADC sequence global data register.*
- `__STATIC_INLINE` `uint32_t` `Chip_ADC_GetDataReg` (`LPC_ADC_T *pADC`, `uint8_t index`)  
*Read a ADC data register.*

- `__STATIC_INLINE void Chip_ADC_SetThrLowValue (LPC_ADC_T *pADC, uint8_t thrnum, uint16_t value)`  
*Set Threshold low value in ADC.*
- `__STATIC_INLINE void Chip_ADC_SetThrHighValue (LPC_ADC_T *pADC, uint8_t thrnum, uint16_t value)`  
*Set Threshold high value in ADC.*
- `__STATIC_INLINE void Chip_ADC_SelectTH0Channels (LPC_ADC_T *pADC, uint32_t channels)`  
*Select threshold 0 values for comparison for selected channels.*
- `__STATIC_INLINE void Chip_ADC_SelectTH1Channels (LPC_ADC_T *pADC, uint32_t channels)`  
*Select threshold 1 value for comparison for selected channels.*
- `__STATIC_INLINE void Chip_ADC_EnableInt (LPC_ADC_T *pADC, uint32_t intMask)`  
*Enable interrupts in ADC (sequencers A/B and overrun)*
- `__STATIC_INLINE void Chip_ADC_DisableInt (LPC_ADC_T *pADC, uint32_t intMask)`  
*Disable interrupts in ADC (sequencers A/B and overrun)*
- `__STATIC_INLINE void Chip_ADC_SetThresholdInt (LPC_ADC_T *pADC, uint8_t ch, ADC_INTEN_THCMP_T thInt)`  
*Enable a threshold event interrupt in ADC.*
- `__STATIC_INLINE uint32_t Chip_ADC_GetFlags (LPC_ADC_T *pADC)`  
*Get flags register in ADC.*
- `__STATIC_INLINE void Chip_ADC_ClearFlags (LPC_ADC_T *pADC, uint32_t flags)`  
*Clear flags register in ADC.*
- `__STATIC_INLINE void Chip_ADC_SetTHRSELBits (LPC_ADC_T *pADC, uint32_t mask)`  
*Set Threshold selection bits.*
- `__STATIC_INLINE void Chip_ADC_ClearTHRSELBits (LPC_ADC_T *pADC, uint32_t mask)`  
*Clear Threshold selection bits.*

## 8.2 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/chip.h File Reference

```
#include "lpc_types.h"
```

```
#include "cmsis.h"
#include "lpc_assert.h"
#include "romapi_5411x.h"
#include "syscon_5411x.h"
#include "cpuctrl_5411x.h"
#include "clock_5411x.h"
#include "pmu_5411x.h"
#include "iocon_5411x.h"
#include "pinint_5411x.h"
#include "inmux_5411x.h"
#include "crc_5411x.h"
#include "gpio_5411x.h"
#include "mrt_5411x.h"
#include "wwdt_5411x.h"
#include "sct_5411x.h"
#include "sct_pwm_5411x.h"
#include "rtc_5411x.h"
#include "timer_5411x.h"
#include "utick_5411x.h"
#include "gpiogroup_5411x.h"
#include "mailbox_5411x.h"
#include "fpu_init.h"
#include "power_lib_5411x.h"
#include "flexcomm_5411x.h"
#include "usbd_5411x.h"
#include "adc_5411x.h"
#include "dma_5411x.h"
#include "dma_service_5411x.h"
#include "dmic_5411x.h"
#include "uart_5411x.h"
#include "spi_common_5411x.h"
#include "spim_5411x.h"
#include "spis_5411x.h"
#include "i2c_common_5411x.h"
#include "i2cm_5411x.h"
#include "i2cs_5411x.h"
#include "i2s_5411x.h"
```

## Macros

- `#define LPC_FLASHMEM_BASE 0x00000000UL`
- `#define LPC_SRAMX_BASE 0x04000000UL`
- `#define LPC_SRAM0_BASE 0x20000000UL`
- `#define LPC_SRAM1_BASE 0x20010000UL`
- `#define LPC_SRAM2_BASE 0x20018000UL`
- `#define LPC_ROM_BASE 0x03000000UL`
- `#define LPC_SYSCON_BASE 0x40000000UL`
- `#define LPC_IOCON_BASE 0x40001000UL`
- `#define LPC_GPIO_GROUPINT0_BASE 0x40002000UL`
- `#define LPC_GPIO_GROUPINT1_BASE 0x40003000UL`
- `#define LPC_PIN_INT_BASE 0x40004000UL`
- `#define LPC_INMUX_BASE 0x40005000UL`
- `#define LPC_TIMER0_BASE 0x40008000UL`
- `#define LPC_TIMER1_BASE 0x40009000UL`
- `#define LPC_WWDT_BASE 0x4000C000UL`

- #define `LPC_MRT_BASE` 0x4000D000UL
- #define `LPC_UTICK_BASE` 0x4000E000UL
- #define `LPC_PMU_BASE` 0x40020000UL
- #define `LPC_TIMER2_BASE` 0x40028000UL
- #define `LPC_RTC_BASE` 0x4002C000UL
- #define `LPC_FMC_BASE` 0x40034000UL
- #define `LPC_ASYNC_SYSCON_BASE` 0x40040000UL
- #define `LPC_TIMER3_BASE` 0x40048000UL
- #define `LPC_TIMER4_BASE` 0x40049000UL
- #define `LPC_SPIFI_BASE` 0x40080000UL
- #define `LPC_DMA_BASE` 0x40082000UL
- #define `LPC_USB_BASE` 0x40084000UL
- #define `LPC_SCT_BASE` 0x40085000UL
- #define `LPC_FLEXCOMM0_BASE` 0x40086000UL
- #define `LPC_FLEXCOMM1_BASE` 0x40087000UL
- #define `LPC_FLEXCOMM2_BASE` 0x40088000UL
- #define `LPC_FLEXCOMM3_BASE` 0x40089000UL
- #define `LPC_FLEXCOMM4_BASE` 0x4008A000UL
- #define `LPC_MBOX_BASE` 0x4008B000UL
- #define `LPC_GPIO_PORT_BASE` 0x4008C000UL
- #define `LPC_DMIC_BASE` 0x40090000UL
- #define `LPC_CRC_BASE` 0x40095000UL
- #define `LPC_FLEXCOMM5_BASE` 0x40096000UL
- #define `LPC_FLEXCOMM6_BASE` 0x40097000UL
- #define `LPC_FLEXCOMM7_BASE` 0x40098000UL
- #define `LPC_ISPAP_BASE` 0x4009C000UL
- #define `LPC_ADC_BASE` 0x400A0000UL
- #define `LPC_GPIO` ((`LPC_GPIO_T` \*) `LPC_GPIO_PORT_BASE`)
- #define `LPC_DMA` ((`LPC_DMA_T` \*) `LPC_DMA_BASE`)
- #define `LPC_CRC` ((`LPC_CRC_T` \*) `LPC_CRC_BASE`)
- #define `LPC_SCT` ((`LPC_SCT_T` \*) `LPC_SCT_BASE`)
- #define `LPC_MBOX` ((`LPC_MBOX_T` \*) `LPC_MBOX_BASE`)
- #define `LPC_ADC` ((`LPC_ADC_T` \*) `LPC_ADC_BASE`)
- #define `LPC_PMU` ((`LPC_PMU_T` \*) `LPC_PMU_BASE`)
- #define `LPC_DMIC` ((`LPC_DMIC_T` \*) `LPC_DMIC_BASE`)
- #define `LPC_USB` ((`LPC_USB_T` \*) `LPC_USB_BASE`)
- #define `LPC_SYSCON` ((`LPC_SYSCON_T` \*) `LPC_SYSCON_BASE`)
- #define `LPC_TIMER2` ((`LPC_TIMER_T` \*) `LPC_TIMER2_BASE`)
- #define `LPC_TIMER3` ((`LPC_TIMER_T` \*) `LPC_TIMER3_BASE`)
- #define `LPC_TIMER4` ((`LPC_TIMER_T` \*) `LPC_TIMER4_BASE`)
- #define `LPC_GINT` ((`LPC_GPIOGROUPINT_T` \*) `LPC_GPIO_GROUPINT0_BASE`)
- #define `LPC_PININT` ((`LPC_PIN_INT_T` \*) `LPC_PIN_INT_BASE`)
- #define `LPC_IOCON` ((`LPC_IOCON_T` \*) `LPC_IOCON_BASE`)
- #define `LPC_UTICK` ((`LPC_UTICK_T` \*) `LPC_UTICK_BASE`)
- #define `LPC_WWDT` ((`LPC_WWDT_T` \*) `LPC_WWDT_BASE`)
- #define `LPC_RTC` ((`LPC_RTC_T` \*) `LPC_RTC_BASE`)
- #define `LPC_ASYNC_SYSCON` ((`LPC_ASYNC_SYSCON_T` \*) `LPC_ASYNC_SYSCON_BASE`)
- #define `LPC_TIMER0` ((`LPC_TIMER_T` \*) `LPC_TIMER0_BASE`)
- #define `LPC_TIMER1` ((`LPC_TIMER_T` \*) `LPC_TIMER1_BASE`)
- #define `LPC_INMUX` ((`LPC_INMUX_T` \*) `LPC_INMUX_BASE`)
- #define `LPC_MRT` ((`LPC_MRT_T` \*) `LPC_MRT_BASE`)



## Functions

- void [SystemCoreClockUpdate](#) (void)  
*Update system core and ASYNC syscon clock rate, should be called if the system has a clock rate change.*
- void [Chip\\_SystemInit](#) (void)  
*Set up and initialize hardware prior to call to main()*
- void [Chip\\_SetupIrcClocking](#) (uint32\_t iFreq)  
*Clock and PLL initialization based on the internal oscillator.*
- void [Chip\\_SetupExtInClocking](#) (uint32\_t iFreq)  
*Clock and PLL initialization based on the external clock input.*
- void [Chip\\_SetupFROClocking](#) (uint32\_t iFreq)  
*Initialize the Core clock to given frequency (12, 48 or 96 MHz)*
- void [Chip\\_USB\\_Init](#) (void)  
*Initialize the USB bus.*
- `__STATIC_INLINE` void [Chip\\_USB\\_TrimOff](#) (int enable)  
*Turn of FRO clock trimming based on USB SOF.*

## Variables

- uint32\_t [SystemCoreClock](#)  
*Current system clock rate, mainly used for peripherals in SYSCON.*
- const uint32\_t [ExtClockIn](#)  
*Clock rate on the CLKIN pin This value is defined externally to the chip layer and contains the value in Hz for the CLKIN pin for the board. If this pin isn't used, this rate can be 0.*

## 8.3 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/clock\_5411x.h File Reference

```
#include "pll_5411x.h"
```

## Macros

- `#define` [SYSCON\\_FRO12MHZ\\_FREQ](#) (12000000)
- `#define` [SYSCON\\_FRO48MHZ\\_FREQ](#) (48000000)
- `#define` [SYSCON\\_FRO96MHZ\\_FREQ](#) (96000000)
- `#define` [SYSCON\\_WDTOSC\\_FREQ](#) (500000)
- `#define` [SYSCON\\_RTC\\_FREQ](#) (32768)
- `#define` [Chip\\_Clock\\_GetIntOscRate](#)() [SYSCON\\_FRO12MHZ\\_FREQ](#)

## Enumerations

- enum [WDT\\_OSC\\_FREQ\\_T](#) {  
[WDT\\_FREQ\\_RESERVED](#), [WDT\\_FREQ\\_400000](#), [WDT\\_FREQ\\_600000](#), [WDT\\_FREQ\\_750000](#),  
[WDT\\_FREQ\\_900000](#), [WDT\\_FREQ\\_1000000](#), [WDT\\_FREQ\\_1200000](#), [WDT\\_FREQ\\_1300000](#),  
[WDT\\_FREQ\\_1400000](#), [WDT\\_FREQ\\_1500000](#), [WDT\\_FREQ\\_1600000](#), [WDT\\_FREQ\\_1700000](#),  
[WDT\\_FREQ\\_1800000](#), [WDT\\_FREQ\\_1900000](#), [WDT\\_FREQ\\_2000000](#), [WDT\\_FREQ\\_2050000](#),  
[WDT\\_FREQ\\_2100000](#), [WDT\\_FREQ\\_2200000](#), [WDT\\_FREQ\\_2250000](#), [WDT\\_FREQ\\_2300000](#),  
[WDT\\_FREQ\\_2400000](#), [WDT\\_FREQ\\_2450000](#), [WDT\\_FREQ\\_2500000](#), [WDT\\_FREQ\\_2600000](#),  
[WDT\\_FREQ\\_2650000](#), [WDT\\_FREQ\\_2700000](#), [WDT\\_FREQ\\_2800000](#), [WDT\\_FREQ\\_2850000](#),  
[WDT\\_FREQ\\_2900000](#), [WDT\\_FREQ\\_2950000](#), [WDT\\_FREQ\\_3000000](#), [WDT\\_FREQ\\_3050000](#) }

*WDT Osc frequency value table.*

- enum `CHIP_SYSCON_MAIN_A_CLKSRC_T` { `SYSCON_MAIN_A_CLKSRC_FRO12MHZ` = 0, `SYSCON_MAIN_A_CLKSRC_CLKIN`, `SYSCON_MAIN_A_CLKSRC_WDTOSC`, `SYSCON_MAIN_A_CLKSRC_FROHF` }
- enum `CHIP_SYSCON_USBCLKSRC_T` { `SYSCON_USBCLKSRC_FROHF`, `SYSCON_USBCLKSRC_PLL`, `SYSCON_USBCLKSRC_DISABLED` = 7 }

*USB Clock source.*

- enum `CHIP_SYSCON_MCLKSRC_T` { `SYSCON_MCLKSRC_FROHF`, `SYSCON_MCLKSRC_PLL`, `SYSCON_MCLKSRC_CLKIN`, `SYSCON_MCLKSRC_DISABLED` = 7 }

*MCLK Clock sources.*

- enum `CHIP_SYSCON_MAIN_B_CLKSRC_T` { `SYSCON_MAIN_B_CLKSRC_MAINCLKSELA` = 0, `SYSCON_MAIN_B_CLKSRC_PLL` = 2, `SYSCON_MAIN_B_CLKSRC_RTC` }
- enum `CHIP_SYSCON_CLKOUTSRC_T` { `SYSCON_CLKOUTSRC_MAINCLK` = 0, `SYSCON_CLKOUTSRC_CLKIN`, `SYSCON_CLKOUTSRC_WDTOSC`, `SYSCON_CLKOUTSRC_FROHF`, `SYSCON_CLKOUTSRC_PLL`, `SYSCON_CLKOUTSRC_FRO12MHZ`, `SYSCON_CLKOUTSRC_RTC`, `SYSCON_CLKOUTSRC_DISABLED` }
- enum `CHIP_SYSCON_CLOCK_T` { `SYSCON_CLOCK_ROM` = 1, `SYSCON_CLOCK_SRAM1` = 3, `SYSCON_CLOCK_SRAM2`, `SYSCON_CLOCK_SRAMX`, `SYSCON_CLOCK_FLASH` = 7, `SYSCON_CLOCK_FMC`, `SYSCON_CLOCK_SPIFI` = 10, `SYSCON_CLOCK_INPUTMUX`, `SYSCON_CLOCK_IOCON` = 13, `SYSCON_CLOCK_GPIO0`, `SYSCON_CLOCK_GPIO1`, `SYSCON_CLOCK_PINT` = 18, `SYSCON_CLOCK_GINT`, `SYSCON_CLOCK_DMA`, `SYSCON_CLOCK_CRC`, `SYSCON_CLOCK_WWDT`, `SYSCON_CLOCK_RTC`, `SYSCON_CLOCK_MAILBOX` = 26, `SYSCON_CLOCK_ADC0`, `SYSCON_CLOCK_MRT` = 32, `SYSCON_CLOCK_SCT0` = 32 + 2, `SYSCON_CLOCK_UTICK` = 32 + 10, `SYSCON_CLOCK_FLEXCOMM0`, `SYSCON_CLOCK_FLEXCOMM1`, `SYSCON_CLOCK_FLEXCOMM2`, `SYSCON_CLOCK_FLEXCOMM3`, `SYSCON_CLOCK_FLEXCOMM4`, `SYSCON_CLOCK_FLEXCOMM5`, `SYSCON_CLOCK_FLEXCOMM6`, `SYSCON_CLOCK_FLEXCOMM7`, `SYSCON_CLOCK_DMIC`, `SYSCON_CLOCK_TIMER2` = 32 + 22, `SYSCON_CLOCK_USB` = 32 + 25, `SYSCON_CLOCK_TIMER0`, `SYSCON_CLOCK_TIMER1`, `SYSCON_CLOCK_TIMER3` = 128 + 13, `SYSCON_CLOCK_TIMER4` }
- enum `CHIP_SYSCON_FLEXCOMMCLKSELSRC_T` { `SYSCON_FLEXCOMMCLKSELSRC_FRO12MHZ` = 0, `SYSCON_FLEXCOMMCLKSELSRC_FROHF`, `SYSCON_FLEXCOMMCLKSELSRC_PLL`, `SYSCON_FLEXCOMMCLKSELSRC_MCLK`, `SYSCON_FLEXCOMMCLKSELSRC_FRG`, `SYSCON_FLEXCOMMCLKSELSRC_NONE` = 7 }
- enum `CHIP_SYSCON_ADCCLKSELSRC_T` { `SYSCON_ADCCLKSELSRC_MAINCLK` = 0, `SYSCON_ADCCLKSELSRC_SYSPLLOUT`, `SYSCON_ADCCLKSELSRC_FROHF` }
- enum `CHIP_ASYNC_SYSCON_SRC_T` { `SYSCON_ASYNC_MAINCLK` = 0, `SYSCON_ASYNC_FRO12MHZ` }
- enum `CHIP_SYSCON_MAINCLKSRC_T` { `SYSCON_MAINCLKSRC_FRO12MHZ` = 0, `SYSCON_MAINCLKSRC_CLKIN`, `SYSCON_MAINCLKSRC_WDTOSC`, `SYSCON_MAINCLKSRC_FROHF`, `SYSCON_MAINCLKSRC_PLLOUT` = 6, `SYSCON_MAINCLKSRC_RTC` }
- enum `CHIP_SYSCON_FRGCLKSRC_T` { `SYSCON_FRGCLKSRC_MAINCLK`, `SYSCON_FRGCLKSRC_PLL`, `SYSCON_FRGCLKSRC_FRO12MHZ`, `SYSCON_FRGCLKSRC_FROHF`, `SYSCON_FRGCLKSRC_NONE` = 7 }

*Fractional Divider clock sources.*

## Functions

- `__STATIC_INLINE uint32_t Chip_Clock_GetExtClockInRate` (void)

- Returns the external clock input rate.*
- `__STATIC_INLINE uint32_t Chip_Clock_GetRTCOscRate (void)`
- Returns the RTC clock rate.*
- `__STATIC_INLINE void Chip_Clock_SetWDTOSCRate (WDT_OSC_FREQ_T freq, uint32_t div)`
- Set the WDT Oscillator frequency and divider.*
- `uint32_t Chip_Clock_GetWDTOSCRate (void)`
- Return estimated watchdog oscillator rate.*
- `__STATIC_INLINE uint32_t Chip_Clock_GetFROHFRate (void)`
- Gets the HF-FRO Frequency rate.*
- `__STATIC_INLINE void Chip_Clock_SetMain_A_ClockSource (CHIP_SYSCON_MAIN_A_CLKSRC_T src)`
- Set main A system clock source.*
- `__STATIC_INLINE void Chip_Clock_SetUSBClockSource (CHIP_SYSCON_USBCLKSRC_T src, uint32_t div)`
- Set USB clock source.*
- `__STATIC_INLINE CHIP_SYSCON_USBCLKSRC_T Chip_Clock_GetUSBClockSource (void)`
- Gets the clock source used by USB.*
- `__STATIC_INLINE uint32_t Chip_Clock_GetUSBClockDiv (void)`
- Gets the clock divider used by USB.*
- `__STATIC_INLINE void Chip_Clock_SetMCLKClockSource (CHIP_SYSCON_MCLKSRC_T src, uint32_t div)`
- Set the MCLK clock source.*
- `__STATIC_INLINE uint32_t Chip_Clock_GetMCLKDiv (void)`
- Get MCLK clock div.*
- `__STATIC_INLINE CHIP_SYSCON_MCLKSRC_T Chip_Clock_GetMCLKSource (void)`
- Get MCLK clock source.*
- `__STATIC_INLINE void Chip_Clock_SetMCLKDirInput (void)`
- Set MCLK pin direction to INPUT.*
- `__STATIC_INLINE void Chip_Clock_SetMCLKDirOutput (void)`
- Set MCLK pin direction to OUTPUT.*
- `__STATIC_INLINE void Chip_Clock_SetMCLKDir (int dir)`
- Set MCLK pin direction to INPUT or OUTPUT.*
- `__STATIC_INLINE int Chip_Clock_GetMCLKDir (void)`
- `__STATIC_INLINE CHIP_SYSCON_MAIN_A_CLKSRC_T Chip_Clock_GetMain_A_ClockSource (void)`
- Returns the main A clock source.*
- `uint32_t Chip_Clock_GetMain_A_ClockRate (void)`
- Return main A clock rate.*
- `__STATIC_INLINE void Chip_Clock_SetMain_B_ClockSource (CHIP_SYSCON_MAIN_B_CLKSRC_T src)`
- Set main B system clock source.*
- `__STATIC_INLINE CHIP_SYSCON_MAIN_B_CLKSRC_T Chip_Clock_GetMain_B_ClockSource (void)`
- Returns the main B clock source.*
- `uint32_t Chip_Clock_GetMain_B_ClockRate (void)`
- Return main B clock rate.*
- `__STATIC_INLINE void Chip_Clock_SetCLKOUTSource (CHIP_SYSCON_CLKOUTSRC_T src, uint32_t div)`
- Set CLKOUT clock source and divider.*
- `__STATIC_INLINE CHIP_SYSCON_CLKOUTSRC_T Chip_Clock_GetCLKOUTSource (void)`
- Get CLKOUT clock source.*
- `__STATIC_INLINE uint32_t Chip_Clock_GetCLKOUTDiv (void)`
- Get CLKOUT clock divider.*
- `void Chip_Clock_EnablePeriphClock (CHIP_SYSCON_CLOCK_T clk)`
- Enable a system or peripheral clock.*

- void [Chip\\_Clock\\_DisablePeriphClock](#) (CHIP\_SYSCON\_CLOCK\_T clk)  
*Disable a system or peripheral clock.*
- \_\_STATIC\_INLINE void [Chip\\_Clock\\_SetSysTickClockDiv](#) (uint32\_t div)  
*Set system tick clock divider (external CLKIN as SYSTICK reference only)*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_Clock\\_GetSysTickClockDiv](#) (void)  
*Returns system tick clock divider.*
- uint32\_t [Chip\\_Clock\\_GetSysTickClockRate](#) (void)  
*Returns the system tick rate as used with the system tick divider.*
- \_\_STATIC\_INLINE void [Chip\\_Clock\\_SetSysClockDiv](#) (uint32\_t div)  
*Set system clock divider.*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_Clock\\_GetSysClockDiv](#) (void)  
*Get system clock divider.*
- \_\_STATIC\_INLINE void [Chip\\_Clock\\_SetADCClockDiv](#) (uint32\_t div)  
*Set system tick clock divider.*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_Clock\\_GetADCClockDiv](#) (void)  
*Returns ADC clock divider.*
- \_\_STATIC\_INLINE void [Chip\\_Clock\\_SetFLEXCOMMClockSource](#) (uint32\_t idx, CHIP\_SYSCON\_FLEXCOMMCLKSELSRC\_T src)  
*Set the FLEXCOMM clock source.*
- \_\_STATIC\_INLINE CHIP\_SYSCON\_FLEXCOMMCLKSELSRC\_T [Chip\\_Clock\\_GetFLEXCOMMClockSource](#) (uint32\_t idx)  
*Returns the FLEXCOMM clock source.*
- uint32\_t [Chip\\_Clock\\_GetFLEXCOMMClockRate](#) (uint32\_t id)  
*Return FlexCOMM clock rate.*
- \_\_STATIC\_INLINE void [Chip\\_Clock\\_SetADCClockSource](#) (CHIP\_SYSCON\_ADCCLKSELSRC\_T src)  
*Set the ADC clock source.*
- \_\_STATIC\_INLINE CHIP\_SYSCON\_ADCCLKSELSRC\_T [Chip\\_Clock\\_GetADCClockSource](#) (void)  
*Returns the ADC clock source.*
- uint32\_t [Chip\\_Clock\\_GetADCClockRate](#) (void)  
*Return ADC clock rate.*
- \_\_STATIC\_INLINE void [Chip\\_Clock\\_EnableRTCOsc](#) (void)  
*Enable the RTC 32KHz output.*
- \_\_STATIC\_INLINE void [Chip\\_Clock\\_DisableRTCOsc](#) (void)  
*Disable the RTC 32KHz output.*
- \_\_STATIC\_INLINE bool [Chip\\_Clock\\_GetRTCOsc](#) (void)
- \_\_STATIC\_INLINE void [Chip\\_Clock\\_SetAsyncSysconClockSource](#) (CHIP\_ASYNC\_SYSCON\_SRC\_T src)  
*Set asynchronous APB clock source.*
- \_\_STATIC\_INLINE CHIP\_ASYNC\_SYSCON\_SRC\_T [Chip\\_Clock\\_GetAsyncSysconClockSource](#) (void)  
*Get asynchronous APB clock source.*
- uint32\_t [Chip\\_Clock\\_GetAsyncSyscon\\_ClockRate](#) (void)  
*Return asynchronous APB clock rate.*
- \_\_STATIC\_INLINE void [Chip\\_Clock\\_SetMainClockSource](#) (CHIP\_SYSCON\_MAINCLKSRC\_T src)  
*Set main system clock source.*
- CHIP\_SYSCON\_MAINCLKSRC\_T [Chip\\_Clock\\_GetMainClockSource](#) (void)  
*Get main system clock source.*
- uint32\_t [Chip\\_Clock\\_GetMainClockRate](#) (void)  
*Return main clock rate.*
- uint32\_t [Chip\\_Clock\\_GetSystemClockRate](#) (void)  
*Return system clock rate.*
- uint32\_t [Chip\\_Clock\\_GetFRGInClockRate](#) (void)  
*Get the input clock frequency of FRG.*

- `__STATIC_INLINE void Chip_Clock_SetFRGClockSource (CHIP_SYSCON_FRGCLKSRC_T src)`  
*Set clock source used by FRG.*
- `__STATIC_INLINE CHIP_SYSCON_FRGCLKSRC_T Chip_Clock_GetFRGClockSource (void)`  
*Get clock source used by FRG.*
- `uint32_t Chip_Clock_GetFRGClockRate (void)`  
*Get Fraction Rate Generator (FRG) clock rate.*
- `uint32_t Chip_Clock_SetFRGClockRate (uint32_t rate)`  
*Set FRG rate to given rate.*

## 8.4 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/cmsis.h File Reference

```
#include "lpc_types.h"
```

## 8.5 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/cmsis\_5411x.h File Reference

### Macros

- `#define __CM4_REV 0x0001`
- `#define __MPU_PRESENT 1`
- `#define __NVIC_PRIO_BITS 3`
- `#define __Vendor_SysTickConfig 0`
- `#define __FPU_PRESENT 1`
- `#define TIMER0_IRQn CT32B0_IRQn`  
*interrupt Alias*
- `#define TIMER1_IRQn CT32B1_IRQn`
- `#define TIMER2_IRQn CT32B2_IRQn`
- `#define TIMER3_IRQn CT32B3_IRQn`
- `#define TIMER4_IRQn CT32B4_IRQn`
- `#define SCT_IRQn SCT0_IRQn`
- `#define ADC0_SEQA_IRQn ADC_SEQA_IRQn`
- `#define ADC0_SEQB_IRQn ADC_SEQB_IRQn`
- `#define ADC0_THCMP_IRQn ADC_THCMP_IRQn`
- `#define TIMER0_IRQHandler CT32B0_IRQHandler`  
*Interrupt handler Alias.*
- `#define TIMER1_IRQHandler CT32B1_IRQHandler`
- `#define TIMER2_IRQHandler CT32B2_IRQHandler`
- `#define TIMER3_IRQHandler CT32B3_IRQHandler`
- `#define TIMER4_IRQHandler CT32B4_IRQHandler`
- `#define SCT_IRQHandler SCT0_IRQHandler`
- `#define ADC0_SEQA_IRQHandler ADC_SEQA_IRQHandler`
- `#define ADC0_SEQB_IRQHandler ADC_SEQB_IRQHandler`
- `#define ADC0_THCMP_IRQHandler ADC_THCMP_IRQHandler`

## Enumerations

- enum `LPC5411X_IRQn_Type` {  
`Reset_IRQn` = -15, `NonMaskableInt_IRQn` = -14, `HardFault_IRQn` = -13, `MemoryManagement_IRQn` = -12,  
`BusFault_IRQn` = -11, `UsageFault_IRQn` = -10, `SVCALL_IRQn` = -5, `DebugMonitor_IRQn` = -4,  
`PendSV_IRQn` = -2, `SysTick_IRQn` = -1, `WDTBOD_IRQn`, `DMA_IRQn`,  
`GINT0_IRQn`, `GINT1_IRQn`, `PIN_INT0_IRQn`, `PIN_INT1_IRQn`,  
`PIN_INT2_IRQn`, `PIN_INT3_IRQn`, `UTICK_IRQn`, `MRT_IRQn`,  
`CT32B0_IRQn`, `CT32B1_IRQn`, `SCT0_IRQn`, `CT32B3_IRQn`,  
`FLEXCOMM0_IRQn`, `FLEXCOMM1_IRQn`, `FLEXCOMM2_IRQn`, `FLEXCOMM3_IRQn`,  
`FLEXCOMM4_IRQn`, `FLEXCOMM5_IRQn`, `FLEXCOMM6_IRQn`, `FLEXCOMM7_IRQn`,  
`ADC_SEQA_IRQn`, `ADC_SEQB_IRQn`, `ADC_THCMP_IRQn`, `DMIC_IRQn`,  
`HWVAD_IRQn`, `USBACT_IRQn`, `USB_IRQn`, `RTC_IRQn`,  
`Reserved_IRQn`, `MAILBOX_IRQn`, `PIN_INT4_IRQn`, `PIN_INT5_IRQn`,  
`PIN_INT6_IRQn`, `PIN_INT7_IRQn`, `CT32B2_IRQn`, `CT32B4_IRQn`,  
`Reserved1_IRQn`, `SPIFI_IRQn` }

## 8.6 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/cmsis\_5411x\_m0.h File Reference

```
#include "lpc_types.h"
```

## Macros

- #define `__CM0PLUS_REV` 0x0001
- #define `__MPU_PRESENT` 0
- #define `__NVIC_PRIO_BITS` 2
- #define `__Vendor_SysTickConfig` 0
- #define `__VTOR_PRESENT` 1
- #define `TIMER0_IRQn` `CT32B0_IRQn`  
*interrupt Alias*
- #define `TIMER1_IRQn` `CT32B1_IRQn`
- #define `TIMER2_IRQn` `CT32B2_IRQn`
- #define `TIMER3_IRQn` `CT32B3_IRQn`
- #define `TIMER4_IRQn` `CT32B4_IRQn`
- #define `SCT_IRQn` `SCT0_IRQn`
- #define `ADC0_SEQA_IRQn` `ADC_SEQA_IRQn`
- #define `ADC0_SEQB_IRQn` `ADC_SEQB_IRQn`
- #define `ADC0_THCMP_IRQn` `ADC_THCMP_IRQn`
- #define `TIMER0_IRQHandler` `CT32B0_IRQHandler`  
*Interrupt handler Alias.*
- #define `TIMER1_IRQHandler` `CT32B1_IRQHandler`
- #define `TIMER2_IRQHandler` `CT32B2_IRQHandler`
- #define `TIMER3_IRQHandler` `CT32B3_IRQHandler`
- #define `TIMER4_IRQHandler` `CT32B4_IRQHandler`
- #define `SCT_IRQHandler` `SCT0_IRQHandler`
- #define `ADC0_SEQA_IRQHandler` `ADC_SEQA_IRQHandler`
- #define `ADC0_SEQB_IRQHandler` `ADC_SEQB_IRQHandler`
- #define `ADC0_THCMP_IRQHandler` `ADC_THCMP_IRQHandler`

## Enumerations

- enum [LPC5411X\\_M0\\_IRQn\\_Type](#) {  
[Reset\\_IRQn](#) = -15, [NonMaskableInt\\_IRQn](#) = -14, [HardFault\\_IRQn](#) = -13, [SVCall\\_IRQn](#) = -5,  
[PendSV\\_IRQn](#) = -2, [SysTick\\_IRQn](#) = -1, [WDTBOD\\_IRQn](#), [DMA\\_IRQn](#),  
[GINT0\\_IRQn](#), [GINT1\\_IRQn](#), [PIN\\_INT0\\_IRQn](#), [PIN\\_INT1\\_IRQn](#),  
[PIN\\_INT2\\_IRQn](#), [PIN\\_INT3\\_IRQn](#), [UTICK\\_IRQn](#), [MRT\\_IRQn](#),  
[CT32B0\\_IRQn](#), [CT32B1\\_IRQn](#), [SCT0\\_IRQn](#), [CT32B3\\_IRQn](#),  
[FLEXCOMM0\\_IRQn](#), [FLEXCOMM1\\_IRQn](#), [FLEXCOMM2\\_IRQn](#), [FLEXCOMM3\\_IRQn](#),  
[FLEXCOMM4\\_IRQn](#), [FLEXCOMM5\\_IRQn](#), [FLEXCOMM6\\_IRQn](#), [FLEXCOMM7\\_IRQn](#),  
[ADC\\_SEQA\\_IRQn](#), [ADC\\_SEQB\\_IRQn](#), [ADC\\_THCMP\\_IRQn](#), [DMIC\\_IRQn](#),  
[HWVAD](#), [USBACT\\_IRQn](#), [USB\\_IRQn](#), [RTC\\_IRQn](#),  
[Reserved\\_IRQn](#), [MAILBOX\\_IRQn](#) }

## 8.7 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm0plus.h File Reference

### 8.7.1 Detailed Description

CMSIS Cortex-M0+ Core Peripheral Access Layer Header File.

#### Version

V3.20

#### Date

25. February 2013

#### Note

```
#include <stdint.h>
#include <core_cmInstr.h>
#include <core_cmFunc.h>
```

## Data Structures

- union [APSR\\_Type](#)  
*Union type to access the Application Program Status Register (APSR).*
- union [IPSR\\_Type](#)  
*Union type to access the Interrupt Program Status Register (IPSR).*
- union [xPSR\\_Type](#)  
*Union type to access the Special-Purpose Program Status Registers (xPSR).*
- union [CONTROL\\_Type](#)  
*Union type to access the Control Registers (CONTROL).*
- struct [NVIC\\_Type](#)  
*Structure type to access the Nested Vectored Interrupt Controller (NVIC).*
- struct [SCB\\_Type](#)  
*Structure type to access the System Control Block (SCB).*
- struct [SysTick\\_Type](#)  
*Structure type to access the System Timer (SysTick).*

## Macros

- `#define __CORE_CM0PLUS_H_GENERIC`
- `#define SCB_CPUID_IMPLEMENTER_Pos 24`
- `#define SCB_CPUID_IMPLEMENTER_Msk (0xFFUL << SCB_CPUID_IMPLEMENTER_Pos)`
- `#define SCB_CPUID_VARIANT_Pos 20`
- `#define SCB_CPUID_VARIANT_Msk (0xFUL << SCB_CPUID_VARIANT_Pos)`
- `#define SCB_CPUID_ARCHITECTURE_Pos 16`
- `#define SCB_CPUID_ARCHITECTURE_Msk (0xFUL << SCB_CPUID_ARCHITECTURE_Pos)`
- `#define SCB_CPUID_PARTNO_Pos 4`
- `#define SCB_CPUID_PARTNO_Msk (0xFFFUL << SCB_CPUID_PARTNO_Pos)`
- `#define SCB_CPUID_REVISION_Pos 0`
- `#define SCB_CPUID_REVISION_Msk (0xFUL << SCB_CPUID_REVISION_Pos)`
- `#define SCB_ICSR_NMIPENDSET_Pos 31`
- `#define SCB_ICSR_NMIPENDSET_Msk (1UL << SCB_ICSR_NMIPENDSET_Pos)`
- `#define SCB_ICSR_PENDSVSET_Pos 28`
- `#define SCB_ICSR_PENDSVSET_Msk (1UL << SCB_ICSR_PENDSVSET_Pos)`
- `#define SCB_ICSR_PENDSVCLR_Pos 27`
- `#define SCB_ICSR_PENDSVCLR_Msk (1UL << SCB_ICSR_PENDSVCLR_Pos)`
- `#define SCB_ICSR_PENDSTSET_Pos 26`
- `#define SCB_ICSR_PENDSTSET_Msk (1UL << SCB_ICSR_PENDSTSET_Pos)`
- `#define SCB_ICSR_PENDSTCLR_Pos 25`
- `#define SCB_ICSR_PENDSTCLR_Msk (1UL << SCB_ICSR_PENDSTCLR_Pos)`
- `#define SCB_ICSR_ISRPREEMPT_Pos 23`
- `#define SCB_ICSR_ISRPREEMPT_Msk (1UL << SCB_ICSR_ISRPREEMPT_Pos)`
- `#define SCB_ICSR_ISRPENDING_Pos 22`
- `#define SCB_ICSR_ISRPENDING_Msk (1UL << SCB_ICSR_ISRPENDING_Pos)`
- `#define SCB_ICSR_VECTPENDING_Pos 12`
- `#define SCB_ICSR_VECTPENDING_Msk (0x1FFUL << SCB_ICSR_VECTPENDING_Pos)`
- `#define SCB_ICSR_VECTACTIVE_Pos 0`
- `#define SCB_ICSR_VECTACTIVE_Msk (0x1FFUL << SCB_ICSR_VECTACTIVE_Pos)`
- `#define SCB_AIRCR_VECTKEY_Pos 16`
- `#define SCB_AIRCR_VECTKEY_Msk (0xFFFFUL << SCB_AIRCR_VECTKEY_Pos)`
- `#define SCB_AIRCR_VECTKEYSTAT_Pos 16`
- `#define SCB_AIRCR_VECTKEYSTAT_Msk (0xFFFFUL << SCB_AIRCR_VECTKEYSTAT_Pos)`
- `#define SCB_AIRCR_ENDIANESS_Pos 15`
- `#define SCB_AIRCR_ENDIANESS_Msk (1UL << SCB_AIRCR_ENDIANESS_Pos)`
- `#define SCB_AIRCR_SYSRESETREQ_Pos 2`
- `#define SCB_AIRCR_SYSRESETREQ_Msk (1UL << SCB_AIRCR_SYSRESETREQ_Pos)`
- `#define SCB_AIRCR_VECTCLRACTIVE_Pos 1`
- `#define SCB_AIRCR_VECTCLRACTIVE_Msk (1UL << SCB_AIRCR_VECTCLRACTIVE_Pos)`
- `#define SCB_SCR_SEVONPEND_Pos 4`
- `#define SCB_SCR_SEVONPEND_Msk (1UL << SCB_SCR_SEVONPEND_Pos)`
- `#define SCB_SCR_SLEEPDEEP_Pos 2`
- `#define SCB_SCR_SLEEPDEEP_Msk (1UL << SCB_SCR_SLEEPDEEP_Pos)`
- `#define SCB_SCR_SLEEPONEXIT_Pos 1`
- `#define SCB_SCR_SLEEPONEXIT_Msk (1UL << SCB_SCR_SLEEPONEXIT_Pos)`
- `#define SCB_CCR_STKALIGN_Pos 9`
- `#define SCB_CCR_STKALIGN_Msk (1UL << SCB_CCR_STKALIGN_Pos)`
- `#define SCB_CCR_UNALIGN_TRP_Pos 3`
- `#define SCB_CCR_UNALIGN_TRP_Msk (1UL << SCB_CCR_UNALIGN_TRP_Pos)`
- `#define SCB_SHCSR_SVCALLPENDED_Pos 15`
- `#define SCB_SHCSR_SVCALLPENDED_Msk (1UL << SCB_SHCSR_SVCALLPENDED_Pos)`
- `#define SysTick_CTRL_COUNTFLAG_Pos 16`
- `#define SysTick_CTRL_COUNTFLAG_Msk (1UL << SysTick_CTRL_COUNTFLAG_Pos)`



- #define [SysTick\\_CTRL\\_CLKSOURCE\\_Pos](#) 2
- #define [SysTick\\_CTRL\\_CLKSOURCE\\_Msk](#) (1UL << SysTick\_CTRL\_CLKSOURCE\_Pos)
- #define [SysTick\\_CTRL\\_TICKINT\\_Pos](#) 1
- #define [SysTick\\_CTRL\\_TICKINT\\_Msk](#) (1UL << SysTick\_CTRL\_TICKINT\_Pos)
- #define [SysTick\\_CTRL\\_ENABLE\\_Pos](#) 0
- #define [SysTick\\_CTRL\\_ENABLE\\_Msk](#) (1UL << SysTick\_CTRL\_ENABLE\_Pos)
- #define [SysTick\\_LOAD\\_RELOAD\\_Pos](#) 0
- #define [SysTick\\_LOAD\\_RELOAD\\_Msk](#) (0xFFFFFUL << SysTick\_LOAD\_RELOAD\_Pos)
- #define [SysTick\\_VAL\\_CURRENT\\_Pos](#) 0
- #define [SysTick\\_VAL\\_CURRENT\\_Msk](#) (0xFFFFFUL << SysTick\_VAL\_CURRENT\_Pos)
- #define [SysTick\\_CALIB\\_NOREF\\_Pos](#) 31
- #define [SysTick\\_CALIB\\_NOREF\\_Msk](#) (1UL << SysTick\_CALIB\_NOREF\_Pos)
- #define [SysTick\\_CALIB\\_SKEW\\_Pos](#) 30
- #define [SysTick\\_CALIB\\_SKEW\\_Msk](#) (1UL << SysTick\_CALIB\_SKEW\_Pos)
- #define [SysTick\\_CALIB\\_TENMS\\_Pos](#) 0
- #define [SysTick\\_CALIB\\_TENMS\\_Msk](#) (0xFFFFFUL << SysTick\_VAL\_CURRENT\_Pos)
- #define [SCS\\_BASE](#) (0xE000E000UL)
- #define [SysTick\\_BASE](#) ([SCS\\_BASE](#) + 0x0010UL)
- #define [NVIC\\_BASE](#) ([SCS\\_BASE](#) + 0x0100UL)
- #define [SCB\\_BASE](#) ([SCS\\_BASE](#) + 0x0D00UL)
- #define [SCB](#) (([SCB\\_Type](#) \*) [SCB\\_BASE](#) )
- #define [SysTick](#) (([SysTick\\_Type](#) \*) [SysTick\\_BASE](#) )
- #define [NVIC](#) (([NVIC\\_Type](#) \*) [NVIC\\_BASE](#) )
- #define [\\_BIT\\_SHIFT](#)(IRQn) ( (((uint32\_t)(IRQn) ) & 0x03) \* 8 )
- #define [\\_SHP\\_IDX](#)(IRQn) ( (((uint32\_t)(IRQn) & 0x0F)-8) >> 2 )
- #define [\\_IP\\_IDX](#)(IRQn) ( ((uint32\_t)(IRQn) >> 2 )
  
- #define [\\_\\_CM0PLUS\\_CMSIS\\_VERSION\\_MAIN](#) (0x03)
- #define [\\_\\_CM0PLUS\\_CMSIS\\_VERSION\\_SUB](#) (0x20)
- #define [\\_\\_CM0PLUS\\_CMSIS\\_VERSION](#)
- #define [\\_\\_CORTEX\\_M](#) (0x00)
- #define [\\_\\_FPU\\_USED](#) 0
- #define [\\_\\_CORE\\_CM0PLUS\\_H\\_DEPENDANT](#)
- #define [\\_\\_I](#) volatile const
- #define [\\_\\_O](#) volatile
- #define [\\_\\_IO](#) volatile

## Functions

- [\\_\\_STATIC\\_INLINE void NVIC\\_EnableIRQ](#) (IRQn\_Type IRQn)  
*Enable External Interrupt.*
- [\\_\\_STATIC\\_INLINE void NVIC\\_DisableIRQ](#) (IRQn\_Type IRQn)  
*Disable External Interrupt.*
- [\\_\\_STATIC\\_INLINE uint32\\_t NVIC\\_GetPendingIRQ](#) (IRQn\_Type IRQn)  
*Get Pending Interrupt.*
- [\\_\\_STATIC\\_INLINE void NVIC\\_SetPendingIRQ](#) (IRQn\_Type IRQn)  
*Set Pending Interrupt.*
- [\\_\\_STATIC\\_INLINE void NVIC\\_ClearPendingIRQ](#) (IRQn\_Type IRQn)  
*Clear Pending Interrupt.*
- [\\_\\_STATIC\\_INLINE void NVIC\\_SetPriority](#) (IRQn\_Type IRQn, uint32\_t priority)  
*Set Interrupt Priority.*
- [\\_\\_STATIC\\_INLINE uint32\\_t NVIC\\_GetPriority](#) (IRQn\_Type IRQn)  
*Get Interrupt Priority.*

- `__STATIC_INLINE void NVIC_SystemReset (void)`  
*System Reset.*
- `__STATIC_INLINE uint32_t SysTick_Config (uint32_t ticks)`  
*System Tick Configuration.*

## 8.7.2 Macro Definition Documentation

### 8.7.2.1 `#define __CM0PLUS_CMSIS_VERSION`

#### Value:

```
((__CM0PLUS_CMSIS_VERSION_MAIN << 16) | \
    __CM0PLUS_CMSIS_VERSION_SUB)
```

CMSIS HAL version number

Definition at line 73 of file `core_cm0plus.h`.

### 8.7.2.2 `#define __CM0PLUS_CMSIS_VERSION_MAIN (0x03)`

+ [31:16] CMSIS HAL main version

Definition at line 71 of file `core_cm0plus.h`.

### 8.7.2.3 `#define __CM0PLUS_CMSIS_VERSION_SUB (0x20)`

[15:0] CMSIS HAL sub version

Definition at line 72 of file `core_cm0plus.h`.

### 8.7.2.4 `#define __CORE_CM0PLUS_H_DEPENDANT`

Definition at line 135 of file `core_cm0plus.h`.

### 8.7.2.5 `#define __CORE_CM0PLUS_H_GENERIC`

Definition at line 47 of file `core_cm0plus.h`.

### 8.7.2.6 `#define __CORTEX_M (0x00)`

Cortex-M Core

Definition at line 76 of file `core_cm0plus.h`.

### 8.7.2.7 `#define __FPU_USED 0`

`__FPU_USED` indicates whether an FPU is used or not. This core does not support an FPU at all

Definition at line 103 of file `core_cm0plus.h`.

### 8.7.2.8 `#define __I volatile const`

Defines 'read only' permissions

Definition at line 176 of file `core_cm0plus.h`.

#### 8.7.2.9 #define \_\_IO volatile

Defines 'read / write' permissions

Definition at line 179 of file core\_cm0plus.h.

#### 8.7.2.10 #define \_\_O volatile

Defines 'write only' permissions

Definition at line 178 of file core\_cm0plus.h.

## 8.8 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm4.h File Reference

### 8.8.1 Detailed Description

CMSIS Cortex-M4 Core Peripheral Access Layer Header File.

#### Version

V3.20

#### Date

25. February 2013

#### Note

```
#include <stdint.h>
#include <core_cmInstr.h>
#include <core_cmFunc.h>
#include <core_cm4_simd.h>
```

### Data Structures

- union [APSR\\_Type](#)  
*Union type to access the Application Program Status Register (APSR).*
- union [IPSR\\_Type](#)  
*Union type to access the Interrupt Program Status Register (IPSR).*
- union [xPSR\\_Type](#)  
*Union type to access the Special-Purpose Program Status Registers (xPSR).*
- union [CONTROL\\_Type](#)  
*Union type to access the Control Registers (CONTROL).*
- struct [NVIC\\_Type](#)  
*Structure type to access the Nested Vectored Interrupt Controller (NVIC).*
- struct [SCB\\_Type](#)  
*Structure type to access the System Control Block (SCB).*
- struct [SCnSCB\\_Type](#)  
*Structure type to access the System Control and ID Register not in the SCB.*
- struct [SysTick\\_Type](#)

- *Structure type to access the System Timer (SysTick).*
- struct [ITM\\_Type](#)
  - *Structure type to access the Instrumentation Trace Macrocell Register (ITM).*
- struct [DWT\\_Type](#)
  - *Structure type to access the Data Watchpoint and Trace Register (DWT).*
- struct [TPI\\_Type](#)
  - *Structure type to access the Trace Port Interface Register (TPI).*
- struct [CoreDebug\\_Type](#)
  - *Structure type to access the Core Debug Register (CoreDebug).*

## Macros

- #define [\\_\\_CORE\\_CM4\\_H\\_GENERIC](#)
- #define [NVIC\\_STIR\\_INTID\\_Pos](#) 0
- #define [NVIC\\_STIR\\_INTID\\_Msk](#) (0x1FFUL << NVIC\_STIR\_INTID\_Pos)
- #define [SCB\\_CPUID\\_IMPLEMENTER\\_Pos](#) 24
- #define [SCB\\_CPUID\\_IMPLEMENTER\\_Msk](#) (0xFFUL << SCB\_CPUID\_IMPLEMENTER\_Pos)
- #define [SCB\\_CPUID\\_VARIANT\\_Pos](#) 20
- #define [SCB\\_CPUID\\_VARIANT\\_Msk](#) (0xFUL << SCB\_CPUID\_VARIANT\_Pos)
- #define [SCB\\_CPUID\\_ARCHITECTURE\\_Pos](#) 16
- #define [SCB\\_CPUID\\_ARCHITECTURE\\_Msk](#) (0xFUL << SCB\_CPUID\_ARCHITECTURE\_Pos)
- #define [SCB\\_CPUID\\_PARTNO\\_Pos](#) 4
- #define [SCB\\_CPUID\\_PARTNO\\_Msk](#) (0xFFFUL << SCB\_CPUID\_PARTNO\_Pos)
- #define [SCB\\_CPUID\\_REVISION\\_Pos](#) 0
- #define [SCB\\_CPUID\\_REVISION\\_Msk](#) (0xFUL << SCB\_CPUID\_REVISION\_Pos)
- #define [SCB\\_ICSR\\_NMIPENDSET\\_Pos](#) 31
- #define [SCB\\_ICSR\\_NMIPENDSET\\_Msk](#) (1UL << SCB\_ICSR\_NMIPENDSET\_Pos)
- #define [SCB\\_ICSR\\_PENDSVSET\\_Pos](#) 28
- #define [SCB\\_ICSR\\_PENDSVSET\\_Msk](#) (1UL << SCB\_ICSR\_PENDSVSET\_Pos)
- #define [SCB\\_ICSR\\_PENDSVCLR\\_Pos](#) 27
- #define [SCB\\_ICSR\\_PENDSVCLR\\_Msk](#) (1UL << SCB\_ICSR\_PENDSVCLR\_Pos)
- #define [SCB\\_ICSR\\_PENDSTSET\\_Pos](#) 26
- #define [SCB\\_ICSR\\_PENDSTSET\\_Msk](#) (1UL << SCB\_ICSR\_PENDSTSET\_Pos)
- #define [SCB\\_ICSR\\_PENDSTCLR\\_Pos](#) 25
- #define [SCB\\_ICSR\\_PENDSTCLR\\_Msk](#) (1UL << SCB\_ICSR\_PENDSTCLR\_Pos)
- #define [SCB\\_ICSR\\_ISRPREEMPT\\_Pos](#) 23
- #define [SCB\\_ICSR\\_ISRPREEMPT\\_Msk](#) (1UL << SCB\_ICSR\_ISRPREEMPT\_Pos)
- #define [SCB\\_ICSR\\_ISRPENDING\\_Pos](#) 22
- #define [SCB\\_ICSR\\_ISRPENDING\\_Msk](#) (1UL << SCB\_ICSR\_ISRPENDING\_Pos)
- #define [SCB\\_ICSR\\_VECTPENDING\\_Pos](#) 12
- #define [SCB\\_ICSR\\_VECTPENDING\\_Msk](#) (0x1FFUL << SCB\_ICSR\_VECTPENDING\_Pos)
- #define [SCB\\_ICSR\\_RETTOBASE\\_Pos](#) 11
- #define [SCB\\_ICSR\\_RETTOBASE\\_Msk](#) (1UL << SCB\_ICSR\_RETTOBASE\_Pos)
- #define [SCB\\_ICSR\\_VECTACTIVE\\_Pos](#) 0
- #define [SCB\\_ICSR\\_VECTACTIVE\\_Msk](#) (0x1FFUL << SCB\_ICSR\_VECTACTIVE\_Pos)
- #define [SCB\\_VTOR\\_TBLOFF\\_Pos](#) 7
- #define [SCB\\_VTOR\\_TBLOFF\\_Msk](#) (0x1FFFFFFUL << SCB\_VTOR\_TBLOFF\_Pos)
- #define [SCB\\_AIRCR\\_VECTKEY\\_Pos](#) 16
- #define [SCB\\_AIRCR\\_VECTKEY\\_Msk](#) (0xFFFFUL << SCB\_AIRCR\_VECTKEY\_Pos)
- #define [SCB\\_AIRCR\\_VECTKEYSTAT\\_Pos](#) 16
- #define [SCB\\_AIRCR\\_VECTKEYSTAT\\_Msk](#) (0xFFFFUL << SCB\_AIRCR\_VECTKEYSTAT\_Pos)
- #define [SCB\\_AIRCR\\_ENDIANESS\\_Pos](#) 15
- #define [SCB\\_AIRCR\\_ENDIANESS\\_Msk](#) (1UL << SCB\_AIRCR\_ENDIANESS\_Pos)

- #define [SCB\\_AICR\\_PRIGROUP\\_Pos](#) 8
- #define [SCB\\_AICR\\_PRIGROUP\\_Msk](#) (7UL << SCB\_AICR\_PRIGROUP\_Pos)
- #define [SCB\\_AICR\\_SYSRESETREQ\\_Pos](#) 2
- #define [SCB\\_AICR\\_SYSRESETREQ\\_Msk](#) (1UL << SCB\_AICR\_SYSRESETREQ\_Pos)
- #define [SCB\\_AICR\\_VECTCLRACTIVE\\_Pos](#) 1
- #define [SCB\\_AICR\\_VECTCLRACTIVE\\_Msk](#) (1UL << SCB\_AICR\_VECTCLRACTIVE\_Pos)
- #define [SCB\\_AICR\\_VECTRESET\\_Pos](#) 0
- #define [SCB\\_AICR\\_VECTRESET\\_Msk](#) (1UL << SCB\_AICR\_VECTRESET\_Pos)
- #define [SCB\\_SCR\\_SEVONPEND\\_Pos](#) 4
- #define [SCB\\_SCR\\_SEVONPEND\\_Msk](#) (1UL << SCB\_SCR\_SEVONPEND\_Pos)
- #define [SCB\\_SCR\\_SLEEPDEEP\\_Pos](#) 2
- #define [SCB\\_SCR\\_SLEEPDEEP\\_Msk](#) (1UL << SCB\_SCR\_SLEEPDEEP\_Pos)
- #define [SCB\\_SCR\\_SLEEPONEXIT\\_Pos](#) 1
- #define [SCB\\_SCR\\_SLEEPONEXIT\\_Msk](#) (1UL << SCB\_SCR\_SLEEPONEXIT\_Pos)
- #define [SCB\\_CCR\\_STKALIGN\\_Pos](#) 9
- #define [SCB\\_CCR\\_STKALIGN\\_Msk](#) (1UL << SCB\_CCR\_STKALIGN\_Pos)
- #define [SCB\\_CCR\\_BFHFMIGN\\_Pos](#) 8
- #define [SCB\\_CCR\\_BFHFMIGN\\_Msk](#) (1UL << SCB\_CCR\_BFHFMIGN\_Pos)
- #define [SCB\\_CCR\\_DIV\\_0\\_TRP\\_Pos](#) 4
- #define [SCB\\_CCR\\_DIV\\_0\\_TRP\\_Msk](#) (1UL << SCB\_CCR\_DIV\_0\_TRP\_Pos)
- #define [SCB\\_CCR\\_UNALIGN\\_TRP\\_Pos](#) 3
- #define [SCB\\_CCR\\_UNALIGN\\_TRP\\_Msk](#) (1UL << SCB\_CCR\_UNALIGN\_TRP\_Pos)
- #define [SCB\\_CCR\\_USERSETMPEND\\_Pos](#) 1
- #define [SCB\\_CCR\\_USERSETMPEND\\_Msk](#) (1UL << SCB\_CCR\_USERSETMPEND\_Pos)
- #define [SCB\\_CCR\\_NONBASETHRDENA\\_Pos](#) 0
- #define [SCB\\_CCR\\_NONBASETHRDENA\\_Msk](#) (1UL << SCB\_CCR\_NONBASETHRDENA\_Pos)
- #define [SCB\\_SHCSR\\_USGFAULTENA\\_Pos](#) 18
- #define [SCB\\_SHCSR\\_USGFAULTENA\\_Msk](#) (1UL << SCB\_SHCSR\_USGFAULTENA\_Pos)
- #define [SCB\\_SHCSR\\_BUSFAULTENA\\_Pos](#) 17
- #define [SCB\\_SHCSR\\_BUSFAULTENA\\_Msk](#) (1UL << SCB\_SHCSR\_BUSFAULTENA\_Pos)
- #define [SCB\\_SHCSR\\_MEMFAULTENA\\_Pos](#) 16
- #define [SCB\\_SHCSR\\_MEMFAULTENA\\_Msk](#) (1UL << SCB\_SHCSR\_MEMFAULTENA\_Pos)
- #define [SCB\\_SHCSR\\_SVCALLPENDE\\_Pos](#) 15
- #define [SCB\\_SHCSR\\_SVCALLPENDE\\_Msk](#) (1UL << SCB\_SHCSR\_SVCALLPENDE\_Pos)
- #define [SCB\\_SHCSR\\_BUSFAULTPENDE\\_Pos](#) 14
- #define [SCB\\_SHCSR\\_BUSFAULTPENDE\\_Msk](#) (1UL << SCB\_SHCSR\_BUSFAULTPENDE\_Pos)
- #define [SCB\\_SHCSR\\_MEMFAULTPENDE\\_Pos](#) 13
- #define [SCB\\_SHCSR\\_MEMFAULTPENDE\\_Msk](#) (1UL << SCB\_SHCSR\_MEMFAULTPENDE\_Pos)
- #define [SCB\\_SHCSR\\_USGFAULTPENDE\\_Pos](#) 12
- #define [SCB\\_SHCSR\\_USGFAULTPENDE\\_Msk](#) (1UL << SCB\_SHCSR\_USGFAULTPENDE\_Pos)
- #define [SCB\\_SHCSR\\_SYSTICKACT\\_Pos](#) 11
- #define [SCB\\_SHCSR\\_SYSTICKACT\\_Msk](#) (1UL << SCB\_SHCSR\_SYSTICKACT\_Pos)
- #define [SCB\\_SHCSR\\_PENDSVACT\\_Pos](#) 10
- #define [SCB\\_SHCSR\\_PENDSVACT\\_Msk](#) (1UL << SCB\_SHCSR\_PENDSVACT\_Pos)
- #define [SCB\\_SHCSR\\_MONITORACT\\_Pos](#) 8
- #define [SCB\\_SHCSR\\_MONITORACT\\_Msk](#) (1UL << SCB\_SHCSR\_MONITORACT\_Pos)
- #define [SCB\\_SHCSR\\_SVCALLACT\\_Pos](#) 7
- #define [SCB\\_SHCSR\\_SVCALLACT\\_Msk](#) (1UL << SCB\_SHCSR\_SVCALLACT\_Pos)
- #define [SCB\\_SHCSR\\_USGFAULTACT\\_Pos](#) 3
- #define [SCB\\_SHCSR\\_USGFAULTACT\\_Msk](#) (1UL << SCB\_SHCSR\_USGFAULTACT\_Pos)
- #define [SCB\\_SHCSR\\_BUSFAULTACT\\_Pos](#) 1
- #define [SCB\\_SHCSR\\_BUSFAULTACT\\_Msk](#) (1UL << SCB\_SHCSR\_BUSFAULTACT\_Pos)
- #define [SCB\\_SHCSR\\_MEMFAULTACT\\_Pos](#) 0
- #define [SCB\\_SHCSR\\_MEMFAULTACT\\_Msk](#) (1UL << SCB\_SHCSR\_MEMFAULTACT\_Pos)
- #define [SCB\\_CFSR\\_USGFAULTSR\\_Pos](#) 16

- #define SCB\_CFSR\_USGFAULTSR\_Msk (0xFFFFFUL << SCB\_CFSR\_USGFAULTSR\_Pos)
- #define SCB\_CFSR\_BUSFAULTSR\_Pos 8
- #define SCB\_CFSR\_BUSFAULTSR\_Msk (0xFFFUL << SCB\_CFSR\_BUSFAULTSR\_Pos)
- #define SCB\_CFSR\_MEMFAULTSR\_Pos 0
- #define SCB\_CFSR\_MEMFAULTSR\_Msk (0xFFFUL << SCB\_CFSR\_MEMFAULTSR\_Pos)
- #define SCB\_HFSR\_DEBUGEVT\_Pos 31
- #define SCB\_HFSR\_DEBUGEVT\_Msk (1UL << SCB\_HFSR\_DEBUGEVT\_Pos)
- #define SCB\_HFSR\_FORCED\_Pos 30
- #define SCB\_HFSR\_FORCED\_Msk (1UL << SCB\_HFSR\_FORCED\_Pos)
- #define SCB\_HFSR\_VECTTBL\_Pos 1
- #define SCB\_HFSR\_VECTTBL\_Msk (1UL << SCB\_HFSR\_VECTTBL\_Pos)
- #define SCB\_DFSR\_EXTERNAL\_Pos 4
- #define SCB\_DFSR\_EXTERNAL\_Msk (1UL << SCB\_DFSR\_EXTERNAL\_Pos)
- #define SCB\_DFSR\_VCATCH\_Pos 3
- #define SCB\_DFSR\_VCATCH\_Msk (1UL << SCB\_DFSR\_VCATCH\_Pos)
- #define SCB\_DFSR\_DWTTRAP\_Pos 2
- #define SCB\_DFSR\_DWTTRAP\_Msk (1UL << SCB\_DFSR\_DWTTRAP\_Pos)
- #define SCB\_DFSR\_BKPT\_Pos 1
- #define SCB\_DFSR\_BKPT\_Msk (1UL << SCB\_DFSR\_BKPT\_Pos)
- #define SCB\_DFSR\_HALTED\_Pos 0
- #define SCB\_DFSR\_HALTED\_Msk (1UL << SCB\_DFSR\_HALTED\_Pos)
- #define SCnSCB\_ICTR\_INTLINESNUM\_Pos 0
- #define SCnSCB\_ICTR\_INTLINESNUM\_Msk (0xFUL << SCnSCB\_ICTR\_INTLINESNUM\_Pos)
- #define SCnSCB\_ACTLR\_DISOOF\_Pos 9
- #define SCnSCB\_ACTLR\_DISOOF\_Msk (1UL << SCnSCB\_ACTLR\_DISOOF\_Pos)
- #define SCnSCB\_ACTLR\_DISFPCA\_Pos 8
- #define SCnSCB\_ACTLR\_DISFPCA\_Msk (1UL << SCnSCB\_ACTLR\_DISFPCA\_Pos)
- #define SCnSCB\_ACTLR\_DISFOLD\_Pos 2
- #define SCnSCB\_ACTLR\_DISFOLD\_Msk (1UL << SCnSCB\_ACTLR\_DISFOLD\_Pos)
- #define SCnSCB\_ACTLR\_DISDEFWBUF\_Pos 1
- #define SCnSCB\_ACTLR\_DISDEFWBUF\_Msk (1UL << SCnSCB\_ACTLR\_DISDEFWBUF\_Pos)
- #define SCnSCB\_ACTLR\_DISMCYCINT\_Pos 0
- #define SCnSCB\_ACTLR\_DISMCYCINT\_Msk (1UL << SCnSCB\_ACTLR\_DISMCYCINT\_Pos)
- #define SysTick\_CTRL\_COUNTFLAG\_Pos 16
- #define SysTick\_CTRL\_COUNTFLAG\_Msk (1UL << SysTick\_CTRL\_COUNTFLAG\_Pos)
- #define SysTick\_CTRL\_CLKSOURCE\_Pos 2
- #define SysTick\_CTRL\_CLKSOURCE\_Msk (1UL << SysTick\_CTRL\_CLKSOURCE\_Pos)
- #define SysTick\_CTRL\_TICKINT\_Pos 1
- #define SysTick\_CTRL\_TICKINT\_Msk (1UL << SysTick\_CTRL\_TICKINT\_Pos)
- #define SysTick\_CTRL\_ENABLE\_Pos 0
- #define SysTick\_CTRL\_ENABLE\_Msk (1UL << SysTick\_CTRL\_ENABLE\_Pos)
- #define SysTick\_LOAD\_RELOAD\_Pos 0
- #define SysTick\_LOAD\_RELOAD\_Msk (0xFFFFFUL << SysTick\_LOAD\_RELOAD\_Pos)
- #define SysTick\_VAL\_CURRENT\_Pos 0
- #define SysTick\_VAL\_CURRENT\_Msk (0xFFFFFUL << SysTick\_VAL\_CURRENT\_Pos)
- #define SysTick\_CALIB\_NOREF\_Pos 31
- #define SysTick\_CALIB\_NOREF\_Msk (1UL << SysTick\_CALIB\_NOREF\_Pos)
- #define SysTick\_CALIB\_SKEW\_Pos 30
- #define SysTick\_CALIB\_SKEW\_Msk (1UL << SysTick\_CALIB\_SKEW\_Pos)
- #define SysTick\_CALIB\_TENMS\_Pos 0
- #define SysTick\_CALIB\_TENMS\_Msk (0xFFFFFUL << SysTick\_VAL\_CURRENT\_Pos)
- #define ITM\_TPR\_PRIVMASK\_Pos 0
- #define ITM\_TPR\_PRIVMASK\_Msk (0xFUL << ITM\_TPR\_PRIVMASK\_Pos)
- #define ITM\_TCR\_BUSY\_Pos 23
- #define ITM\_TCR\_BUSY\_Msk (1UL << ITM\_TCR\_BUSY\_Pos)

- `#define ITM_TCR_TraceBusID_Pos` 16
- `#define ITM_TCR_TraceBusID_Msk` (0x7FUL << ITM\_TCR\_TraceBusID\_Pos)
- `#define ITM_TCR_GTSFREQ_Pos` 10
- `#define ITM_TCR_GTSFREQ_Msk` (3UL << ITM\_TCR\_GTSFREQ\_Pos)
- `#define ITM_TCR_TSPrescale_Pos` 8
- `#define ITM_TCR_TSPrescale_Msk` (3UL << ITM\_TCR\_TSPrescale\_Pos)
- `#define ITM_TCR_SWOENA_Pos` 4
- `#define ITM_TCR_SWOENA_Msk` (1UL << ITM\_TCR\_SWOENA\_Pos)
- `#define ITM_TCR_DWTENA_Pos` 3
- `#define ITM_TCR_DWTENA_Msk` (1UL << ITM\_TCR\_DWTENA\_Pos)
- `#define ITM_TCR_SYNCENA_Pos` 2
- `#define ITM_TCR_SYNCENA_Msk` (1UL << ITM\_TCR\_SYNCENA\_Pos)
- `#define ITM_TCR_TSENA_Pos` 1
- `#define ITM_TCR_TSENA_Msk` (1UL << ITM\_TCR\_TSENA\_Pos)
- `#define ITM_TCR_ITMENA_Pos` 0
- `#define ITM_TCR_ITMENA_Msk` (1UL << ITM\_TCR\_ITMENA\_Pos)
- `#define ITM_IWR_ATVALIDM_Pos` 0
- `#define ITM_IWR_ATVALIDM_Msk` (1UL << ITM\_IWR\_ATVALIDM\_Pos)
- `#define ITM_IRR_ATREADYM_Pos` 0
- `#define ITM_IRR_ATREADYM_Msk` (1UL << ITM\_IRR\_ATREADYM\_Pos)
- `#define ITM_IMCR_INTEGRATION_Pos` 0
- `#define ITM_IMCR_INTEGRATION_Msk` (1UL << ITM\_IMCR\_INTEGRATION\_Pos)
- `#define ITM_LSR_ByteAcc_Pos` 2
- `#define ITM_LSR_ByteAcc_Msk` (1UL << ITM\_LSR\_ByteAcc\_Pos)
- `#define ITM_LSR_Access_Pos` 1
- `#define ITM_LSR_Access_Msk` (1UL << ITM\_LSR\_Access\_Pos)
- `#define ITM_LSR_Present_Pos` 0
- `#define ITM_LSR_Present_Msk` (1UL << ITM\_LSR\_Present\_Pos)
- `#define DWT_CTRL_NUMCOMP_Pos` 28
- `#define DWT_CTRL_NUMCOMP_Msk` (0xFUL << DWT\_CTRL\_NUMCOMP\_Pos)
- `#define DWT_CTRL_NOTRCPKT_Pos` 27
- `#define DWT_CTRL_NOTRCPKT_Msk` (0x1UL << DWT\_CTRL\_NOTRCPKT\_Pos)
- `#define DWT_CTRL_NOEXTTRIG_Pos` 26
- `#define DWT_CTRL_NOEXTTRIG_Msk` (0x1UL << DWT\_CTRL\_NOEXTTRIG\_Pos)
- `#define DWT_CTRL_NOCYCCNT_Pos` 25
- `#define DWT_CTRL_NOCYCCNT_Msk` (0x1UL << DWT\_CTRL\_NOCYCCNT\_Pos)
- `#define DWT_CTRL_NOPRFCNT_Pos` 24
- `#define DWT_CTRL_NOPRFCNT_Msk` (0x1UL << DWT\_CTRL\_NOPRFCNT\_Pos)
- `#define DWT_CTRL_CYCEVTENA_Pos` 22
- `#define DWT_CTRL_CYCEVTENA_Msk` (0x1UL << DWT\_CTRL\_CYCEVTENA\_Pos)
- `#define DWT_CTRL_FOLDEVTENA_Pos` 21
- `#define DWT_CTRL_FOLDEVTENA_Msk` (0x1UL << DWT\_CTRL\_FOLDEVTENA\_Pos)
- `#define DWT_CTRL_LSUEVTENA_Pos` 20
- `#define DWT_CTRL_LSUEVTENA_Msk` (0x1UL << DWT\_CTRL\_LSUEVTENA\_Pos)
- `#define DWT_CTRL_SLEEPEVTENA_Pos` 19
- `#define DWT_CTRL_SLEEPEVTENA_Msk` (0x1UL << DWT\_CTRL\_SLEEPEVTENA\_Pos)
- `#define DWT_CTRL_EXCEVTENA_Pos` 18
- `#define DWT_CTRL_EXCEVTENA_Msk` (0x1UL << DWT\_CTRL\_EXCEVTENA\_Pos)
- `#define DWT_CTRL_CPIEVTENA_Pos` 17
- `#define DWT_CTRL_CPIEVTENA_Msk` (0x1UL << DWT\_CTRL\_CPIEVTENA\_Pos)
- `#define DWT_CTRL_EXCTRCENA_Pos` 16
- `#define DWT_CTRL_EXCTRCENA_Msk` (0x1UL << DWT\_CTRL\_EXCTRCENA\_Pos)
- `#define DWT_CTRL_PCSAMPLENA_Pos` 12
- `#define DWT_CTRL_PCSAMPLENA_Msk` (0x1UL << DWT\_CTRL\_PCSAMPLENA\_Pos)
- `#define DWT_CTRL_SYNCTAP_Pos` 10



- `#define DWT_CTRL_SYNCTAP_Msk` (0x3UL << DWT\_CTRL\_SYNCTAP\_Pos)
- `#define DWT_CTRL_CYCTAP_Pos` 9
- `#define DWT_CTRL_CYCTAP_Msk` (0x1UL << DWT\_CTRL\_CYCTAP\_Pos)
- `#define DWT_CTRL_POSTINIT_Pos` 5
- `#define DWT_CTRL_POSTINIT_Msk` (0xFUL << DWT\_CTRL\_POSTINIT\_Pos)
- `#define DWT_CTRL_POSTPRESET_Pos` 1
- `#define DWT_CTRL_POSTPRESET_Msk` (0xFUL << DWT\_CTRL\_POSTPRESET\_Pos)
- `#define DWT_CTRL_CYCCNTENA_Pos` 0
- `#define DWT_CTRL_CYCCNTENA_Msk` (0x1UL << DWT\_CTRL\_CYCCNTENA\_Pos)
- `#define DWT_CPICNT_CPICNT_Pos` 0
- `#define DWT_CPICNT_CPICNT_Msk` (0xFFUL << DWT\_CPICNT\_CPICNT\_Pos)
- `#define DWT_EXCCNT_EXCCNT_Pos` 0
- `#define DWT_EXCCNT_EXCCNT_Msk` (0xFFUL << DWT\_EXCCNT\_EXCCNT\_Pos)
- `#define DWT_SLEEPCNT_SLEEPCNT_Pos` 0
- `#define DWT_SLEEPCNT_SLEEPCNT_Msk` (0xFFUL << DWT\_SLEEPCNT\_SLEEPCNT\_Pos)
- `#define DWT_LSUCNT_LSUCNT_Pos` 0
- `#define DWT_LSUCNT_LSUCNT_Msk` (0xFFUL << DWT\_LSUCNT\_LSUCNT\_Pos)
- `#define DWT_FOLDCNT_FOLDCNT_Pos` 0
- `#define DWT_FOLDCNT_FOLDCNT_Msk` (0xFFUL << DWT\_FOLDCNT\_FOLDCNT\_Pos)
- `#define DWT_MASK_MASK_Pos` 0
- `#define DWT_MASK_MASK_Msk` (0x1FUL << DWT\_MASK\_MASK\_Pos)
- `#define DWT_FUNCTION_MATCHED_Pos` 24
- `#define DWT_FUNCTION_MATCHED_Msk` (0x1UL << DWT\_FUNCTION\_MATCHED\_Pos)
- `#define DWT_FUNCTION_DATAVADDR1_Pos` 16
- `#define DWT_FUNCTION_DATAVADDR1_Msk` (0xFUL << DWT\_FUNCTION\_DATAVADDR1\_Pos)
- `#define DWT_FUNCTION_DATAVADDR0_Pos` 12
- `#define DWT_FUNCTION_DATAVADDR0_Msk` (0xFUL << DWT\_FUNCTION\_DATAVADDR0\_Pos)
- `#define DWT_FUNCTION_DATAVSIZE_Pos` 10
- `#define DWT_FUNCTION_DATAVSIZE_Msk` (0x3UL << DWT\_FUNCTION\_DATAVSIZE\_Pos)
- `#define DWT_FUNCTION_LNK1ENA_Pos` 9
- `#define DWT_FUNCTION_LNK1ENA_Msk` (0x1UL << DWT\_FUNCTION\_LNK1ENA\_Pos)
- `#define DWT_FUNCTION_DATAVMATCH_Pos` 8
- `#define DWT_FUNCTION_DATAVMATCH_Msk` (0x1UL << DWT\_FUNCTION\_DATAVMATCH\_Pos)
- `#define DWT_FUNCTION_CYCMATCH_Pos` 7
- `#define DWT_FUNCTION_CYCMATCH_Msk` (0x1UL << DWT\_FUNCTION\_CYCMATCH\_Pos)
- `#define DWT_FUNCTION_EMITRANGE_Pos` 5
- `#define DWT_FUNCTION_EMITRANGE_Msk` (0x1UL << DWT\_FUNCTION\_EMITRANGE\_Pos)
- `#define DWT_FUNCTION_FUNCTION_Pos` 0
- `#define DWT_FUNCTION_FUNCTION_Msk` (0xFUL << DWT\_FUNCTION\_FUNCTION\_Pos)
- `#define TPI_ACPR_PRESCALER_Pos` 0
- `#define TPI_ACPR_PRESCALER_Msk` (0x1FFFUL << TPI\_ACPR\_PRESCALER\_Pos)
- `#define TPI_SPPR_TXMODE_Pos` 0
- `#define TPI_SPPR_TXMODE_Msk` (0x3UL << TPI\_SPPR\_TXMODE\_Pos)
- `#define TPI_FFSR_FtNonStop_Pos` 3
- `#define TPI_FFSR_FtNonStop_Msk` (0x1UL << TPI\_FFSR\_FtNonStop\_Pos)
- `#define TPI_FFSR_TCPresent_Pos` 2
- `#define TPI_FFSR_TCPresent_Msk` (0x1UL << TPI\_FFSR\_TCPresent\_Pos)
- `#define TPI_FFSR_FtStopped_Pos` 1
- `#define TPI_FFSR_FtStopped_Msk` (0x1UL << TPI\_FFSR\_FtStopped\_Pos)
- `#define TPI_FFSR_FlInProg_Pos` 0
- `#define TPI_FFSR_FlInProg_Msk` (0x1UL << TPI\_FFSR\_FlInProg\_Pos)
- `#define TPI_FFCR_TrigIn_Pos` 8
- `#define TPI_FFCR_TrigIn_Msk` (0x1UL << TPI\_FFCR\_TrigIn\_Pos)
- `#define TPI_FFCR_EnFCont_Pos` 1
- `#define TPI_FFCR_EnFCont_Msk` (0x1UL << TPI\_FFCR\_EnFCont\_Pos)



- #define [TPI\\_TRIGGER\\_TRIGGER\\_Pos](#) 0
- #define [TPI\\_TRIGGER\\_TRIGGER\\_Msk](#) (0x1UL << TPI\_TRIGGER\_TRIGGER\_Pos)
- #define [TPI\\_FIFO0\\_ITM\\_ATVALID\\_Pos](#) 29
- #define [TPI\\_FIFO0\\_ITM\\_ATVALID\\_Msk](#) (0x3UL << TPI\_FIFO0\_ITM\_ATVALID\_Pos)
- #define [TPI\\_FIFO0\\_ITM\\_bytecount\\_Pos](#) 27
- #define [TPI\\_FIFO0\\_ITM\\_bytecount\\_Msk](#) (0x3UL << TPI\_FIFO0\_ITM\_bytecount\_Pos)
- #define [TPI\\_FIFO0\\_ETM\\_ATVALID\\_Pos](#) 26
- #define [TPI\\_FIFO0\\_ETM\\_ATVALID\\_Msk](#) (0x3UL << TPI\_FIFO0\_ETM\_ATVALID\_Pos)
- #define [TPI\\_FIFO0\\_ETM\\_bytecount\\_Pos](#) 24
- #define [TPI\\_FIFO0\\_ETM\\_bytecount\\_Msk](#) (0x3UL << TPI\_FIFO0\_ETM\_bytecount\_Pos)
- #define [TPI\\_FIFO0\\_ETM2\\_Pos](#) 16
- #define [TPI\\_FIFO0\\_ETM2\\_Msk](#) (0xFFUL << TPI\_FIFO0\_ETM2\_Pos)
- #define [TPI\\_FIFO0\\_ETM1\\_Pos](#) 8
- #define [TPI\\_FIFO0\\_ETM1\\_Msk](#) (0xFFUL << TPI\_FIFO0\_ETM1\_Pos)
- #define [TPI\\_FIFO0\\_ETM0\\_Pos](#) 0
- #define [TPI\\_FIFO0\\_ETM0\\_Msk](#) (0xFFUL << TPI\_FIFO0\_ETM0\_Pos)
- #define [TPI\\_ITATBCTR2\\_ATREADY\\_Pos](#) 0
- #define [TPI\\_ITATBCTR2\\_ATREADY\\_Msk](#) (0x1UL << TPI\_ITATBCTR2\_ATREADY\_Pos)
- #define [TPI\\_FIFO1\\_ITM\\_ATVALID\\_Pos](#) 29
- #define [TPI\\_FIFO1\\_ITM\\_ATVALID\\_Msk](#) (0x3UL << TPI\_FIFO1\_ITM\_ATVALID\_Pos)
- #define [TPI\\_FIFO1\\_ITM\\_bytecount\\_Pos](#) 27
- #define [TPI\\_FIFO1\\_ITM\\_bytecount\\_Msk](#) (0x3UL << TPI\_FIFO1\_ITM\_bytecount\_Pos)
- #define [TPI\\_FIFO1\\_ETM\\_ATVALID\\_Pos](#) 26
- #define [TPI\\_FIFO1\\_ETM\\_ATVALID\\_Msk](#) (0x3UL << TPI\_FIFO1\_ETM\_ATVALID\_Pos)
- #define [TPI\\_FIFO1\\_ETM\\_bytecount\\_Pos](#) 24
- #define [TPI\\_FIFO1\\_ETM\\_bytecount\\_Msk](#) (0x3UL << TPI\_FIFO1\_ETM\_bytecount\_Pos)
- #define [TPI\\_FIFO1\\_ITM2\\_Pos](#) 16
- #define [TPI\\_FIFO1\\_ITM2\\_Msk](#) (0xFFUL << TPI\_FIFO1\_ITM2\_Pos)
- #define [TPI\\_FIFO1\\_ITM1\\_Pos](#) 8
- #define [TPI\\_FIFO1\\_ITM1\\_Msk](#) (0xFFUL << TPI\_FIFO1\_ITM1\_Pos)
- #define [TPI\\_FIFO1\\_ITM0\\_Pos](#) 0
- #define [TPI\\_FIFO1\\_ITM0\\_Msk](#) (0xFFUL << TPI\_FIFO1\_ITM0\_Pos)
- #define [TPI\\_ITATBCTR0\\_ATREADY\\_Pos](#) 0
- #define [TPI\\_ITATBCTR0\\_ATREADY\\_Msk](#) (0x1UL << TPI\_ITATBCTR0\_ATREADY\_Pos)
- #define [TPI\\_ITCTRL\\_Mode\\_Pos](#) 0
- #define [TPI\\_ITCTRL\\_Mode\\_Msk](#) (0x1UL << TPI\_ITCTRL\_Mode\_Pos)
- #define [TPI\\_DEVID\\_NRZVALID\\_Pos](#) 11
- #define [TPI\\_DEVID\\_NRZVALID\\_Msk](#) (0x1UL << TPI\_DEVID\_NRZVALID\_Pos)
- #define [TPI\\_DEVID\\_MANCVALID\\_Pos](#) 10
- #define [TPI\\_DEVID\\_MANCVALID\\_Msk](#) (0x1UL << TPI\_DEVID\_MANCVALID\_Pos)
- #define [TPI\\_DEVID\\_PTINVALID\\_Pos](#) 9
- #define [TPI\\_DEVID\\_PTINVALID\\_Msk](#) (0x1UL << TPI\_DEVID\_PTINVALID\_Pos)
- #define [TPI\\_DEVID\\_MinBufSz\\_Pos](#) 6
- #define [TPI\\_DEVID\\_MinBufSz\\_Msk](#) (0x7UL << TPI\_DEVID\_MinBufSz\_Pos)
- #define [TPI\\_DEVID\\_AsynClkIn\\_Pos](#) 5
- #define [TPI\\_DEVID\\_AsynClkIn\\_Msk](#) (0x1UL << TPI\_DEVID\_AsynClkIn\_Pos)
- #define [TPI\\_DEVID\\_NrTraceInput\\_Pos](#) 0
- #define [TPI\\_DEVID\\_NrTraceInput\\_Msk](#) (0x1FUL << TPI\_DEVID\_NrTraceInput\_Pos)
- #define [TPI\\_DEVTYPE\\_SubType\\_Pos](#) 0
- #define [TPI\\_DEVTYPE\\_SubType\\_Msk](#) (0xFUL << TPI\_DEVTYPE\_SubType\_Pos)
- #define [TPI\\_DEVTYPE\\_MajorType\\_Pos](#) 4
- #define [TPI\\_DEVTYPE\\_MajorType\\_Msk](#) (0xFUL << TPI\_DEVTYPE\_MajorType\_Pos)
- #define [CoreDebug\\_DHCSR\\_DBGKEY\\_Pos](#) 16
- #define [CoreDebug\\_DHCSR\\_DBGKEY\\_Msk](#) (0xFFFFFUL << CoreDebug\_DHCSR\_DBGKEY\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_RESET\\_ST\\_Pos](#) 25

- #define [CoreDebug\\_DHCSR\\_S\\_RESET\\_ST\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_RESET\_ST\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_RETIRE\\_ST\\_Pos](#) 24
- #define [CoreDebug\\_DHCSR\\_S\\_RETIRE\\_ST\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_RETIRE\_ST\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_LOCKUP\\_Pos](#) 19
- #define [CoreDebug\\_DHCSR\\_S\\_LOCKUP\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_LOCKUP\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_SLEEP\\_Pos](#) 18
- #define [CoreDebug\\_DHCSR\\_S\\_SLEEP\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_SLEEP\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_HALT\\_Pos](#) 17
- #define [CoreDebug\\_DHCSR\\_S\\_HALT\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_HALT\_Pos)
- #define [CoreDebug\\_DHCSR\\_S\\_REGRDY\\_Pos](#) 16
- #define [CoreDebug\\_DHCSR\\_S\\_REGRDY\\_Msk](#) (1UL << CoreDebug\_DHCSR\_S\_REGRDY\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_SNAPSTALL\\_Pos](#) 5
- #define [CoreDebug\\_DHCSR\\_C\\_SNAPSTALL\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_SNAPSTALL\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_MASKINTS\\_Pos](#) 3
- #define [CoreDebug\\_DHCSR\\_C\\_MASKINTS\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_MASKINTS\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_STEP\\_Pos](#) 2
- #define [CoreDebug\\_DHCSR\\_C\\_STEP\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_STEP\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_HALT\\_Pos](#) 1
- #define [CoreDebug\\_DHCSR\\_C\\_HALT\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_HALT\_Pos)
- #define [CoreDebug\\_DHCSR\\_C\\_DEBUGEN\\_Pos](#) 0
- #define [CoreDebug\\_DHCSR\\_C\\_DEBUGEN\\_Msk](#) (1UL << CoreDebug\_DHCSR\_C\_DEBUGEN\_Pos)
- #define [CoreDebug\\_DCRSR\\_REGWnR\\_Pos](#) 16
- #define [CoreDebug\\_DCRSR\\_REGWnR\\_Msk](#) (1UL << CoreDebug\_DCRSR\_REGWnR\_Pos)
- #define [CoreDebug\\_DCRSR\\_REGSEL\\_Pos](#) 0
- #define [CoreDebug\\_DCRSR\\_REGSEL\\_Msk](#) (0x1FUL << CoreDebug\_DCRSR\_REGSEL\_Pos)
- #define [CoreDebug\\_DEMCR\\_TRCENA\\_Pos](#) 24
- #define [CoreDebug\\_DEMCR\\_TRCENA\\_Msk](#) (1UL << CoreDebug\_DEMCR\_TRCENA\_Pos)
- #define [CoreDebug\\_DEMCR\\_MON\\_REQ\\_Pos](#) 19
- #define [CoreDebug\\_DEMCR\\_MON\\_REQ\\_Msk](#) (1UL << CoreDebug\_DEMCR\_MON\_REQ\_Pos)
- #define [CoreDebug\\_DEMCR\\_MON\\_STEP\\_Pos](#) 18
- #define [CoreDebug\\_DEMCR\\_MON\\_STEP\\_Msk](#) (1UL << CoreDebug\_DEMCR\_MON\_STEP\_Pos)
- #define [CoreDebug\\_DEMCR\\_MON\\_PEND\\_Pos](#) 17
- #define [CoreDebug\\_DEMCR\\_MON\\_PEND\\_Msk](#) (1UL << CoreDebug\_DEMCR\_MON\_PEND\_Pos)
- #define [CoreDebug\\_DEMCR\\_MON\\_EN\\_Pos](#) 16
- #define [CoreDebug\\_DEMCR\\_MON\\_EN\\_Msk](#) (1UL << CoreDebug\_DEMCR\_MON\_EN\_Pos)
- #define [CoreDebug\\_DEMCR\\_VC\\_HARDERR\\_Pos](#) 10
- #define [CoreDebug\\_DEMCR\\_VC\\_HARDERR\\_Msk](#) (1UL << CoreDebug\_DEMCR\_VC\_HARDERR\_Pos)
- #define [CoreDebug\\_DEMCR\\_VC\\_INTERR\\_Pos](#) 9
- #define [CoreDebug\\_DEMCR\\_VC\\_INTERR\\_Msk](#) (1UL << CoreDebug\_DEMCR\_VC\_INTERR\_Pos)
- #define [CoreDebug\\_DEMCR\\_VC\\_BUSERR\\_Pos](#) 8
- #define [CoreDebug\\_DEMCR\\_VC\\_BUSERR\\_Msk](#) (1UL << CoreDebug\_DEMCR\_VC\_BUSERR\_Pos)
- #define [CoreDebug\\_DEMCR\\_VC\\_STATERR\\_Pos](#) 7
- #define [CoreDebug\\_DEMCR\\_VC\\_STATERR\\_Msk](#) (1UL << CoreDebug\_DEMCR\_VC\_STATERR\_Pos)
- #define [CoreDebug\\_DEMCR\\_VC\\_CHKERR\\_Pos](#) 6
- #define [CoreDebug\\_DEMCR\\_VC\\_CHKERR\\_Msk](#) (1UL << CoreDebug\_DEMCR\_VC\_CHKERR\_Pos)
- #define [CoreDebug\\_DEMCR\\_VC\\_NOCPERR\\_Pos](#) 5
- #define [CoreDebug\\_DEMCR\\_VC\\_NOCPERR\\_Msk](#) (1UL << CoreDebug\_DEMCR\_VC\_NOCPERR\_Pos)
- #define [CoreDebug\\_DEMCR\\_VC\\_MMERR\\_Pos](#) 4
- #define [CoreDebug\\_DEMCR\\_VC\\_MMERR\\_Msk](#) (1UL << CoreDebug\_DEMCR\_VC\_MMERR\_Pos)
- #define [CoreDebug\\_DEMCR\\_VC\\_CORERESET\\_Pos](#) 0
- #define [CoreDebug\\_DEMCR\\_VC\\_CORERESET\\_Msk](#) (1UL << CoreDebug\_DEMCR\_VC\_CORERESET\_Pos)
- #define [SCS\\_BASE](#) (0xE000E000UL)
- #define [ITM\\_BASE](#) (0xE0000000UL)
- #define [DWT\\_BASE](#) (0xE0001000UL)

- #define `TPI_BASE` (0xE0040000UL)
- #define `CoreDebug_BASE` (0xE000EDF0UL)
- #define `SysTick_BASE` (`SCS_BASE` + 0x0010UL)
- #define `NVIC_BASE` (`SCS_BASE` + 0x0100UL)
- #define `SCB_BASE` (`SCS_BASE` + 0x0D00UL)
- #define `SCnSCB` ((`SCnSCB_Type` \*) `SCS_BASE` )
- #define `SCB` ((`SCB_Type` \*) `SCB_BASE` )
- #define `SysTick` ((`SysTick_Type` \*) `SysTick_BASE` )
- #define `NVIC` ((`NVIC_Type` \*) `NVIC_BASE` )
- #define `ITM` ((`ITM_Type` \*) `ITM_BASE` )
- #define `DWT` ((`DWT_Type` \*) `DWT_BASE` )
- #define `TPI` ((`TPI_Type` \*) `TPI_BASE` )
- #define `CoreDebug` ((`CoreDebug_Type` \*) `CoreDebug_BASE` )
- #define `ITM_RXBUFFER_EMPTY` 0x5AA55AA5
  
- #define `__CM4_CMSIS_VERSION_MAIN` (0x03)
- #define `__CM4_CMSIS_VERSION_SUB` (0x20)
- #define `__CM4_CMSIS_VERSION`
- #define `__CORTEX_M` (0x04)
- #define `__CORE_CM4_H_DEPENDANT`
- #define `__I` volatile const
- #define `__O` volatile
- #define `__IO` volatile

## Functions

- `__STATIC_INLINE void NVIC_SetPriorityGrouping` (uint32\_t PriorityGroup)  
*Set Priority Grouping.*
- `__STATIC_INLINE uint32_t NVIC_GetPriorityGrouping` (void)  
*Get Priority Grouping.*
- `__STATIC_INLINE void NVIC_EnableIRQ` (IRQn\_Type IRQn)  
*Enable External Interrupt.*
- `__STATIC_INLINE void NVIC_DisableIRQ` (IRQn\_Type IRQn)  
*Disable External Interrupt.*
- `__STATIC_INLINE uint32_t NVIC_GetPendingIRQ` (IRQn\_Type IRQn)  
*Get Pending Interrupt.*
- `__STATIC_INLINE void NVIC_SetPendingIRQ` (IRQn\_Type IRQn)  
*Set Pending Interrupt.*
- `__STATIC_INLINE void NVIC_ClearPendingIRQ` (IRQn\_Type IRQn)  
*Clear Pending Interrupt.*
- `__STATIC_INLINE uint32_t NVIC_GetActive` (IRQn\_Type IRQn)  
*Get Active Interrupt.*
- `__STATIC_INLINE void NVIC_SetPriority` (IRQn\_Type IRQn, uint32\_t priority)  
*Set Interrupt Priority.*
- `__STATIC_INLINE uint32_t NVIC_GetPriority` (IRQn\_Type IRQn)  
*Get Interrupt Priority.*
- `__STATIC_INLINE uint32_t NVIC_EncodePriority` (uint32\_t PriorityGroup, uint32\_t PreemptPriority, uint32\_t SubPriority)  
*Encode Priority.*
- `__STATIC_INLINE void NVIC_DecodePriority` (uint32\_t Priority, uint32\_t PriorityGroup, uint32\_t \*pPreemptPriority, uint32\_t \*pSubPriority)  
*Decode Priority.*

- `__STATIC_INLINE void NVIC_SystemReset (void)`  
*System Reset.*
- `__STATIC_INLINE uint32_t SysTick_Config (uint32_t ticks)`  
*System Tick Configuration.*
- `__STATIC_INLINE uint32_t ITM_SendChar (uint32_t ch)`  
*ITM Send Character.*
- `__STATIC_INLINE int32_t ITM_ReceiveChar (void)`  
*ITM Receive Character.*
- `__STATIC_INLINE int32_t ITM_CheckChar (void)`  
*ITM Check Character.*

## Variables

- `volatile int32_t ITM_RxBuffer`

## 8.8.2 Macro Definition Documentation

### 8.8.2.1 `#define __CM4_CMSIS_VERSION`

#### Value:

```
((__CM4_CMSIS_VERSION_MAIN << 16) | \
    __CM4_CMSIS_VERSION_SUB)
```

CMSIS HAL version number

Definition at line 73 of file core\_cm4.h.

### 8.8.2.2 `#define __CM4_CMSIS_VERSION_MAIN (0x03)`

[31:16] CMSIS HAL main version

Definition at line 71 of file core\_cm4.h.

### 8.8.2.3 `#define __CM4_CMSIS_VERSION_SUB (0x20)`

[15:0] CMSIS HAL sub version

Definition at line 72 of file core\_cm4.h.

### 8.8.2.4 `#define __CORE_CM4_H_DEPENDANT`

`__FPU_USED` indicates whether an FPU is used or not. For this, `__FPU_PRESENT` has to be checked prior to making use of FPU specific registers and functions.

Definition at line 178 of file core\_cm4.h.

### 8.8.2.5 `#define __CORE_CM4_H_GENERIC`

Definition at line 47 of file core\_cm4.h.

#### 8.8.2.6 #define \_\_CORTEX\_M (0x04)

Cortex-M Core

Definition at line 76 of file core\_cm4.h.

#### 8.8.2.7 #define \_\_I volatile const

Defines 'read only' permissions

Definition at line 219 of file core\_cm4.h.

#### 8.8.2.8 #define \_\_IO volatile

Defines 'read / write' permissions

Definition at line 222 of file core\_cm4.h.

#### 8.8.2.9 #define \_\_O volatile

Defines 'write only' permissions

Definition at line 221 of file core\_cm4.h.

## 8.9 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm4\_simd.h File Reference

### 8.9.1 Detailed Description

CMSIS Cortex-M4 SIMD Header File.

Version

V3.20

Date

25. February 2013

Note

### Macros

- [#define \\_\\_CORE\\_CM4\\_SIMD\\_H](#)

### 8.9.2 Macro Definition Documentation

#### 8.9.2.1 #define \_\_CORE\_CM4\_SIMD\_H

Definition at line 43 of file core\_cm4\_simd.h.

## 8.10 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cmFunc.h File Reference

### 8.10.1 Detailed Description

CMSIS Cortex-M Core Function Access Header File.

Version

V3.20

Date

25. February 2013

Note

## 8.11 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cmInstr.h File Reference

### 8.11.1 Detailed Description

CMSIS Cortex-M Core Instruction Access Header File.

Version

V3.20

Date

05. March 2013

Note

## 8.12 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/cpuctrl\_5411x.h File Reference

### Enumerations

- enum [CORESELECT\\_T](#) { [CORESELECT\\_M0PLUS](#) = 0, [CORESELECT\\_M4](#) }

### Functions

- `__STATIC_INLINE bool Chip\_CPU\_IsM4Core (void)`  
*Determine which MCU this code is running on.*
- void [Chip\\_CPU\\_SelectMasterCore](#) ([CORESELECT\\_T](#) master, [CORESELECT\\_T](#) ownerPower)  
*Select master core and system power control ownership.*
- bool [Chip\\_CPU\\_IsMasterCore](#) (void)  
*Determine if this core is a slave or master.*
- void [Chip\\_CPU\\_CM0Boot](#) (uint32\_t \*coentry, uint32\_t \*costackptr)  
*Setup M0+ boot and reset M0+ core.*
- void [Chip\\_CPU\\_CM4Boot](#) (uint32\_t \*coentry, uint32\_t \*costackptr)  
*Setup M4 boot and reset M4 core.*

## 8.13 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/crc\_5411x.h File Reference

### Data Structures

- struct [LPC\\_CRC\\_T](#)  
*CRC register block structure.*

### Macros

- #define [CRC\\_MODE\\_POLY\\_BITMASK](#) ((0x03)) /\*\* CRC polynomial Bit mask \*/
- #define [CRC\\_MODE\\_POLY\\_CCITT](#) (0x00) /\*\* Select CRC-CCITT polynomial \*/
- #define [CRC\\_MODE\\_POLY\\_CRC16](#) (0x01) /\*\* Select CRC-16 polynomial \*/
- #define [CRC\\_MODE\\_POLY\\_CRC32](#) (0x02) /\*\* Select CRC-32 polynomial \*/
- #define [CRC\\_MODE\\_WRDATA\\_BITMASK](#) (0x03 << 2) /\*\* CRC WR\_Data Config Bit mask \*/
- #define [CRC\\_MODE\\_WRDATA\\_BIT\\_RVS](#) (1 << 2) /\*\* Select Bit order reverse for WR\_DATA (per byte) \*/
- #define [CRC\\_MODE\\_WRDATA\\_CMPL](#) (1 << 3) /\*\* Select One's complement for WR\_DATA \*/
- #define [CRC\\_MODE\\_SUM\\_BITMASK](#) (0x03 << 4) /\*\* CRC Sum Config Bit mask \*/
- #define [CRC\\_MODE\\_SUM\\_BIT\\_RVS](#) (1 << 4) /\*\* Select Bit order reverse for CRC\_SUM \*/
- #define [CRC\\_MODE\\_SUM\\_CMPL](#) (1 << 5) /\*\* Select One's complement for CRC\_SUM \*/
- #define [MODE\\_CFG\\_CCITT](#) (0x00) /\*\* Pre-defined mode word for default CCITT setup \*/
- #define [MODE\\_CFG\\_CRC16](#) (0x15) /\*\* Pre-defined mode word for default CRC16 setup \*/
- #define [MODE\\_CFG\\_CRC32](#) (0x36) /\*\* Pre-defined mode word for default CRC32 setup \*/
- #define [CRC\\_SEED\\_CCITT](#) (0x0000FFFF) /\*\* Initial seed value for CCITT mode \*/
- #define [CRC\\_SEED\\_CRC16](#) (0x00000000) /\*\* Initial seed value for CRC16 mode \*/
- #define [CRC\\_SEED\\_CRC32](#) (0xFFFFFFFF) /\*\* Initial seed value for CRC32 mode \*/

### Enumerations

- enum [CRC\\_POLY\\_T](#) { [CRC\\_POLY\\_CCITT](#) = [CRC\\_MODE\\_POLY\\_CCITT](#), [CRC\\_POLY\\_CRC16](#) = [CRC\\_MODE\\_POLY\\_CRC16](#), [CRC\\_POLY\\_CRC32](#) = [CRC\\_MODE\\_POLY\\_CRC32](#), [CRC\\_POLY\\_LAST](#) }
- CRC polynomial.*

### Functions

- `__STATIC_INLINE void Chip\_CRC\_Init (LPC\_CRC\_T *pCRC)`  
*Initializes the CRC Engine.*
- `__STATIC_INLINE void Chip\_CRC\_Deinit (LPC\_CRC\_T *pCRC)`  
*Deinitializes the CRC Engine.*
- `__STATIC_INLINE void Chip\_CRC\_SetPoly (LPC\_CRC\_T *pCRC, CRC\_POLY\_T poly, uint32_t flags)`  
*Set the polynomial used for the CRC calculation.*
- `__STATIC_INLINE void Chip\_CRC\_UseCRC16 (LPC\_CRC\_T *pCRC)`  
*Sets up the CRC engine for CRC16 mode.*
- `__STATIC_INLINE void Chip\_CRC\_UseCRC32 (LPC\_CRC\_T *pCRC)`  
*Sets up the CRC engine for CRC32 mode.*
- `__STATIC_INLINE void Chip\_CRC\_UseCCITT (LPC\_CRC\_T *pCRC)`  
*Sets up the CRC engine for CCITT mode.*
- `__STATIC_INLINE void Chip\_CRC\_UseDefaultConfig (LPC\_CRC\_T *pCRC, CRC\_POLY\_T poly)`  
*Engage the CRC engine with defaults based on the polynomial to be used.*
- `__STATIC_INLINE void Chip\_CRC\_SetMode (LPC\_CRC\_T *pCRC, uint32_t mode)`  
*Set the CRC Mode bits.*
- `__STATIC_INLINE uint32_t Chip\_CRC\_GetMode (LPC\_CRC\_T *pCRC)`

- Get the CRC Mode bits.*

    - `__STATIC_INLINE void Chip\_CRC\_SetSeed (LPC\_CRC\_T *pCRC, uint32_t seed)`

*Set the seed bits used by the CRC\_SUM register.*

  - `__STATIC_INLINE uint32_t Chip\_CRC\_GetSeed (LPC\_CRC\_T *pCRC)`
- Get the CRC seed value.*
- `__STATIC_INLINE void Chip\_CRC\_Write8 (LPC\_CRC\_T *pCRC, uint8_t data)`
- Convenience function for writing 8-bit data to the CRC engine.*
- `__STATIC_INLINE void Chip\_CRC\_Write16 (LPC\_CRC\_T *pCRC, uint16_t data)`
- Convenience function for writing 16-bit data to the CRC engine.*
- `__STATIC_INLINE void Chip\_CRC\_Write32 (LPC\_CRC\_T *pCRC, uint32_t data)`
- Convenience function for writing 32-bit data to the CRC engine.*
- `__STATIC_INLINE uint32_t Chip\_CRC\_Sum (LPC\_CRC\_T *pCRC)`
- Gets the CRC Sum based on the Mode and Seed as previously configured.*
- `__STATIC_INLINE uint32_t Chip\_CRC\_CRC8 (LPC\_CRC\_T *pCRC, const uint8_t *data, uint32_t bytes)`
- Convenience function for computing a standard CCITT checksum from an 8-bit data block.*
- `__STATIC_INLINE uint32_t Chip\_CRC\_CRC16 (LPC\_CRC\_T *pCRC, const uint16_t *data, uint32_t hwords)`
- Convenience function for computing a standard CRC16 checksum from 16-bit data block.*
- `__STATIC_INLINE uint32_t Chip\_CRC\_CRC32 (LPC\_CRC\_T *pCRC, const uint32_t *data, uint32_t words)`
- Convenience function for computing a standard CRC32 checksum from 32-bit data block.*

## 8.14 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/dma\_5411x.h File Reference

### Data Structures

- struct [LPC\\_DMA\\_COMMON\\_T](#)  
*DMA Controller shared registers structure.*
- struct [LPC\\_DMA\\_CHANNEL\\_T](#)  
*DMA Controller shared registers structure.*
- struct [LPC\\_DMA\\_T](#)  
*DMA Controller register block structure.*
- struct [DMA\\_CHDESC\\_T](#)

### Macros

- `#define MAX\_DMA\_CHANNEL (20)`
- `#define DMA\_INTSTAT\_ACTIVEINT 0x2`
- `#define DMA\_INTSTAT\_ACTIVEERRINT 0x4`
- `#define DMA\_ADDR(addr) ((uint32_t) (addr))`
- `#define DMA\_CFG\_PERIPHREQEN (1 << 0)`
- `#define DMA\_CFG\_HWTRIGEN (1 << 1)`
- `#define DMA\_CFG\_TRIGPOL\_LOW (0 << 4)`
- `#define DMA\_CFG\_TRIGPOL\_HIGH (1 << 4)`
- `#define DMA\_CFG\_TRIGTYPE\_EDGE (0 << 5)`
- `#define DMA\_CFG\_TRIGTYPE\_LEVEL (1 << 5)`
- `#define DMA\_CFG\_TRIGBURST\_SINGL (0 << 6)`
- `#define DMA\_CFG\_TRIGBURST\_BURST (1 << 6)`
- `#define DMA\_CFG\_BURSTPOWER\_1 (0 << 8)`
- `#define DMA\_CFG\_BURSTPOWER\_2 (1 << 8)`
- `#define DMA\_CFG\_BURSTPOWER\_4 (2 << 8)`
- `#define DMA\_CFG\_BURSTPOWER\_8 (3 << 8)`



- #define DMA\_CFG\_BURSTPOWER\_16 (4 << 8)
- #define DMA\_CFG\_BURSTPOWER\_32 (5 << 8)
- #define DMA\_CFG\_BURSTPOWER\_64 (6 << 8)
- #define DMA\_CFG\_BURSTPOWER\_128 (7 << 8)
- #define DMA\_CFG\_BURSTPOWER\_256 (8 << 8)
- #define DMA\_CFG\_BURSTPOWER\_512 (9 << 8)
- #define DMA\_CFG\_BURSTPOWER\_1024 (10 << 8)
- #define DMA\_CFG\_BURSTPOWER(n) ((n) << 8)
- #define DMA\_CFG\_SRCBURSTWRAP (1 << 14)
- #define DMA\_CFG\_DSTBURSTWRAP (1 << 15)
- #define DMA\_CFG\_CHPRIORITY(p) ((p) << 16)
- #define DMA\_CTLSTAT\_VALIDPENDING (1 << 0)
- #define DMA\_CTLSTAT\_TRIG (1 << 2)
- #define DMA\_XFERCFG\_CFGVALID (1 << 0)
- #define DMA\_XFERCFG\_RELOAD (1 << 1)
- #define DMA\_XFERCFG\_SWTRIG (1 << 2)
- #define DMA\_XFERCFG\_CLRTRIG (1 << 3)
- #define DMA\_XFERCFG\_SETINTA (1 << 4)
- #define DMA\_XFERCFG\_SETINTB (1 << 5)
- #define DMA\_XFERCFG\_WIDTH\_8 (0 << 8)
- #define DMA\_XFERCFG\_WIDTH\_16 (1 << 8)
- #define DMA\_XFERCFG\_WIDTH\_32 (2 << 8)
- #define DMA\_XFERCFG\_SRCINC\_0 (0 << 12)
- #define DMA\_XFERCFG\_SRCINC\_1 (1 << 12)
- #define DMA\_XFERCFG\_SRCINC\_2 (2 << 12)
- #define DMA\_XFERCFG\_SRCINC\_4 (3 << 12)
- #define DMA\_XFERCFG\_DSTINC\_0 (0 << 14)
- #define DMA\_XFERCFG\_DSTINC\_1 (1 << 14)
- #define DMA\_XFERCFG\_DSTINC\_2 (2 << 14)
- #define DMA\_XFERCFG\_DSTINC\_4 (3 << 14)
- #define DMA\_XFERCFG\_XFERCOUNT(n) ((n - 1) << 16)

## Enumerations

- enum DMA\_CHID\_T {  
DMA\_CH0, DMA\_CH1, DMA\_CH2, DMA\_CH3,  
DMA\_CH4, DMA\_CH5, DMA\_CH6, DMA\_CH7,  
DMA\_CH8, DMA\_CH9, DMA\_CH10, DMA\_CH11,  
DMA\_CH12, DMA\_CH13, DMA\_CH14, DMA\_CH15,  
DMA\_CH16, DMA\_CH17, DMA\_CH18, DMA\_CH19,  
DMAREQ\_FLEXCOMM0\_RX = DMA\_CH0, DMAREQ\_FLEXCOMM0\_TX, DMAREQ\_FLEXCOMM1\_RX,  
DMAREQ\_FLEXCOMM1\_TX,  
DMAREQ\_FLEXCOMM2\_RX, DMAREQ\_FLEXCOMM2\_TX, DMAREQ\_FLEXCOMM3\_RX, DMAREQ\_FLEXCOMM3\_TX,  
DMAREQ\_FLEXCOMM4\_RX, DMAREQ\_FLEXCOMM4\_TX, DMAREQ\_FLEXCOMM5\_RX, DMAREQ\_FLEXCOMM5\_TX,  
DMAREQ\_FLEXCOMM6\_RX, DMAREQ\_FLEXCOMM6\_TX, DMAREQ\_FLEXCOMM7\_RX, DMAREQ\_FLEXCOMM7\_TX,  
DMAREQ\_DMIC0, DMAREQ\_DMIC1, DMAREQ\_SPIFI }

## Functions

- `__STATIC_INLINE void Chip_DMA_Init (LPC_DMA_T *pDMA)`  
*Initialize DMA controller.*
- `__STATIC_INLINE void Chip_DMA_DeInit (LPC_DMA_T *pDMA)`  
*De-Initialize DMA controller.*
- `__STATIC_INLINE void Chip_DMA_Enable (LPC_DMA_T *pDMA)`  
*Enable DMA controller.*
- `__STATIC_INLINE void Chip_DMA_Disable (LPC_DMA_T *pDMA)`  
*Disable DMA controller.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetIntStatus (LPC_DMA_T *pDMA)`  
*Get pending interrupt or error interrupts.*
- `__STATIC_INLINE void Chip_DMA_SetSRAMBase (LPC_DMA_T *pDMA, uint32_t base)`  
*Set DMA controller SRAM base address.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetSRAMBase (LPC_DMA_T *pDMA)`  
*Returns DMA controller SRAM base address.*
- `__STATIC_INLINE void Chip_DMA_EnableChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Enables a single DMA channel.*
- `__STATIC_INLINE void Chip_DMA_DisableChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Disables a single DMA channel.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetEnabledChannels (LPC_DMA_T *pDMA)`  
*Returns all enabled DMA channels.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetActiveChannels (LPC_DMA_T *pDMA)`  
*Returns all active DMA channels.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetBusyChannels (LPC_DMA_T *pDMA)`  
*Returns all busy DMA channels.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetErrorIntChannels (LPC_DMA_T *pDMA)`  
*Returns pending error interrupt status for all DMA channels.*
- `__STATIC_INLINE void Chip_DMA_ClearErrorIntChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Clears a pending error interrupt status for a single DMA channel.*
- `__STATIC_INLINE void Chip_DMA_EnableIntChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Enables a single DMA channel's interrupt used in common DMA interrupt.*
- `__STATIC_INLINE void Chip_DMA_DisableIntChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Disables a single DMA channel's interrupt used in common DMA interrupt.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetEnableIntChannels (LPC_DMA_T *pDMA)`  
*Returns all enabled interrupt channels.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetActiveIntAChannels (LPC_DMA_T *pDMA)`  
*Returns active A interrupt status for all channels.*
- `__STATIC_INLINE void Chip_DMA_ClearActiveIntAChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Clears active A interrupt status for a single channel.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetActiveIntBChannels (LPC_DMA_T *pDMA)`  
*Returns active B interrupt status for all channels.*
- `__STATIC_INLINE void Chip_DMA_ClearActiveIntBChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Clears active B interrupt status for a single channel.*
- `__STATIC_INLINE void Chip_DMA_SetValidChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Sets the VALIDPENDING control bit for a single channel.*
- `__STATIC_INLINE void Chip_DMA_SetTrigChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Sets the TRIG bit for a single channel.*
- `__STATIC_INLINE void Chip_DMA_AbortChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Aborts a DMA operation for a single channel.*

- `__STATIC_INLINE void Chip_DMA_SetupChannelConfig (LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t cfg)`  
*Setup a DMA channel configuration.*
- `__STATIC_INLINE uint32_t Chip_DMA_GetChannelStatus (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Returns channel specific status flags.*
- `__STATIC_INLINE void Chip_DMA_SetupChannelTransfer (LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t cfg)`  
*Setup a DMA channel transfer configuration.*
- `__STATIC_INLINE void Chip_DMA_SetTranBits (LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t mask)`  
*Set DMA transfer register interrupt bits (safe)*
- `__STATIC_INLINE void Chip_DMA_ClearTranBits (LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t mask)`  
*Clear DMA transfer register interrupt bits (safe)*
- `__STATIC_INLINE void Chip_DMA_SetupChannelTransferSize (LPC_DMA_T *pDMA, DMA_CHID_T ch, uint32_t trans)`  
*Update the transfer size in an existing DMA channel transfer configuration.*
- `__STATIC_INLINE void Chip_DMA_SetChannelValid (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Sets a DMA channel configuration as valid.*
- `__STATIC_INLINE void Chip_DMA_SetChannelInvalid (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Sets a DMA channel configuration as invalid.*
- `__STATIC_INLINE void Chip_DMA_SWTriggerChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch)`  
*Performs a software trigger of the DMA channel.*
- `__STATIC_INLINE bool Chip_DMA_SetupTranChannel (LPC_DMA_T *pDMA, DMA_CHID_T ch, DMA_CHDESC_T *desc)`  
*Sets up a DMA channel with the passed DMA transfer descriptor.*

## Variables

- `DMA_CHDESC_T Chip_DMA_Table [MAX_DMA_CHANNEL]`

## 8.15 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/dma\_service\_5411x.h File Reference

### Data Structures

- struct `DMA_PERIPHERAL_CONTEXT_T`
- struct `DMA_DUAL_DESCRIPTOR_T`

### Typedefs

- typedef void(\* `DMA_CALLBACK_T`) (int32\_t)

### Functions

- void `Chip_DMASERVICE_Init (DMA_CHDESC_T *base)`  
*Initialize DMA service.*
- void `Chip_DMASERVICE_Isr (void)`  
*DMA service interrupt handler.*
- void `Chip_DMASERVICE_RegisterCb (const DMA_PERIPHERAL_CONTEXT_T *pContext, DMA_CALLBACK_T pCallback)`  
*Register callback function.*

- void `Chip_DMASERVICE_SingleBuffer` (const `DMA_PERIPHERAL_CONTEXT_T` \*pContext, uint32\_t p↔ Mem, uint32\_t length)  
*Use Single buffer mechanism.*
- void `Chip_DMASERVICE_DoubleBuffer` (const `DMA_PERIPHERAL_CONTEXT_T` \*pContext, uint32\_t p↔ Mem, uint32\_t length, `DMA_DUAL_DESCRIPTOR_T` \*pD)  
*Use double buffer mechanism.*

## 8.16 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/dmic\_5411x.h File Reference

### Data Structures

- struct `LPC_DMIC_Channel_Type`
- struct `LPC_DMIC_T`
- struct `DMIC_STATISTICS_T`  
*DMIC statistics structure.*
- struct `DMIC_CHANNEL_CONFIG_T`

### Macros

- #define `DMIC_FIFO_ENABLE_P` 0
- #define `DMIC_FIFO_RESETN_P` 1
- #define `DMIC_FIFO_INTREN_P` 2
- #define `DMIC_FIFO_DMAEN_P` 3
- #define `DMIC_FIFO_TLVL_P` 16
- #define `DMIC_FIFO_ENABLE` (1<<DMIC\_FIFO\_ENABLE\_P)
- #define `DMIC_FIFO_RESETN` (1<<DMIC\_FIFO\_RESETN\_P)
- #define `DMIC_FIFO_INTREN` (1<<DMIC\_FIFO\_INTREN\_P)
- #define `DMIC_FIFO_DMAEN` (1<<DMIC\_FIFO\_DMAEN\_P)
- #define `DMIC_FIFO_INT_P` 0
- #define `DMIC_FIFO_OVERRUN_P` 1
- #define `DMIC_FIFO_UNDERRUN_P` 2
- #define `DMIC_FIFO_INT` (1<<DMIC\_FIFO\_INT\_P)
- #define `DMIC_FIFO_OVERRUN` (1<<DMIC\_FIFO\_OVERRUN\_P)
- #define `DMIC_FIFO_UNDERRUN` (1<<DMIC\_FIFO\_UNDERRUN\_P)
- #define `DMIC_PHY_FALL_P` 0
- #define `DMIC_PHY_HALF_P` 1
- #define `DMIC_PHY_FALL` (1<<DMIC\_PHY\_FALL\_P)
- #define `DMIC_PHY_HALF` (1<<DMIC\_PHY\_HALF\_P)
- #define `DMIC_DCPOLE_P` 0
- #define `DMIC_DCGAIN_REDUCE_P` 4
- #define `DMIC_SATURATE_AT16BIT_P` 8

### Enumerations

- enum `OP_MODE_T` { `DMIC_OP_POLL`, `DMIC_OP_INTR`, `DMIC_OP_DMA` }
- enum `STEREO_SIDE_T` { `DMIC_LEFT` = 0, `DMIC_RIGHT` = 1 }
- enum `PDM_DIV_T` {  
`DMIC_PDM_DIV1` = 0, `DMIC_PDM_DIV2` = 1, `DMIC_PDM_DIV3` = 2, `DMIC_PDM_DIV4` = 3,  
`DMIC_PDM_DIV6` = 4, `DMIC_PDM_DIV8` = 5, `DMIC_PDM_DIV12` = 6, `DMIC_PDM_DIV16` = 7,  
`DMIC_PDM_DIV24` = 8, `DMIC_PDM_DIV32` = 9, `DMIC_PDM_DIV48` = 10, `DMIC_PDM_DIV64` = 11,  
`DMIC_PDM_DIV96` = 12, `DMIC_PDM_DIV128` = 13 }

- enum `COMPENSATION_T` { `DMIC_COMP0_0` = 0, `DMIC_COMP0_16` = 1, `DMIC_COMP0_15` = 2, `DMIC_COMP0_13` = 3 }
- enum `DC_REMOVAL_T` { `DMIC_DC_NOREMOVE` = 0, `DMIC_DC_CUT155` = 1, `DMIC_DC_CUT78` = 2, `DMIC_DC_CUT39` = 3 }
- enum `DMIC_IO_T` { `pdm_dual` = 0, `pdm_stereo` = 4, `pdm_bypass` = 3, `pdm_bypass_clk0` = 1, `pdm_bypass_clk1` = 2 }

## Functions

- void `Chip_DMIC_Init` (const `CHIP_SYSCON_CLOCK_T` clock, const `CHIP_SYSCON_PERIPH_RESET_T` reset)  
*Initialize DMIC interface.*
- void `Chip_DMIC_CfgIO` (`LPC_DMIC_T` \*pDMIC, `DMIC_IO_T` cfg)  
*Configure DMIC io.*
- void `Chip_DMIC_SetOpMode` (`LPC_DMIC_T` \*pDMIC, `OP_MODE_T` mode)  
*Set DMIC operating mode.*
- void `Chip_DMIC_CfgChannel` (`LPC_DMIC_T` \*pDMIC, uint32\_t channel, `DMIC_CHANNEL_CONFIG_T` \*channel\_cfg)  
*Configure DMIC channel.*
- void `Chip_DMIC_CfgChannelDc` (`LPC_DMIC_T` \*pDMIC, uint32\_t channel, `DC_REMOVAL_T` dc\_cut\_level, uint32\_t post\_dc\_gain\_reduce, bool saturate16bit)  
*Configure DMIC channel DC removal setting.*
- void `Chip_DMIC_Use2fs` (`LPC_DMIC_T` \*pDMIC, bool use2fs)  
*Configure Clock scaling.*
- void `Chip_DMIC_EnableChannel` (`LPC_DMIC_T` \*pDMIC, uint32\_t channelmask)  
*Configure Clock scaling.*
- void `Chip_DMIC_FifoChannel` (`LPC_DMIC_T` \*pDMIC, uint32\_t channel, uint32\_t trig\_level, uint32\_t enable, uint32\_t resetn)  
*Configure fifo settings for DMIC channel.*
- `__STATIC_INLINE` uint32\_t `Chip_DMIC_FifoGetStatus` (`LPC_DMIC_T` \*pDMIC, uint32\_t channel)  
*Get FIFO status.*
- `__STATIC_INLINE` void `Chip_DMIC_FifoClearStatus` (`LPC_DMIC_T` \*pDMIC, uint32\_t channel, uint32\_t mask)  
*Clear FIFO status.*
- `__STATIC_INLINE` uint32\_t `Chip_DMIC_FifoGetData` (`LPC_DMIC_T` \*pDMIC, uint32\_t channel)  
*Get FIFO data.*

## Variables

- `DMA_PERIPHERAL_CONTEXT_T` dmic\_ch0\_dma\_context
- `DMA_PERIPHERAL_CONTEXT_T` dmic\_ch1\_dma\_context
- `DMA_PERIPHERAL_CONTEXT_T` dmic\_ch0\_dma\_interleaved\_context
- `DMA_PERIPHERAL_CONTEXT_T` dmic\_ch1\_dma\_interleaved\_context

## 8.17 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/error.h File Reference

### Macros

- `#define` `offsetof`(s, m) (int) &(((s \*) 0)->m)
- `#define` `COMPILE_TIME_ASSERT`(pred)

## Enumerations

```

• enum ErrorCode_t {
    LPC_OK = 0, ERR_FAILED = -1, ERR_TIME_OUT = -2, ERR_BUSY = -3,
    ERR_ISP_BASE = 0x00000000, ERR_ISP_INVALID_COMMAND = ERR_ISP_BASE + 1, ERR_ISP_SRC_ADDR_ERROR, ERR_ISP_DST_ADDR_ERROR,
    ERR_ISP_SRC_ADDR_NOT_MAPPED, ERR_ISP_DST_ADDR_NOT_MAPPED, ERR_ISP_COUNT_ERROR, ERR_ISP_INVALID_SECTOR,
    ERR_ISP_SECTOR_NOT_BLANK, ERR_ISP_SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION, ERR_ISP_COMPARE_ERROR, ERR_ISP_BUSY,
    ERR_ISP_PARAM_ERROR, ERR_ISP_ADDR_ERROR, ERR_ISP_ADDR_NOT_MAPPED, ERR_ISP_CMD_LOCKED,
    ERR_ISP_INVALID_CODE, ERR_ISP_INVALID_BAUD_RATE, ERR_ISP_INVALID_STOP_BIT, ERR_ISP_CODE_READ_PROTECTION_ENABLED,
    ERR_ISP_INVALID_FLASH_UNIT, ERR_ISP_USER_CODE_CHECKSUM, ERR_ISP_SETTING_ACTIVE_PARTITION, ERR_ISP_IRC_NO_POWER,
    ERR_ISP_FLASH_NO_POWER, ERR_ISP_EEPROM_NO_POWER, ERR_ISP_EEPROM_NO_CLOCK, ERR_ISP_FLASH_NO_CLOCK,
    ERR_ISP_REINVOKE_ISP_CONFIG, ERR_API_BASE = 0x00010000, ERR_API_INVALID_PARAMS = ERR_API_BASE + 1, ERR_API_INVALID_PARAM1,
    ERR_API_INVALID_PARAM2, ERR_API_INVALID_PARAM3, ERR_API_MOD_INIT, ERR_SPIFI_BASE = 0x00020000,
    ERR_SPIFI_DEVICE_ERROR = ERR_SPIFI_BASE + 1, ERR_SPIFI_INTERNAL_ERROR, ERR_SPIFI_TIMEOUT, ERR_SPIFI_OPERAND_ERROR,
    ERR_SPIFI_STATUS_PROBLEM, ERR_SPIFI_UNKNOWN_EXT, ERR_SPIFI_UNKNOWN_ID, ERR_SPIFI_UNKNOWN_TYPE,
    ERR_SPIFI_UNKNOWN_MFG, ERR_SPIFI_NO_DEVICE, ERR_SPIFI_ERASE_NEEDED, ERR_SEC_AES_NO_ERROR = 0,
    ERR_SEC_AES_BASE = 0x00030000, ERR_SEC_AES_WRONG_CMD = ERR_SEC_AES_BASE + 1, ERR_SEC_AES_NOT_SUPPORTED, ERR_SEC_AES_KEY_ALREADY_PROGRAMMED,
    ERR_SEC_AES_DMA_CHANNEL_CFG, ERR_SEC_AES_DMA_MUX_CFG, ERR_SEC_AES_DMA_BUSY, ERR_USBD_BASE = 0x00040000,
    ERR_USBD_INVALID_REQ = ERR_USBD_BASE + 1, ERR_USBD_UNHANDLED, ERR_USBD_STALL, ERR_USBD_SEND_ZLP,
    ERR_USBD_SEND_DATA, ERR_USBD_BAD_DESC, ERR_USBD_BAD_CFG_DESC, ERR_USBD_BAD_INTF_DESC,
    ERR_USBD_BAD_EP_DESC, ERR_USBD_BAD_MEM_BUF, ERR_USBD_TOO_MANY_CLASS_HDLR, ERR_CGU_BASE = 0x00050000,
    ERR_CGU_NOT_IMPL = ERR_CGU_BASE + 1, ERR_CGU_INVALID_PARAM, ERR_CGU_INVALID_SLICE, ERR_CGU_OUTPUT_GEN,
    ERR_CGU_DIV_SRC, ERR_CGU_DIV_VAL, ERR_CGU_SRC, ERR_I2C_BASE = 0x00060000, ERR_I2C_BUSY = ERR_I2C_BASE, ERR_I2C_NAK, ERR_I2C_BUFFER_OVERFLOW, ERR_I2C_BYTE_COUNT_ERR,
    ERR_I2C_LOSS_OF_ARBITRATION, ERR_I2C_SLAVE_NOT_ADDRESSSED, ERR_I2C_LOSS_OF_ARBITRATION_NAK_BIT, ERR_I2C_GENERAL_FAILURE,
    ERR_I2C_REGS_SET_TO_DEFAULT, ERR_I2C_TIMEOUT, ERR_I2C_BUFFER_UNDERFLOW, ERR_I2C_PARAM,
    ERR_OTP_BASE = 0x00070000, ERR_OTP_WR_ENABLE_INVALID = ERR_OTP_BASE + 1, ERR_OTP_SOME_BITS_ALREADY_PROGRAMMED, ERR_OTP_ALL_DATA_OR_MASK_ZERO,
    ERR_OTP_WRITE_ACCESS_LOCKED, ERR_OTP_READ_DATA_MISMATCH, ERR_OTP_USB_ID_ENABLED, ERR_OTP_ETH_MAC_ENABLED,
    ERR_OTP_AES_KEYS_ENABLED, ERR_OTP_ILLEGAL_BANK, ERR_UART_BASE = 0x00080000, ERR_UART_RXD_BUSY = ERR_UART_BASE + 1,
    ERR_UART_TXD_BUSY, ERR_UART_OVERRUN_FRAME_PARITY_NOISE, ERR_UART_UNDERRUN, ERR_UART_PARAM,
    ERR_UART_BAUDRATE, ERR_CAN_BASE = 0x00090000, ERR_CAN_BAD_MEM_BUF = ERR_CAN_BASE + 1, ERR_CAN_INIT_FAIL,
    ERR_CANOPEN_INIT_FAIL, ERR_SPIFI_LITE_BASE = 0x000A0000, ERR_SPIFI_LITE_INVALID_ARG

```

```

UMENTS = ERR_SPIFI_LITE_BASE+1, ERR_SPIFI_LITE_BUSY,
ERR_SPIFI_LITE_MEMORY_MODE_ON, ERR_SPIFI_LITE_MEMORY_MODE_OFF, ERR_SPIFI_LITE_↵
_IN_DMA, ERR_SPIFI_LITE_NOT_IN_DMA,
PENDING_SPIFI_LITE, ERR_CLK_BASE = 0x000B0000, ERR_CLK_NOT_IMPL =ERR_CLK_BASE+1,
ERR_CLK_INVALID_PARAM,
ERR_CLK_INVALID_SLICE, ERR_CLK_OUTPUT_GEN, ERR_CLK_DIV_SRC, ERR_CLK_DIV_VAL,
ERR_CLK_SRC, ERR_CLK_PLL_FIN_TOO_SMALL, ERR_CLK_PLL_FIN_TOO_LARGE, ERR_CLK_PL↵
L_FOUT_TOO_SMALL,
ERR_CLK_PLL_FOUT_TOO_LARGE, ERR_CLK_PLL_NO_SOLUTION, ERR_CLK_PLL_MIN_PCT, ER↵
R_CLK_PLL_MAX_PCT,
ERR_CLK_OSC_FREQ, ERR_CLK_CFG, ERR_CLK_TIMEOUT, ERR_CLK_BASE_OFF,
ERR_CLK_OFF_DEADLOCK, ERR_PWR_BASE = 0x000C0000, PWR_ERROR_ILLEGAL_MODE =ER↵
R_PWR_BASE+1, PWR_ERROR_CLOCK_FREQ_TOO_HIGH,
PWR_ERROR_INVALID_STATE, PWR_ERROR_INVALID_CFG, PWR_ERROR_PVT_DETECT, ERR_↵
DMA_BASE = 0x000D0000,
ERR_DMA_ERROR_INT =ERR_DMA_BASE+1, ERR_DMA_CHANNEL_NUMBER, ERR_DMA_CHANN↵
EL_DISABLED, ERR_DMA_BUSY,
ERR_DMA_NOT_ALIGNMENT, ERR_DMA_PING_PONG_EN, ERR_DMA_CHANNEL_VALID_PENDING,
ERR_DMA_PARAM,
ERR_DMA_QUEUE_EMPTY, ERR_DMA_GENERAL, ERR_SPI_BASE = 0x000E0000, ERR_SPI_BUSY
=ERR_SPI_BASE,
ERR_SPI_RXOVERRUN, ERR_SPI_TXUNDERRUN, ERR_SPI_SELNASSERT, ERR_SPI_SELNDEAS↵
SERT,
ERR_SPI_CLKSTALL, ERR_SPI_PARAM, ERR_SPI_INVALID_LENGTH, ERR_ADC_BASE = 0x000↵
F0000,
ERR_ADC_OVERRUN =ERR_ADC_BASE+1, ERR_ADC_INVALID_CHANNEL, ERR_ADC_INVALID_S↵
EQUENCE, ERR_ADC_INVALID_SETUP,
ERR_ADC_PARAM, ERR_ADC_INVALID_LENGTH, ERR_ADC_NO_POWER, ERR_DM_BASE =
0x00100000,
ERR_DM_NOT_ENTERED =ERR_DM_BASE+1, ERR_DM_UNKNOWN_CMD, ERR_DM_COMM_FAIL }

```

## 8.17.1 Macro Definition Documentation

### 8.17.1.1 #define COMPILE\_TIME\_ASSERT( pred )

**Value:**

```

switch (0) { \
  case 0: \
  case pred:; }

```

Definition at line 268 of file error.h.

### 8.17.1.2 #define offsetof( s, m ) (int) &(((s \*) 0)->m)

Definition at line 265 of file error.h.

## 8.17.2 Enumeration Type Documentation

### 8.17.2.1 enum ErrorCode\_t

Error code returned by Boot ROM drivers/library functions

Error codes are a 32-bit value with :

- The 16 MSB contains the peripheral code number

- The 16 LSB contains an error code number associated to that peripheral

#### Enumerator

**LPC\_OK** 0x00000000 enum value returned on Success  
**ERR\_FAILED** 0xFFFFFFFF enum value returned on general failure  
**ERR\_TIME\_OUT** 0xFFFFFFF0 enum value returned on general timeout  
**ERR\_BUSY** 0xFFFFFFF1 enum value returned when resource is busy  
**ERR\_ISP\_BASE**  
**ERR\_ISP\_INVALID\_COMMAND**  
**ERR\_ISP\_SRC\_ADDR\_ERROR**  
**ERR\_ISP\_DST\_ADDR\_ERROR**  
**ERR\_ISP\_SRC\_ADDR\_NOT\_MAPPED**  
**ERR\_ISP\_DST\_ADDR\_NOT\_MAPPED**  
**ERR\_ISP\_COUNT\_ERROR**  
**ERR\_ISP\_INVALID\_SECTOR**  
**ERR\_ISP\_SECTOR\_NOT\_BLANK**  
**ERR\_ISP\_SECTOR\_NOT\_PREPARED\_FOR\_WRITE\_OPERATION**  
**ERR\_ISP\_COMPARE\_ERROR**  
**ERR\_ISP\_BUSY**  
**ERR\_ISP\_PARAM\_ERROR**  
**ERR\_ISP\_ADDR\_ERROR**  
**ERR\_ISP\_ADDR\_NOT\_MAPPED**  
**ERR\_ISP\_CMD\_LOCKED**  
**ERR\_ISP\_INVALID\_CODE**  
**ERR\_ISP\_INVALID\_BAUD\_RATE**  
**ERR\_ISP\_INVALID\_STOP\_BIT**  
**ERR\_ISP\_CODE\_READ\_PROTECTION\_ENABLED**  
**ERR\_ISP\_INVALID\_FLASH\_UNIT**  
**ERR\_ISP\_USER\_CODE\_CHECKSUM**  
**ERR\_ISP\_SETTING\_ACTIVE\_PARTITION**  
**ERR\_ISP\_IRC\_NO\_POWER**  
**ERR\_ISP\_FLASH\_NO\_POWER**  
**ERR\_ISP\_EEPROM\_NO\_POWER**  
**ERR\_ISP\_EEPROM\_NO\_CLOCK**  
**ERR\_ISP\_FLASH\_NO\_CLOCK**  
**ERR\_ISP\_REINVOKE\_ISP\_CONFIG**  
**ERR\_API\_BASE**  
**ERR\_API\_INVALID\_PARAMS** 0x00010001 Invalid parameters  
**ERR\_API\_INVALID\_PARAM1** 0x00010002 PARAM1 is invalid  
**ERR\_API\_INVALID\_PARAM2** 0x00010003 PARAM2 is invalid  
**ERR\_API\_INVALID\_PARAM3** 0x00010004 PARAM3 is invalid  
**ERR\_API\_MOD\_INIT** 0x00010005 API is called before module init  
**ERR\_SPIFI\_BASE**  
**ERR\_SPIFI\_DEVICE\_ERROR**  
**ERR\_SPIFI\_INTERNAL\_ERROR**



***ERR\_SPIFI\_TIMEOUT***  
***ERR\_SPIFI\_OPERAND\_ERROR***  
***ERR\_SPIFI\_STATUS\_PROBLEM***  
***ERR\_SPIFI\_UNKNOWN\_EXT***  
***ERR\_SPIFI\_UNKNOWN\_ID***  
***ERR\_SPIFI\_UNKNOWN\_TYPE***  
***ERR\_SPIFI\_UNKNOWN\_MFG***  
***ERR\_SPIFI\_NO\_DEVICE***  
***ERR\_SPIFI\_ERASE\_NEEDED***  
***SEC\_AES\_NO\_ERROR***  
***ERR\_SEC\_AES\_BASE***  
***ERR\_SEC\_AES\_WRONG\_CMD***  
***ERR\_SEC\_AES\_NOT\_SUPPORTED***  
***ERR\_SEC\_AES\_KEY\_ALREADY\_PROGRAMMED***  
***ERR\_SEC\_AES\_DMA\_CHANNEL\_CFG***  
***ERR\_SEC\_AES\_DMA\_MUX\_CFG***  
***SEC\_AES\_DMA\_BUSY***  
***ERR\_USBD\_BASE***  
***ERR\_USBD\_INVALID\_REQ*** 0x00040001 invalid request  
***ERR\_USBD\_UNHANDLED*** 0x00040002 Callback did not process the event  
***ERR\_USBD\_STALL*** 0x00040003 Stall the endpoint on which the call back is called  
***ERR\_USBD\_SEND\_ZLP*** 0x00040004 Send ZLP packet on the endpoint on which the call back is called  
***ERR\_USBD\_SEND\_DATA*** 0x00040005 Send data packet on the endpoint on which the call back is called  
***ERR\_USBD\_BAD\_DESC*** 0x00040006 Bad descriptor  
***ERR\_USBD\_BAD\_CFG\_DESC*** 0x00040007 Bad config descriptor  
***ERR\_USBD\_BAD\_INTF\_DESC*** 0x00040008 Bad interface descriptor  
***ERR\_USBD\_BAD\_EP\_DESC*** 0x00040009 Bad endpoint descriptor  
***ERR\_USBD\_BAD\_MEM\_BUF*** 0x0004000a Bad alignment of buffer passed.  
***ERR\_USBD\_TOO\_MANY\_CLASS\_HDLR*** 0x0004000b Too many class handlers.  
***ERR\_CGU\_BASE***  
***ERR\_CGU\_NOT\_IMPL***  
***ERR\_CGU\_INVALID\_PARAM***  
***ERR\_CGU\_INVALID\_SLICE***  
***ERR\_CGU\_OUTPUT\_GEN***  
***ERR\_CGU\_DIV\_SRC***  
***ERR\_CGU\_DIV\_VAL***  
***ERR\_CGU\_SRC***  
***ERR\_I2C\_BASE***  
***ERR\_I2C\_BUSY***  
***ERR\_I2C\_NAK***  
***ERR\_I2C\_BUFFER\_OVERFLOW***  
***ERR\_I2C\_BYTE\_COUNT\_ERR***  
***ERR\_I2C\_LOSS\_OF\_ARBRITRATION***  
***ERR\_I2C\_SLAVE\_NOT\_ADDRESSED***  
***ERR\_I2C\_LOSS\_OF\_ARBRITRATION\_NAK\_BIT***

*ERR\_I2C\_GENERAL\_FAILURE*  
*ERR\_I2C\_REGS\_SET\_TO\_DEFAULT*  
*ERR\_I2C\_TIMEOUT*  
*ERR\_I2C\_BUFFER\_UNDERFLOW*  
*ERR\_I2C\_PARAM*  
*ERR\_OTP\_BASE*  
*ERR\_OTP\_WR\_ENABLE\_INVALID*  
*ERR\_OTP\_SOME\_BITS\_ALREADY\_PROGRAMMED*  
*ERR\_OTP\_ALL\_DATA\_OR\_MASK\_ZERO*  
*ERR\_OTP\_WRITE\_ACCESS\_LOCKED*  
*ERR\_OTP\_READ\_DATA\_MISMATCH*  
*ERR\_OTP\_USB\_ID\_ENABLED*  
*ERR\_OTP\_ETH\_MAC\_ENABLED*  
*ERR\_OTP\_AES\_KEYS\_ENABLED*  
*ERR\_OTP\_ILLEGAL\_BANK*  
*ERR\_UART\_BASE*  
*ERR\_UART\_RXD\_BUSY*  
*ERR\_UART\_TXD\_BUSY*  
*ERR\_UART\_OVERRUN\_FRAME\_PARITY\_NOISE*  
*ERR\_UART\_UNDERRUN*  
*ERR\_UART\_PARAM*  
*ERR\_UART\_BAUDRATE*  
*ERR\_CAN\_BASE*  
*ERR\_CAN\_BAD\_MEM\_BUF*  
*ERR\_CAN\_INIT\_FAIL*  
*ERR\_CANOPEN\_INIT\_FAIL*  
*ERR\_SPIFI\_LITE\_BASE*  
*ERR\_SPIFI\_LITE\_INVALID\_ARGUMENTS*  
*ERR\_SPIFI\_LITE\_BUSY*  
*ERR\_SPIFI\_LITE\_MEMORY\_MODE\_ON*  
*ERR\_SPIFI\_LITE\_MEMORY\_MODE\_OFF*  
*ERR\_SPIFI\_LITE\_IN\_DMA*  
*ERR\_SPIFI\_LITE\_NOT\_IN\_DMA*  
*PENDING\_SPIFI\_LITE*  
*ERR\_CLK\_BASE*  
*ERR\_CLK\_NOT\_IMPL*  
*ERR\_CLK\_INVALID\_PARAM*  
*ERR\_CLK\_INVALID\_SLICE*  
*ERR\_CLK\_OUTPUT\_GEN*  
*ERR\_CLK\_DIV\_SRC*  
*ERR\_CLK\_DIV\_VAL*  
*ERR\_CLK\_SRC*  
*ERR\_CLK\_PLL\_FIN\_TOO\_SMALL*  
*ERR\_CLK\_PLL\_FIN\_TOO\_LARGE*  
*ERR\_CLK\_PLL\_FOUT\_TOO\_SMALL*

***ERR\_CLK\_PLL\_FOUT\_TOO\_LARGE***  
***ERR\_CLK\_PLL\_NO\_SOLUTION***  
***ERR\_CLK\_PLL\_MIN\_PCT***  
***ERR\_CLK\_PLL\_MAX\_PCT***  
***ERR\_CLK\_OSC\_FREQ***  
***ERR\_CLK\_CFG***  
***ERR\_CLK\_TIMEOUT***  
***ERR\_CLK\_BASE\_OFF***  
***ERR\_CLK\_OFF\_DEADLOCK***  
***ERR\_PWR\_BASE***  
***PWR\_ERROR\_ILLEGAL\_MODE***  
***PWR\_ERROR\_CLOCK\_FREQ\_TOO\_HIGH***  
***PWR\_ERROR\_INVALID\_STATE***  
***PWR\_ERROR\_INVALID\_CFG***  
***PWR\_ERROR\_PVT\_DETECT***  
***ERR\_DMA\_BASE***  
***ERR\_DMA\_ERROR\_INT***  
***ERR\_DMA\_CHANNEL\_NUMBER***  
***ERR\_DMA\_CHANNEL\_DISABLED***  
***ERR\_DMA\_BUSY***  
***ERR\_DMA\_NOT\_ALIGNMENT***  
***ERR\_DMA\_PING\_PONG\_EN***  
***ERR\_DMA\_CHANNEL\_VALID\_PENDING***  
***ERR\_DMA\_PARAM***  
***ERR\_DMA\_QUEUE\_EMPTY***  
***ERR\_DMA\_GENERAL***  
***ERR\_SPI\_BASE***  
***ERR\_SPI\_BUSY***  
***ERR\_SPI\_RXOVERRUN***  
***ERR\_SPI\_TXUNDERRUN***  
***ERR\_SPI\_SELNASSERT***  
***ERR\_SPI\_SELNDEASSERT***  
***ERR\_SPI\_CLKSTALL***  
***ERR\_SPI\_PARAM***  
***ERR\_SPI\_INVALID\_LENGTH***  
***ERR\_ADC\_BASE***  
***ERR\_ADC\_OVERRUN***  
***ERR\_ADC\_INVALID\_CHANNEL***  
***ERR\_ADC\_INVALID\_SEQUENCE***  
***ERR\_ADC\_INVALID\_SETUP***  
***ERR\_ADC\_PARAM***  
***ERR\_ADC\_INVALID\_LENGTH***  
***ERR\_ADC\_NO\_POWER***  
***ERR\_DM\_BASE***  
***ERR\_DM\_NOT\_ENTERED***  
***ERR\_DM\_UNKNOWN\_CMD***  
***ERR\_DM\_COMM\_FAIL***

Definition at line 46 of file error.h.

## 8.18 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/flexcomm\_5411x.h File Reference

```
#include "cmsis.h"
```

### Macros

- #define [FLEXCOMM\\_LOCK](#) (1 << 3)  
*FlexCOMM PSEL register bits.*
- #define [FLEXCOMM\\_ID\\_USART](#) (1 << 4)
- #define [FLEXCOMM\\_ID\\_SPI](#) (1 << 5)
- #define [FLEXCOMM\\_ID\\_I2C](#) (1 << 6)
- #define [FLEXCOMM\\_ID\\_I2S](#) (1 << 7)
- #define [ERR\\_FLEXCOMM\\_FUNCNOTSUPPORTED](#) -1  
*Flexcomm error and offset values.*
- #define [ERR\\_FLEXCOMM\\_NOTFREE](#) -2
- #define [ERR\\_FLEXCOMM\\_INVALIDBASE](#) -3
- #define [FLEXCOMM\\_PSEL\\_OFFSET](#) 0xFF8

### Typedefs

- typedef void [LPC\\_FLEXCOMM\\_T](#)

### Enumerations

- enum [FLEXCOMM\\_PERIPH\\_T](#) {  
[FLEXCOMM\\_PERIPH\\_NONE](#), [FLEXCOMM\\_PERIPH\\_USART](#), [FLEXCOMM\\_PERIPH\\_SPI](#), [FLEXCOMM\\_↵](#)  
[\\_PERIPH\\_I2C](#),  
[FLEXCOMM\\_PERIPH\\_I2S\\_TX](#), [FLEXCOMM\\_PERIPH\\_I2S\\_RX](#) }  
*FLEXCOMM Peripheral functions.*

### Functions

- [\\_\\_STATIC\\_INLINE FLEXCOMM\\_PERIPH\\_T Chip\\_FLEXCOMM\\_GetFunc](#) ([LPC\\_FLEXCOMM\\_T](#) \*pFCO↵  
MM)  
*Get currently enabled FLEXCOMM function.*
- [\\_\\_STATIC\\_INLINE int Chip\\_FLEXCOMM\\_IsLocked](#) ([LPC\\_FLEXCOMM\\_T](#) \*pFCOMM)  
*Checks if given FLEXCOMM is locked to a function.*
- [\\_\\_STATIC\\_INLINE void Chip\\_FLEXCOMM\\_Lock](#) ([LPC\\_FLEXCOMM\\_T](#) \*pFCOMM)  
*Lock FLEXCOMM to a function.*
- [int Chip\\_FLEXCOMM\\_SetPeriph](#) ([LPC\\_FLEXCOMM\\_T](#) \*pFCOMM, [FLEXCOMM\\_PERIPH\\_T](#) periph, int  
lock)  
*Set FLEXCOMM to a peripheral function.*
- [int Chip\\_FLEXCOMM\\_GetIndex](#) ([LPC\\_FLEXCOMM\\_T](#) \*pFCOMM)  
*Get index of the FLEXCOMM corresponding to the given base address.*
- [int Chip\\_FLEXCOMM\\_Init](#) ([LPC\\_FLEXCOMM\\_T](#) \*pFCOMM, [FLEXCOMM\\_PERIPH\\_T](#) periph)  
*Initialize FlexCOMM and associate it with a given peripheral.*
- [void Chip\\_FLEXCOMM\\_DeInit](#) ([LPC\\_FLEXCOMM\\_T](#) \*pFCOMM)  
*Uninitialize the FlexCOMM.*

## 8.19 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/fpu\_init.h File Reference

### Functions

- void [fpuInit](#) (void)  
*Early initialization of the FPU.*

## 8.20 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/gpio\_5411x.h File Reference

### Data Structures

- struct [LPC\\_GPIO\\_T](#)  
*GPIO port register block structure.*

### Functions

- `__STATIC_INLINE void Chip\_GPIO\_Init (LPC_GPIO_T *pGPIO)`  
*Initialize GPIO block.*
- `__STATIC_INLINE void Chip\_GPIO\_DeInit (LPC_GPIO_T *pGPIO)`  
*De-Initialize GPIO block.*
- `__STATIC_INLINE void Chip\_GPIO\_WritePortBit (LPC_GPIO_T *pGPIO, uint32_t port, uint8_t pin, bool setting)`  
*Set a GPIO port/pin state.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPinState (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin, bool setting)`  
*Set a GPIO pin state via the GPIO byte register.*
- `__STATIC_INLINE bool Chip\_GPIO\_ReadPortBit (LPC_GPIO_T *pGPIO, uint32_t port, uint8_t pin)`  
*Read a GPIO pin state via the GPIO byte register.*
- `__STATIC_INLINE bool Chip\_GPIO\_GetPinState (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`  
*Get a GPIO pin state via the GPIO byte register.*
- `__STATIC_INLINE void Chip\_GPIO\_WriteDirBit (LPC_GPIO_T *pGPIO, uint32_t port, uint8_t pin, bool setting)`  
*Set GPIO direction for a single GPIO pin.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPinDIROutput (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`  
*Set GPIO direction for a single GPIO pin to an output.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPinDIRInput (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`  
*Set GPIO direction for a single GPIO pin to an input.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPinDIR (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin, bool output)`  
*Set GPIO direction for a single GPIO pin.*
- `__STATIC_INLINE bool Chip\_GPIO\_ReadDirBit (LPC_GPIO_T *pGPIO, uint32_t port, uint8_t bit)`  
*Read a GPIO direction (out or in)*
- `__STATIC_INLINE bool Chip\_GPIO\_GetPinDIR (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`  
*Get GPIO direction for a single GPIO pin.*
- `__STATIC_INLINE void Chip\_GPIO\_SetDir (LPC_GPIO_T *pGPIO, uint8_t portNum, uint32_t bitValue, uint8_t out)`  
*Set Direction for a GPIO port.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPortDIROutput (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pin↔Mask)`  
*Set GPIO direction for a all selected GPIO pins to an output.*
- `__STATIC_INLINE void Chip\_GPIO\_SetPortDIRInput (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pin↔Mask)`

- Set GPIO direction for a all selected GPIO pins to an input.*
- `__STATIC_INLINE void Chip_GPIO_SetPortDIR (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pinMask, bool outSet)`
- Set GPIO direction for a all selected GPIO pins to an input or output.*
- `__STATIC_INLINE uint32_t Chip_GPIO_GetPortDIR (LPC_GPIO_T *pGPIO, uint8_t port)`
- Get GPIO direction for a all GPIO pins.*
- `__STATIC_INLINE void Chip_GPIO_SetPortMask (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t mask)`
- Set GPIO port mask value for GPIO masked read and write.*
- `__STATIC_INLINE uint32_t Chip_GPIO_GetPortMask (LPC_GPIO_T *pGPIO, uint8_t port)`
- Get GPIO port mask value used for GPIO masked read and write.*
- `__STATIC_INLINE void Chip_GPIO_SetPortValue (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t value)`
- Set all GPIO raw pin states (regardless of masking)*
- `__STATIC_INLINE uint32_t Chip_GPIO_GetPortValue (LPC_GPIO_T *pGPIO, uint8_t port)`
- Get all GPIO raw pin states (regardless of masking)*
- `__STATIC_INLINE void Chip_GPIO_SetMaskedPortValue (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t value)`
- Set all GPIO pin states, but mask via the MASKP0 register.*
- `__STATIC_INLINE uint32_t Chip_GPIO_GetMaskedPortValue (LPC_GPIO_T *pGPIO, uint8_t port)`
- Get all GPIO pin statesm but mask via the MASKP0 register.*
- `__STATIC_INLINE void Chip_GPIO_SetValue (LPC_GPIO_T *pGPIO, uint8_t portNum, uint32_t bitValue)`
- Set a GPIO port/bit to the high state.*
- `__STATIC_INLINE void Chip_GPIO_SetPortOutHigh (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pins)`
- Set selected GPIO output pins to the high state.*
- `__STATIC_INLINE void Chip_GPIO_SetPinOutHigh (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`
- Set an individual GPIO output pin to the high state.*
- `__STATIC_INLINE void Chip_GPIO_ClearValue (LPC_GPIO_T *pGPIO, uint8_t portNum, uint32_t bitValue)`
- Set a GPIO port/bit to the low state.*
- `__STATIC_INLINE void Chip_GPIO_SetPortOutLow (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pins)`
- Set selected GPIO output pins to the low state.*
- `__STATIC_INLINE void Chip_GPIO_SetPinOutLow (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`
- Set an individual GPIO output pin to the low state.*
- `__STATIC_INLINE void Chip_GPIO_SetPortToggle (LPC_GPIO_T *pGPIO, uint8_t port, uint32_t pins)`
- Toggle selected GPIO output pins to the opposite state.*
- `__STATIC_INLINE void Chip_GPIO_SetPinToggle (LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin)`
- Toggle an individual GPIO output pin to the opposite state.*
- `__STATIC_INLINE uint32_t Chip_GPIO_ReadValue (LPC_GPIO_T *pGPIO, uint8_t portNum)`
- Read current bit states for the selected port.*

## 8.21 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/gpiogroup\_5411x.h File Reference

### Data Structures

- struct `LPC_GPIOGROUPINT_T`  
*GPIO grouped interrupt register block structure.*

### Macros

- `#define GPIOGR_INT (1 << 0)`
- `#define GPIOGR_COMB (1 << 1)`
- `#define GPIOGR_TRIG (1 << 2)`

## Functions

- `__STATIC_INLINE void Chip_GPIOGP_Init (LPC_GPIOGROUPINT_T *pGPIOGPINT)`  
*Initialize GPIO group interrupt block.*
- `__STATIC_INLINE void Chip_GPIOGP_DeInit (LPC_GPIOGROUPINT_T *pGPIOGPINT)`  
*De-Initialize GPIO group interrupt block.*
- `__STATIC_INLINE void Chip_GPIOGP_ClearIntStatus (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)`  
*Clear interrupt pending status for the selected group.*
- `__STATIC_INLINE bool Chip_GPIOGP_GetIntStatus (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)`  
*Returns current GPIO group inetrrupt pending status.*
- `__STATIC_INLINE void Chip_GPIOGP_SelectOrMode (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)`  
*Selected GPIO group functionality for trigger on any pin in group (OR mode)*
- `__STATIC_INLINE void Chip_GPIOGP_SelectAndMode (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)`  
*Selected GPIO group functionality for trigger on all matching pins in group (AND mode)*
- `__STATIC_INLINE void Chip_GPIOGP_SelectEdgeMode (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)`  
*Selected GPIO group functionality edge trigger mode.*
- `__STATIC_INLINE void Chip_GPIOGP_SelectLevelMode (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group)`  
*Selected GPIO group functionality level trigger mode.*
- `__STATIC_INLINE void Chip_GPIOGP_SelectLowLevel (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)`  
*Set selected pins for the group and port to low level trigger.*
- `__STATIC_INLINE void Chip_GPIOGP_SelectHighLevel (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)`  
*Set selected pins for the group and port to high level trigger.*
- `__STATIC_INLINE void Chip_GPIOGP_DisableGroupPins (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)`  
*Disabled selected pins for the group interrupt.*
- `__STATIC_INLINE void Chip_GPIOGP_EnableGroupPins (LPC_GPIOGROUPINT_T *pGPIOGPINT, uint8_t group, uint8_t port, uint32_t pinMask)`  
*Enable selected pins for the group interrupt.*

## 8.22 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/i2c\_common\_5411x.h File Reference

### Macros

- `#define I2C_CFG_MSTEN (1 << 0)`
- `#define I2C_CFG_SLVEN (1 << 1)`
- `#define I2C_CFG_MONEN (1 << 2)`
- `#define I2C_CFG_TIMEOUTEN (1 << 3)`
- `#define I2C_CFG_MONCLKSTR (1 << 4)`
- `#define I2C_CFG_MASK ((uint32_t) 0x1F)`
- `#define I2C_STAT_MSTPENDING (1 << 0)`
- `#define I2C_STAT_MSTSTATE (0x7 << 1)`
- `#define I2C_STAT_MSTRARBLOSS (1 << 4)`
- `#define I2C_STAT_MSTSTSTPERR (1 << 6)`
- `#define I2C_STAT_SLVPENDING (1 << 8)`

- `#define I2C_STAT_SLVSTATE (0x3 << 9)`
- `#define I2C_STAT_SLVNOTSTR (1 << 11)`
- `#define I2C_STAT_SLVIDX (0x3 << 12)`
- `#define I2C_STAT_SLVSEL (1 << 14)`
- `#define I2C_STAT_SLVDESEL (1 << 15)`
- `#define I2C_STAT_MONRDY (1 << 16)`
- `#define I2C_STAT_MONOV (1 << 17)`
- `#define I2C_STAT_MONACTIVE (1 << 18)`
- `#define I2C_STAT_MONIDLE (1 << 19)`
- `#define I2C_STAT_EVENTTIMEOUT (1 << 24)`
- `#define I2C_STAT_SCLTIMEOUT (1 << 25)`
- `#define I2C_STAT_MSTCODE_IDLE (0)`
- `#define I2C_STAT_MSTCODE_RXREADY (1)`
- `#define I2C_STAT_MSTCODE_TXREADY (2)`
- `#define I2C_STAT_MSTCODE_NACKADR (3)`
- `#define I2C_STAT_MSTCODE_NACKDAT (4)`
- `#define I2C_STAT_SLVCODE_ADDR (0)`
- `#define I2C_STAT_SLVCODE_RX (1)`
- `#define I2C_STAT_SLVCODE_TX (2)`
- `#define I2C_INTENSET_MSTPENDING (1 << 0)`
- `#define I2C_INTENSET_MSTRARBLOSS (1 << 4)`
- `#define I2C_INTENSET_MSTSTSPERR (1 << 6)`
- `#define I2C_INTENSET_SLVPENDING (1 << 8)`
- `#define I2C_INTENSET_SLVNOTSTR (1 << 11)`
- `#define I2C_INTENSET_SLVDESEL (1 << 15)`
- `#define I2C_INTENSET_MONRDY (1 << 16)`
- `#define I2C_INTENSET_MONOV (1 << 17)`
- `#define I2C_INTENSET_MONIDLE (1 << 19)`
- `#define I2C_INTENSET_EVENTTIMEOUT (1 << 24)`
- `#define I2C_INTENSET_SCLTIMEOUT (1 << 25)`
- `#define I2C_INTENCLR_MSTPENDING (1 << 0)`
- `#define I2C_INTENCLR_MSTRARBLOSS (1 << 4)`
- `#define I2C_INTENCLR_MSTSTSPERR (1 << 6)`
- `#define I2C_INTENCLR_SLVPENDING (1 << 8)`
- `#define I2C_INTENCLR_SLVNOTSTR (1 << 11)`
- `#define I2C_INTENCLR_SLVDESEL (1 << 15)`
- `#define I2C_INTENCLR_MONRDY (1 << 16)`
- `#define I2C_INTENCLR_MONOV (1 << 17)`
- `#define I2C_INTENCLR_MONIDLE (1 << 19)`
- `#define I2C_INTENCLR_EVENTTIMEOUT (1 << 24)`
- `#define I2C_INTENCLR_SCLTIMEOUT (1 << 25)`
- `#define I2C_TIMEOUT_VAL(n) (((uint32_t) ((n) - 1) & 0xFFF0) | 0x000F)`
- `#define I2C_INTSTAT_MSTPENDING (1 << 0)`
- `#define I2C_INTSTAT_MSTRARBLOSS (1 << 4)`
- `#define I2C_INTSTAT_MSTSTSPERR (1 << 6)`
- `#define I2C_INTSTAT_SLVPENDING (1 << 8)`
- `#define I2C_INTSTAT_SLVNOTSTR (1 << 11)`
- `#define I2C_INTSTAT_SLVDESEL (1 << 15)`
- `#define I2C_INTSTAT_MONRDY (1 << 16)`
- `#define I2C_INTSTAT_MONOV (1 << 17)`
- `#define I2C_INTSTAT_MONIDLE (1 << 19)`
- `#define I2C_INTSTAT_EVENTTIMEOUT (1 << 24)`
- `#define I2C_INTSTAT_SCLTIMEOUT (1 << 25)`
- `#define I2C_MSTCTL_MSTCONTINUE (1 << 0)`
- `#define I2C_MSTCTL_MSTSTART (1 << 1)`



- #define I2C\_MSTCTL\_MSTSTOP (1 << 2)
- #define I2C\_MSTCTL\_MSTDMA (1 << 3)
- #define I2C\_MSTTIME\_MSTSCLOW (0x07 << 0)
- #define I2C\_MSTTIME\_MSTSCHIGH (0x07 << 4)
- #define I2C\_MSTDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)
- #define I2C\_SLVCTL\_SLVCONTINUE (1 << 0)
- #define I2C\_SLVCTL\_SLVNACK (1 << 1)
- #define I2C\_SLVCTL\_SLVDMA (1 << 3)
- #define I2C\_SLVDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)
- #define I2C\_SLVADR\_SADISABLE (1 << 0)
- #define I2C\_SLVADR\_SLVADR (0x7F << 1)
- #define I2C\_SLVADR\_MASK ((uint32\_t) 0x00FF)
- #define I2C\_SLVQUAL\_QUALMODE0 (1 << 0)
- #define I2C\_SLVQUAL\_SLVQUAL0 (0x7F << 1)
- #define I2C\_MONRXDAT\_DATA (0xFF << 0)
- #define I2C\_MONRXDAT\_MONSTART (1 << 8)
- #define I2C\_MONRXDAT\_MONRESTART (1 << 9)
- #define I2C\_MONRXDAT\_MONNACK (1 << 10)
- #define I2C\_CFG\_MSTEN (1 << 0)
- #define I2C\_CFG\_SLVEN (1 << 1)
- #define I2C\_CFG\_MONEN (1 << 2)
- #define I2C\_CFG\_TIMEOUTEN (1 << 3)
- #define I2C\_CFG\_MONCLKSTR (1 << 4)
- #define I2C\_CFG\_MASK ((uint32\_t) 0x1F)
- #define I2C\_STAT\_MSTPENDING (1 << 0)
- #define I2C\_STAT\_MSTSTATE (0x7 << 1)
- #define I2C\_STAT\_MSTRARBLOSS (1 << 4)
- #define I2C\_STAT\_MSTSTSTPERR (1 << 6)
- #define I2C\_STAT\_SLVPENDING (1 << 8)
- #define I2C\_STAT\_SLVSTATE (0x3 << 9)
- #define I2C\_STAT\_SLVNOTSTR (1 << 11)
- #define I2C\_STAT\_SLVIDX (0x3 << 12)
- #define I2C\_STAT\_SLVSEL (1 << 14)
- #define I2C\_STAT\_SLVDESEL (1 << 15)
- #define I2C\_STAT\_MONRDY (1 << 16)
- #define I2C\_STAT\_MONOV (1 << 17)
- #define I2C\_STAT\_MONACTIVE (1 << 18)
- #define I2C\_STAT\_MONIDLE (1 << 19)
- #define I2C\_STAT\_EVENTTIMEOUT (1 << 24)
- #define I2C\_STAT\_SCLTIMEOUT (1 << 25)
- #define I2C\_STAT\_MSTCODE\_IDLE (0)
- #define I2C\_STAT\_MSTCODE\_RXREADY (1)
- #define I2C\_STAT\_MSTCODE\_TXREADY (2)
- #define I2C\_STAT\_MSTCODE\_NACKADR (3)
- #define I2C\_STAT\_MSTCODE\_NACKDAT (4)
- #define I2C\_STAT\_SLVCODE\_ADDR (0)
- #define I2C\_STAT\_SLVCODE\_RX (1)
- #define I2C\_STAT\_SLVCODE\_TX (2)
- #define I2C\_INTENSET\_MSTPENDING (1 << 0)
- #define I2C\_INTENSET\_MSTRARBLOSS (1 << 4)
- #define I2C\_INTENSET\_MSTSTSTPERR (1 << 6)
- #define I2C\_INTENSET\_SLVPENDING (1 << 8)
- #define I2C\_INTENSET\_SLVNOTSTR (1 << 11)
- #define I2C\_INTENSET\_SLVDESEL (1 << 15)
- #define I2C\_INTENSET\_MONRDY (1 << 16)

- #define I2C\_INTENSET\_MONOV (1 << 17)
- #define I2C\_INTENSET\_MONIDLE (1 << 19)
- #define I2C\_INTENSET\_EVENTTIMEOUT (1 << 24)
- #define I2C\_INTENSET\_SCLTIMEOUT (1 << 25)
- #define I2C\_INTENCLR\_MSTPENDING (1 << 0)
- #define I2C\_INTENCLR\_MSTRARBLOSS (1 << 4)
- #define I2C\_INTENCLR\_MSTSTSPERR (1 << 6)
- #define I2C\_INTENCLR\_SLVPENDING (1 << 8)
- #define I2C\_INTENCLR\_SLVNOTSTR (1 << 11)
- #define I2C\_INTENCLR\_SLVDESEL (1 << 15)
- #define I2C\_INTENCLR\_MONRDY (1 << 16)
- #define I2C\_INTENCLR\_MONOV (1 << 17)
- #define I2C\_INTENCLR\_MONIDLE (1 << 19)
- #define I2C\_INTENCLR\_EVENTTIMEOUT (1 << 24)
- #define I2C\_INTENCLR\_SCLTIMEOUT (1 << 25)
- #define I2C\_TIMEOUT\_VAL(n) (((uint32\_t) ((n) - 1) & 0xFFF0) | 0x000F)
- #define I2C\_INTSTAT\_MSTPENDING (1 << 0)
- #define I2C\_INTSTAT\_MSTRARBLOSS (1 << 4)
- #define I2C\_INTSTAT\_MSTSTSPERR (1 << 6)
- #define I2C\_INTSTAT\_SLVPENDING (1 << 8)
- #define I2C\_INTSTAT\_SLVNOTSTR (1 << 11)
- #define I2C\_INTSTAT\_SLVDESEL (1 << 15)
- #define I2C\_INTSTAT\_MONRDY (1 << 16)
- #define I2C\_INTSTAT\_MONOV (1 << 17)
- #define I2C\_INTSTAT\_MONIDLE (1 << 19)
- #define I2C\_INTSTAT\_EVENTTIMEOUT (1 << 24)
- #define I2C\_INTSTAT\_SCLTIMEOUT (1 << 25)
- #define I2C\_MSTCTL\_MSTCONTINUE (1 << 0)
- #define I2C\_MSTCTL\_MSTSTART (1 << 1)
- #define I2C\_MSTCTL\_MSTSTOP (1 << 2)
- #define I2C\_MSTCTL\_MSTDMA (1 << 3)
- #define I2C\_MSTTIME\_MSTSCLLOW (0x07 << 0)
- #define I2C\_MSTTIME\_MSTSCLHIGH (0x07 << 4)
- #define I2C\_MSTDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)
- #define I2C\_SLVCTL\_SLVCONTINUE (1 << 0)
- #define I2C\_SLVCTL\_SLVNACK (1 << 1)
- #define I2C\_SLVCTL\_SLVDMA (1 << 3)
- #define I2C\_SLVDAT\_DATAMASK ((uint32\_t) 0x00FF << 0)
- #define I2C\_SLVADR\_SADISABLE (1 << 0)
- #define I2C\_SLVADR\_SLVADR (0x7F << 1)
- #define I2C\_SLVADR\_MASK ((uint32\_t) 0x00FF)
- #define I2C\_SLVQUAL\_QUALMODE0 (1 << 0)
- #define I2C\_SLVQUAL\_SLVQUAL0 (0x7F << 1)
- #define I2C\_MONRXDAT\_DATA (0xFF << 0)
- #define I2C\_MONRXDAT\_MONSTART (1 << 8)
- #define I2C\_MONRXDAT\_MONRESTART (1 << 9)
- #define I2C\_MONRXDAT\_MONNACK (1 << 10)

## Functions

- `__STATIC_INLINE int Chip_I2C_Init (LPC_I2C_T *pI2C)`  
*Initialize I2C Interface.*
- `__STATIC_INLINE void Chip_I2C_DeInit (LPC_I2C_T *pI2C)`  
*Shutdown I2C Interface.*
- `__STATIC_INLINE void Chip_I2C_SetClockDiv (LPC_I2C_T *pI2C, uint32_t clkdiv)`  
*Sets I2C Clock Divider registers.*
- `static INLINE uint32_t Chip_I2C_GetClockDiv (LPC_I2C_T *pI2C)`  
*Get I2C Clock Divider registers.*
- `static INLINE void Chip_I2C_EnableInt (LPC_I2C_T *pI2C, uint32_t intEn)`  
*Enable I2C Interrupts.*
- `static INLINE void Chip_I2C_DisableInt (LPC_I2C_T *pI2C, uint32_t intClr)`  
*Disable I2C Interrupts.*
- `static INLINE void Chip_I2C_ClearInt (LPC_I2C_T *pI2C, uint32_t intClr)`  
*Disable I2C Interrupts.*
- `static INLINE uint32_t Chip_I2C_GetPendingInt (LPC_I2C_T *pI2C)`  
*Returns pending I2C Interrupts.*

## Variables

- `__IO uint32_t STAT`  
*I2C register block structure.*
- `__IO uint32_t INTENSET`
- `__O uint32_t INTENCLR`
- `__IO uint32_t TIMEOUT`
- `__IO uint32_t CLKDIV`
- `__IO uint32_t INTSTAT`
- `__I uint32_t RESERVED0`
- `__IO uint32_t MSTCTL`
- `__IO uint32_t MSTTIME`
- `__IO uint32_t MSTDAT`
- `__IO uint32_t RESERVED1 [5]`
- `__IO uint32_t SLVCTL`
- `__IO uint32_t SLVDAT`
- `__IO uint32_t SLVADR [4]`
- `__IO uint32_t SLVQUAL0`
- `__IO uint32_t RESERVED2 [9]`
- `__I uint32_t MONRXDAT`
- `__IO uint32_t PSELID`
- `__I uint32_t PID`
- `LPC_I2C_T`

## 8.23 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/i2cm\_5411x.h File Reference

```
#include "chip.h"
```

## Data Structures

- `struct I2CM_XFER_T`  
*Master transfer data structure definitions.*

## Macros

- `#define I2CM_STATUS_OK 0x00`
- `#define I2CM_STATUS_ERROR 0x01`
- `#define I2CM_STATUS_NAK_ADR 0x02`
- `#define I2CM_STATUS_BUS_ERROR 0x03`
- `#define I2CM_STATUS_NAK_DAT 0x04`
- `#define I2CM_STATUS_ARBLOST 0x05`
- `#define I2CM_STATUS_BUSY 0xFF`

## Functions

- static `INLINE` void `Chip_I2CM_SetDutyCycle` (`LPC_I2C_T *pI2C`, `uint16_t sclH`, `uint16_t sclL`)  
*Sets HIGH and LOW duty cycle registers.*
- void `Chip_I2CM_SetBusSpeed` (`LPC_I2C_T *pI2C`, `uint32_t busSpeed`)  
*Set up bus speed for LPC\_I2C controller.*
- static `INLINE` void `Chip_I2CM_Enable` (`LPC_I2C_T *pI2C`)  
*Enable I2C Master interface.*
- static `INLINE` void `Chip_I2CM_Disable` (`LPC_I2C_T *pI2C`)  
*Disable I2C Master interface.*
- static `INLINE` `uint32_t` `Chip_I2CM_GetStatus` (`LPC_I2C_T *pI2C`)  
*Get I2C Status.*
- static `INLINE` void `Chip_I2CM_ClearStatus` (`LPC_I2C_T *pI2C`, `uint32_t clrStatus`)  
*Clear I2C status bits (master)*
- static `INLINE` bool `Chip_I2CM_IsMasterPending` (`LPC_I2C_T *pI2C`)  
*Check if I2C Master is pending.*
- static `INLINE` `uint32_t` `Chip_I2CM_GetMasterState` (`LPC_I2C_T *pI2C`)  
*Get current state of the I2C Master.*
- static `INLINE` void `Chip_I2CM_SendStart` (`LPC_I2C_T *pI2C`)  
*Transmit START or Repeat-START signal on I2C bus.*
- static `INLINE` void `Chip_I2CM_SendStop` (`LPC_I2C_T *pI2C`)  
*Transmit STOP signal on I2C bus.*
- static `INLINE` void `Chip_I2CM_MasterContinue` (`LPC_I2C_T *pI2C`)  
*Master Continue transfer operation.*
- static `INLINE` void `Chip_I2CM_WriteByte` (`LPC_I2C_T *pI2C`, `uint8_t data`)  
*Transmit a single data byte through the I2C peripheral (master)*
- static `INLINE` `uint8_t` `Chip_I2CM_ReadByte` (`LPC_I2C_T *pI2C`)  
*Read a single byte data from the I2C peripheral (master)*
- `uint32_t` `Chip_I2CM_XferHandler` (`LPC_I2C_T *pI2C`, `I2CM_XFER_T *xfer`)  
*Transfer state change handler.*
- void `Chip_I2CM_Xfer` (`LPC_I2C_T *pI2C`, `I2CM_XFER_T *xfer`)  
*Transmit and Receive data in master mode.*
- `uint32_t` `Chip_I2CM_XferBlocking` (`LPC_I2C_T *pI2C`, `I2CM_XFER_T *xfer`)  
*Transmit and Receive data in master mode.*

## 8.24 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/i2cs\_5411x.h File Reference

```
#include "i2c_common_5411x.h"
```

## Data Structures

- struct [I2CS\\_XFER\\_T](#)

## Typedefs

- typedef void(\* [I2CSlaveXferStart](#)) (uint8\_t addr)  
*I2C slave service start callback This callback is called from the I2C slave handler when an I2C slave address is received and needs servicing. It's used to indicate the start of a slave transfer that will happen on the slave bus.*
- typedef uint8\_t(\* [I2CSlaveXferSend](#)) (uint8\_t \*data)  
*I2C slave send data callback This callback is called from the I2C slave handler when an I2C slave address needs data to send.  
If you want to NAK the master, return I2C\_SLVCTL\_SLVNACK to the caller. Return I2C\_SLVCTL\_SLVCONTINUE or 0 to the caller for normal non-DMA data transfer. If you've setup a DMA descriptor for the transfer, return I2C\_SLVCTL\_SLVDMA to the caller.*
- typedef uint8\_t(\* [I2CSlaveXferRecv](#)) (uint8\_t data)  
*I2C slave receive data callback This callback is called from the I2C slave handler when an I2C slave address has receive data.  
If you want to NAK the master, return I2C\_SLVCTL\_SLVNACK to the caller. Return I2C\_SLVCTL\_SLVCONTINUE or 0 to the caller for normal non-DMA data transfer. If you've setup a DMA descriptor for the transfer, return I2C\_SLVCTL\_SLVDMA to the caller.*
- typedef void(\* [I2CSlaveXferDone](#)) (void)  
*I2C slave service done callback This callback is called from the I2C slave handler when an I2C slave transfer is completed. It's used to indicate the end of a slave transfer.*

## Functions

- \_\_STATIC\_INLINE void [Chip\\_I2CS\\_Enable](#) (LPC\_I2C\_T \*pI2C)  
*Enable I2C slave interface.*
- \_\_STATIC\_INLINE void [Chip\\_I2CS\\_Disable](#) (LPC\_I2C\_T \*pI2C)  
*Disable I2C slave interface.*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_I2CS\\_GetStatus](#) (LPC\_I2C\_T \*pI2C)  
*Get I2C Status.*
- \_\_STATIC\_INLINE void [Chip\\_I2CS\\_ClearStatus](#) (LPC\_I2C\_T \*pI2C, uint32\_t clrStatus)  
*Clear I2C status bits (slave)*
- \_\_STATIC\_INLINE bool [Chip\\_I2CS\\_IsSlavePending](#) (LPC\_I2C\_T \*pI2C)  
*Check if I2C slave is pending.*
- \_\_STATIC\_INLINE bool [Chip\\_I2CS\\_IsSlaveSelected](#) (LPC\_I2C\_T \*pI2C)  
*Check if I2C slave is selected.*
- \_\_STATIC\_INLINE bool [Chip\\_I2CS\\_IsSlaveDeSelected](#) (LPC\_I2C\_T \*pI2C)  
*Check if I2C slave is deselected.*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_I2CS\\_GetSlaveState](#) (LPC\_I2C\_T \*pI2C)  
*Get current state of the I2C slave.*
- \_\_STATIC\_INLINE uint32\_t [Chip\\_I2CS\\_GetSlaveMatchIndex](#) (LPC\_I2C\_T \*pI2C)  
*Returns the current slave address match index.*
- \_\_STATIC\_INLINE void [Chip\\_I2CS\\_SlaveContinue](#) (LPC\_I2C\_T \*pI2C)  
*Slave Continue transfer operation (ACK)*
- \_\_STATIC\_INLINE void [Chip\\_I2CS\\_SlaveNACK](#) (LPC\_I2C\_T \*pI2C)  
*Slave NACK operation.*
- \_\_STATIC\_INLINE void [Chip\\_I2CS\\_SlaveEnableDMA](#) (LPC\_I2C\_T \*pI2C)  
*Enable slave DMA operation.*

- `__STATIC_INLINE void Chip_I2CS_SlaveDisableDMA (LPC_I2C_T *pI2C)`  
*Disable slave DMA operation.*
- `__STATIC_INLINE void Chip_I2CS_WriteByte (LPC_I2C_T *pI2C, uint8_t data)`  
*Transmit a single data byte through the I2C peripheral (slave)*
- `__STATIC_INLINE uint8_t Chip_I2CS_ReadByte (LPC_I2C_T *pI2C)`  
*Read a single byte data from the I2C peripheral (slave)*
- `__STATIC_INLINE void Chip_I2CS_SetSlaveAddr (LPC_I2C_T *pI2C, uint8_t slvNum, uint8_t slvAddr)`  
*Set a I2C slave address for slave operation.*
- `__STATIC_INLINE uint8_t Chip_I2CS_GetSlaveAddr (LPC_I2C_T *pI2C, uint8_t slvNum)`  
*Return a I2C programmed slave address.*
- `__STATIC_INLINE void Chip_I2CS_EnableSlaveAddr (LPC_I2C_T *pI2C, uint8_t slvNum)`  
*Enable a I2C address.*
- `__STATIC_INLINE void Chip_I2CS_DisableSlaveAddr (LPC_I2C_T *pI2C, uint8_t slvNum)`  
*Disable a I2C address.*
- `__STATIC_INLINE void Chip_I2CS_SetSlaveQual0 (LPC_I2C_T *pI2C, bool extend, uint8_t slvAddr)`  
*Setup slave qialifier address.*
- `uint32_t Chip_I2CS_XferHandler (LPC_I2C_T *pI2C, const I2CS_XFER_T *xfers)`  
*Slave transfer state change handler.*

## 8.25 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/i2s\_5411x.h File Reference

### Data Structures

- struct `LPC_I2S_T`  
*I2S register block structure.*
- struct `I2S_AUDIO_FORMAT_T`
- struct `I2S_STATISTICS_T`  
*I2S statistics structure.*

### Macros

- `#define I2S_CFG1_MAINENABLE (0x01 << 0)`  
*I2S CFG1 register bits.*
- `#define I2S_CFG1_DATAPAUSE (0x01 << 1)`
- `#define I2S_CFG1_2NDCOUNT(p) (((p) & 0x03) << 2)`
- `#define I2S_CFG1_MSTSLVCFG(p) (((p) & 0x03) << 4)`
- `#define I2S_CFG1_MODE(p) (((p) & 0x03) << 6)`
- `#define I2S_CFG1_RIGHTLOW (0x01 << 8)`
- `#define I2S_CFG1_LEFTJUST (0x01 << 9)`
- `#define I2S_CFG1_ONECHANNEL (0x01 << 10)`
- `#define I2S_CFG1_PDMDATA (0x01 << 11)`
- `#define I2S_CFG1_SCK_POL (0x01 << 12)`
- `#define I2S_CFG1_WS_POL (0x01 << 13)`
- `#define I2S_CFG1_DATALEN(p) (((p) & 0x1f) << 16)`
- `#define I2S_CFG2_FRAMELEN(p) (((p) & 0x1ff) << 0)`  
*I2S CFG2 register bits.*
- `#define I2S_CFG2_POSITION(p) (((p) & 0x1ff) << 16)`
- `#define I2S_STAT_BUSY (0x01 << 0)`  
*I2S status register bits.*
- `#define I2S_STAT_SLVFRMERR (0x01 << 1)`
- `#define I2S_STAT_LR (0x01 << 2)`

- #define `I2S_STAT_PAUSED` (0x01 << 3)
- #define `I2S_FIFO_CFG_ENABLETX` (0x01 << 0)
- I2S FIFO configuration bits.*
- #define `I2S_FIFO_CFG_ENABLERX` (0x01 << 1)
- #define `I2S_FIFO_CFG_TXI2SE0` (0x01 << 2)
- #define `I2S_FIFO_CFG_PACK48` (0x01 << 3)
- #define `I2S_FIFO_CFG_SIZE(p)` (((p) & 0x03) << 4)
- #define `I2S_FIFO_CFG_DMATX` (0x01 << 12)
- #define `I2S_FIFO_CFG_DMARX` (0x01 << 13)
- #define `I2S_FIFO_CFG_WAKETX` (0x01 << 14)
- #define `I2S_FIFO_CFG_WAKERX` (0x01 << 15)
- #define `I2S_FIFO_CFG_EMPTYTX` (0x01 << 16)
- #define `I2S_FIFO_CFG_EMPTYRX` (0x01 << 17)
- #define `I2S_FIFO_CFG_POPDBG` (0x01 << 18)
- #define `I2S_FIFO_STAT_TXERR` (0x01 << 0)
- I2S FIFO status bits.*
- #define `I2S_FIFO_STAT_RXERR` (0x01 << 1)
- #define `I2S_FIFO_STAT_PERINT` (0x01 << 3)
- #define `I2S_FIFO_STAT_TXEMPTY` (0x01 << 4)
- #define `I2S_FIFO_STAT_TXNOTFULL` (0x01 << 5)
- #define `I2S_FIFO_STAT_RXNOTEMPTY` (0x01 << 6)
- #define `I2S_FIFO_STAT_RXFULL` (0x01 << 7)
- #define `I2S_FIFO_STAT_TXLVL(p)` (((p) & 0x0f) << 8)
- #define `I2S_FIFO_STAT_RXLVL(p)` (((p) & 0x0f) << 16)
- #define `I2S_FIFO_TRIG_TXLVLENA` (0x01 << 0)
- I2S FIFO trigger settings bits.*
- #define `I2S_FIFO_TRIG_RXLVLENA` (0x01 << 1)
- #define `I2S_FIFO_TRIG_TXLVL(p)` (((p) & 0x0f) << 8)
- #define `I2S_FIFO_TRIG_RXLVL(p)` (((p) & 0x0f) << 16)
- #define `I2S_FIFO_INT_BITMASK` (0x001F) /\*\* FIFO interrupt Bit mask \*/
- I2S FIFO interrupt enable. Used for interrupt status too.*
- #define `I2S_FIFO_INT_TXERR` (0x01 << 0)
- #define `I2S_FIFO_INT_RXERR` (0x01 << 1)
- #define `I2S_FIFO_INT_TXLVL` (0x01 << 2)
- #define `I2S_FIFO_INT_RXLVL` (0x01 << 3)
- #define `I2S_FIFO_INT_PERINT` (0x01 << 4)
- #define `LPC_I2S6_BASE` \_\_APPEND3(LPC\_FLEXCOMM,I2S6\_FLEXCOMM,\_BASE)
- #define `LPC_I2S6` ((LPC\_I2S\_T \*) LPC\_I2S6\_BASE)
- #define `I2S6_IRQHandler` \_\_APPEND3(FLEXCOMM,I2S6\_FLEXCOMM,\_IRQHandler)
- #define `I2S6_IRQn` \_\_APPEND3(FLEXCOMM,I2S6\_FLEXCOMM,\_IRQn)
- #define `DMAREQ_I2S6_RX` \_\_APPEND3(DMAREQ\_FLEXCOMM,I2S6\_FLEXCOMM,\_RX)
- #define `DMAREQ_I2S6_TX` \_\_APPEND3(DMAREQ\_FLEXCOMM,I2S6\_FLEXCOMM,\_TX)
- #define `LPC_I2S7_BASE` \_\_APPEND3(LPC\_FLEXCOMM,I2S7\_FLEXCOMM,\_BASE)
- #define `LPC_I2S7` ((LPC\_I2S\_T \*) LPC\_I2S7\_BASE)
- #define `I2S7_IRQHandler` \_\_APPEND3(FLEXCOMM,I2S7\_FLEXCOMM,\_IRQHandler)
- #define `I2S7_IRQn` \_\_APPEND3(FLEXCOMM,I2S7\_FLEXCOMM,\_IRQn)
- #define `DMAREQ_I2S7_RX` \_\_APPEND3(DMAREQ\_FLEXCOMM,I2S7\_FLEXCOMM,\_RX)
- #define `DMAREQ_I2S7_TX` \_\_APPEND3(DMAREQ\_FLEXCOMM,I2S7\_FLEXCOMM,\_TX)

## Enumerations

- enum `I2S_FIFO_CMD_T` {  
`I2S_FIFO_ENABLE`, `I2S_FIFO_DISABLE`, `I2S_FIFO_DMA_ENABLE`, `I2S_FIFO_DMA_DISABLE`,  
`I2S_FIFO_CLEAR`, `I2S_FIFO_TXZ_ENABLE`, `I2S_FIFO_TXZ_DISABLE` }
- enum `I2S_DIR_T` { `I2S_TX`, `I2S_RX` }
- enum `I2S_MSTSLVCFG_T` { `NORMAL_SLAVE`, `WS_SYNC_MASTER`, `EXT_SCLCK_MASTER`, `NORMAL_MASTER` }
- enum `I2S_MODE_T` { `I2S_CLASSIC`, `DSP_WS_50`, `DSP_WS_SHORT`, `DSP_WS_LONG` }

## Functions

- int `Chip_I2S_Init` (`LPC_I2S_T` \*pl2S, `I2S_AUDIO_FORMAT_T` \*fmt)  
Initialize I2S driver.
- \_\_STATIC\_INLINE int `Chip_I2S_TX_Init` (`LPC_I2S_T` \*pl2S)
- \_\_STATIC\_INLINE int `Chip_I2S_RX_Init` (`LPC_I2S_T` \*pl2S)
- \_\_STATIC\_INLINE void `Chip_I2S_DeInit` (`LPC_I2S_T` \*pl2S)  
Shutdown I2S driver.
- `Status` `Chip_I2S_Config` (`LPC_I2S_T` \*pl2S, `I2S_AUDIO_FORMAT_T` \*fmt)  
Configure I2S Port.
- \_\_STATIC\_INLINE uint32\_t `Chip_I2S_GetStatus` (`LPC_I2S_T` \*pl2S)  
Get the I2S status register.
- \_\_STATIC\_INLINE void `Chip_I2S_ClearStatus` (`LPC_I2S_T` \*pl2S, uint32\_t stsMask)  
Clear the I2S status register.
- \_\_STATIC\_INLINE void `Chip_I2S_Send` (`LPC_I2S_T` \*pl2S, uint32\_t data)  
Send a 32-bit data to TXFIFO for transmission.
- \_\_STATIC\_INLINE uint32\_t `Chip_I2S_Receive` (`LPC_I2S_T` \*pl2S)  
Get received data from RXFIFO.
- \_\_STATIC\_INLINE void `Chip_I2S_Start` (`LPC_I2S_T` \*pl2S)  
Start I2S port.
- \_\_STATIC\_INLINE void `Chip_I2S_Stop` (`LPC_I2S_T` \*pl2S)  
Stop I2S asynchronously.
- \_\_STATIC\_INLINE void `Chip_I2S_Pause` (`LPC_I2S_T` \*pl2S)  
Pause the I2S port.
- \_\_STATIC\_INLINE void `Chip_I2S_Play` (`LPC_I2S_T` \*pl2S)  
Play (un-Pause) the I2S port.
- void `Chip_I2S_FIFO_Config` (`LPC_I2S_T` \*pl2S, `I2S_AUDIO_FORMAT_T` \*fmt)  
Configure I2S FIFO.
- void `Chip_I2S_FIFO_Control` (`LPC_I2S_T` \*pl2S, `I2S_AUDIO_FORMAT_T` \*fmt, `I2S_FIFO_CMD_T` cmd)  
Execute I2S FIFO control commands.
- \_\_STATIC\_INLINE uint32\_t `Chip_I2S_GetFIFOStatus` (`LPC_I2S_T` \*pl2S)  
Get FIFO status.
- \_\_STATIC\_INLINE void `Chip_I2S_ClrFIFOStatus` (`LPC_I2S_T` \*pl2S, uint32\_t mask)  
Clear FIFO status.
- \_\_STATIC\_INLINE void `Chip_I2S_SetFIFOTrigLevel` (`LPC_I2S_T` \*pl2S, uint8\_t tx\_lvl, uint8\_t rx\_lvl)  
Setup I2S FIFO trigger-level.
- \_\_STATIC\_INLINE uint32\_t `Chip_I2S_GetFIFOTrigLevel` (`LPC_I2S_T` \*pl2S)  
Get I2S FIFO trigger-level.
- \_\_STATIC\_INLINE uint8\_t `Chip_I2S_GetFIFOTxLevel` (`LPC_I2S_T` \*pl2S)  
Get the current level of the Transmit FIFO.
- \_\_STATIC\_INLINE uint8\_t `Chip_I2S_GetFIFORxLevel` (`LPC_I2S_T` \*pl2S)



- Get the current level of the Receive FIFO.*
- `__STATIC_INLINE void Chip_I2S_FIFO_SetInterrupt (LPC_I2S_T *pI2S, uint32_t int_val)`  
*Set I2S FIFO interrupts.*
- `__STATIC_INLINE void Chip_I2S_FIFO_ClrInterrupt (LPC_I2S_T *pI2S, uint32_t int_val)`  
*Clear I2S FIFO interrupts.*
- `__STATIC_INLINE uint32_t Chip_I2S_FIFO_GetPendingInts (LPC_I2S_T *pI2S)`  
*Get I2S FIFO interrupts status.*
- `__STATIC_INLINE void Chip_I2S_FIFO_ClearStatus (LPC_I2S_T *pI2S, uint32_t mask)`  
*Clear the FIFO status register.*
- `void Chip_I2S_ErrorHandler (LPC_I2S_T *pI2S, I2S_STATISTICS_T *stat)`  
*I2S error handler.*

## 8.25.1 Macro Definition Documentation

### 8.25.1.1 `#define DMAREQ_I2S6_RX __APPEND3(DMAREQ_FLEXCOMM,I2S6_FLEXCOMM,_RX)`

Definition at line 506 of file i2s\_5411x.h.

### 8.25.1.2 `#define DMAREQ_I2S6_TX __APPEND3(DMAREQ_FLEXCOMM,I2S6_FLEXCOMM,_TX)`

Definition at line 507 of file i2s\_5411x.h.

### 8.25.1.3 `#define DMAREQ_I2S7_RX __APPEND3(DMAREQ_FLEXCOMM,I2S7_FLEXCOMM,_RX)`

Definition at line 514 of file i2s\_5411x.h.

### 8.25.1.4 `#define DMAREQ_I2S7_TX __APPEND3(DMAREQ_FLEXCOMM,I2S7_FLEXCOMM,_TX)`

Definition at line 515 of file i2s\_5411x.h.

### 8.25.1.5 `#define I2S6_IRQHandler __APPEND3(FLEXCOMM,I2S6_FLEXCOMM,_IRQHandler)`

Definition at line 504 of file i2s\_5411x.h.

### 8.25.1.6 `#define I2S6_IRQn __APPEND3(FLEXCOMM,I2S6_FLEXCOMM,_IRQn)`

Definition at line 505 of file i2s\_5411x.h.

### 8.25.1.7 `#define I2S7_IRQHandler __APPEND3(FLEXCOMM,I2S7_FLEXCOMM,_IRQHandler)`

Definition at line 512 of file i2s\_5411x.h.

### 8.25.1.8 `#define I2S7_IRQn __APPEND3(FLEXCOMM,I2S7_FLEXCOMM,_IRQn)`

Definition at line 513 of file i2s\_5411x.h.

### 8.25.1.9 `#define I2S_CFG1_2NDCOUNT( p ) (((p) & 0x03) << 2)`

additional I2S channel pair ct (read only)

Definition at line 80 of file i2s\_5411x.h.

**8.25.1.10** `#define I2S_CFG1_DATALEN( p ) (((p) & 0x1f) << 16)`

Data Length (minus 1 encoded)

Definition at line 89 of file i2s\_5411x.h.

**8.25.1.11** `#define I2S_CFG1_DATAPAUSE (0x01 << 1)`

Data flow Pause

Definition at line 79 of file i2s\_5411x.h.

**8.25.1.12** `#define I2S_CFG1_LEFTJUST (0x01 << 9)`

Left Justify data

Definition at line 84 of file i2s\_5411x.h.

**8.25.1.13** `#define I2S_CFG1_MAINENABLE (0x01 << 0)`

I2S CFG1 register bits.

I2S Main enable

Definition at line 78 of file i2s\_5411x.h.

**8.25.1.14** `#define I2S_CFG1_MODE( p ) (((p) & 0x03) << 6)`

Selects basic I2S operating mode

Definition at line 82 of file i2s\_5411x.h.

**8.25.1.15** `#define I2S_CFG1_MSTSLVCFG( p ) (((p) & 0x03) << 4)`

Master / slave configuration selection

Definition at line 81 of file i2s\_5411x.h.

**8.25.1.16** `#define I2S_CFG1_ONECHANNEL (0x01 << 10)`

Single channel mode

Definition at line 85 of file i2s\_5411x.h.

**8.25.1.17** `#define I2S_CFG1_PDMDATA (0x01 << 11)`

PDM Data selection

Definition at line 86 of file i2s\_5411x.h.

**8.25.1.18** `#define I2S_CFG1_RIGHTLOW (0x01 << 8)`

Right channel data is in the Low portion of FIFO data

Definition at line 83 of file i2s\_5411x.h.

8.25.1.19 `#define I2S_CFG1_SCK_POL (0x01 << 12)`

SCK polarity

Definition at line 87 of file i2s\_5411x.h.

8.25.1.20 `#define I2S_CFG1_WS_POL (0x01 << 13)`

WS polarity

Definition at line 88 of file i2s\_5411x.h.

8.25.1.21 `#define I2S_CFG2_FRAMELEN( p ) (((p) & 0x1ff) << 0)`

I2S CFG2 register bits.

I2S Main enable

Definition at line 94 of file i2s\_5411x.h.

8.25.1.22 `#define I2S_CFG2_POSITION( p ) (((p) & 0x1ff) << 16)`

Data flow Pause

Definition at line 95 of file i2s\_5411x.h.

8.25.1.23 `#define I2S_FIFO_CFG_DMARX (0x01 << 13)`

DMA configuration for receive

Definition at line 114 of file i2s\_5411x.h.

8.25.1.24 `#define I2S_FIFO_CFG_DMATX (0x01 << 12)`

DMA configuration for transmit

Definition at line 113 of file i2s\_5411x.h.

8.25.1.25 `#define I2S_FIFO_CFG_EMPTYRX (0x01 << 17)`

Empty command for the receive FIFO

Definition at line 118 of file i2s\_5411x.h.

8.25.1.26 `#define I2S_FIFO_CFG_EMPTYTX (0x01 << 16)`

Empty command for the transmit FIFO

Definition at line 117 of file i2s\_5411x.h.

8.25.1.27 `#define I2S_FIFO_CFG_ENBLERX (0x01 << 1)`

Enable the receive FIFO

Definition at line 109 of file i2s\_5411x.h.

**8.25.1.28 #define I2S\_FIFO\_CFG\_ENABLETX (0x01 << 0)**

I2S FIFO configuration bits.

Enable the transmit FIFO

Definition at line 108 of file i2s\_5411x.h.

**8.25.1.29 #define I2S\_FIFO\_CFG\_PACK48 (0x01 << 3)**

Packing format for 48-bit data

Definition at line 111 of file i2s\_5411x.h.

**8.25.1.30 #define I2S\_FIFO\_CFG\_POPDBG (0x01 << 18)**

Pop FIFO for debug reads

Definition at line 119 of file i2s\_5411x.h.

**8.25.1.31 #define I2S\_FIFO\_CFG\_SIZE( p ) (((p) & 0x03) << 4)**

FIFO size (READ ONLY)

Definition at line 112 of file i2s\_5411x.h.

**8.25.1.32 #define I2S\_FIFO\_CFG\_TXI2SE0 (0x01 << 2)**

Transmit I2S empty 0

Definition at line 110 of file i2s\_5411x.h.

**8.25.1.33 #define I2S\_FIFO\_CFG\_WAKERX (0x01 << 15)**

Wake-up for receive FIFO level

Definition at line 116 of file i2s\_5411x.h.

**8.25.1.34 #define I2S\_FIFO\_CFG\_WAKETX (0x01 << 14)**

Wake-up for transmit FIFO level

Definition at line 115 of file i2s\_5411x.h.

**8.25.1.35 #define I2S\_FIFO\_INT\_BITMASK (0x001F) /\*\* FIFO interrupt Bit mask \*/**

I2S FIFO interrupt enable. Used for interrupt status too.

Definition at line 146 of file i2s\_5411x.h.

**8.25.1.36 #define I2S\_FIFO\_INT\_PERINT (0x01 << 4)**

I2S peripheral interrupt [BIT-4 of FIFOINTSTAT register]

Definition at line 151 of file i2s\_5411x.h.

**8.25.1.37 #define I2S\_FIFO\_INT\_RXERR (0x01 << 1)**

Interrupt on RX error

Definition at line 148 of file i2s\_5411x.h.

**8.25.1.38 #define I2S\_FIFO\_INT\_RXLVL (0x01 << 3)**

Interrupt on RX level

Definition at line 150 of file i2s\_5411x.h.

**8.25.1.39 #define I2S\_FIFO\_INT\_TXERR (0x01 << 0)**

Interrupt on TX error

Definition at line 147 of file i2s\_5411x.h.

**8.25.1.40 #define I2S\_FIFO\_INT\_TXLVL (0x01 << 2)**

Interrupt on TX level

Definition at line 149 of file i2s\_5411x.h.

**8.25.1.41 #define I2S\_FIFO\_STAT\_PERINT (0x01 << 3)**

Peripheral interrupt

Definition at line 126 of file i2s\_5411x.h.

**8.25.1.42 #define I2S\_FIFO\_STAT\_RXERR (0x01 << 1)**

RX FIFO error

Definition at line 125 of file i2s\_5411x.h.

**8.25.1.43 #define I2S\_FIFO\_STAT\_RXFULL (0x01 << 7)**

Receive FIFO full

Definition at line 130 of file i2s\_5411x.h.

**8.25.1.44 #define I2S\_FIFO\_STAT\_RXLVL( p ) (((p) & 0x0f) << 16)**

Receive FIFO current level

Definition at line 132 of file i2s\_5411x.h.

**8.25.1.45 #define I2S\_FIFO\_STAT\_RXNOTEMPTY (0x01 << 6)**

Receive FIFO not empty

Definition at line 129 of file i2s\_5411x.h.

**8.25.1.46** `#define I2S_FIFO_STAT_TXEMPTY (0x01 << 4)`

Transmit FIFO empty

Definition at line 127 of file i2s\_5411x.h.

**8.25.1.47** `#define I2S_FIFO_STAT_TXERR (0x01 << 0)`

I2S FIFO status bits.

TX FIFO error

Definition at line 124 of file i2s\_5411x.h.

**8.25.1.48** `#define I2S_FIFO_STAT_TXLVL( p ) (((p) & 0x0f) << 8)`

Transmit FIFO current level

Definition at line 131 of file i2s\_5411x.h.

**8.25.1.49** `#define I2S_FIFO_STAT_TXNOTFULL (0x01 << 5)`

Transmit FIFO not full

Definition at line 128 of file i2s\_5411x.h.

**8.25.1.50** `#define I2S_FIFO_TRIG_RXLVL( p ) (((p) & 0x0f) << 16)`

Receive FIFO level trigger point

Definition at line 140 of file i2s\_5411x.h.

**8.25.1.51** `#define I2S_FIFO_TRIG_RXLVLENA (0x01 << 1)`

Receive FIFO level trigger enable

Definition at line 138 of file i2s\_5411x.h.

**8.25.1.52** `#define I2S_FIFO_TRIG_TXLVL( p ) (((p) & 0x0f) << 8)`

Transmit FIFO level trigger point

Definition at line 139 of file i2s\_5411x.h.

**8.25.1.53** `#define I2S_FIFO_TRIG_TXLVLENA (0x01 << 0)`

I2S FIFO trigger settings bits.

Transmit FIFO level trigger enable

Definition at line 137 of file i2s\_5411x.h.

**8.25.1.54** `#define I2S_STAT_BUSY (0x01 << 0)`

I2S status register bits.

Busy status for the primary channel pair

Definition at line 100 of file i2s\_5411x.h.

**8.25.1.55** `#define I2S_STAT_LR (0x01 << 2)`

Left/Right indication

Definition at line 102 of file i2s\_5411x.h.

**8.25.1.56** `#define I2S_STAT_PAUSED (0x01 << 3)`

Data Paused status flag

Definition at line 103 of file i2s\_5411x.h.

**8.25.1.57** `#define I2S_STAT_SLVFRMERR (0x01 << 1)`

Slave Frame Error flag

Definition at line 101 of file i2s\_5411x.h.

**8.25.1.58** `#define LPC_I2S6 ((LPC_I2S_T *) LPC_I2S6_BASE)`

Definition at line 503 of file i2s\_5411x.h.

**8.25.1.59** `#define LPC_I2S6_BASE __APPEND3(LPC_FLEXCOMM,I2S6_FLEXCOMM,_BASE)`

Definition at line 502 of file i2s\_5411x.h.

**8.25.1.60** `#define LPC_I2S7 ((LPC_I2S_T *) LPC_I2S7_BASE)`

Definition at line 511 of file i2s\_5411x.h.

**8.25.1.61** `#define LPC_I2S7_BASE __APPEND3(LPC_FLEXCOMM,I2S7_FLEXCOMM,_BASE)`

Definition at line 510 of file i2s\_5411x.h.

## **8.25.2 Enumeration Type Documentation**

**8.25.2.1** `enum I2S_DIR_T`

Enumerator

***I2S\_TX***

***I2S\_RX***

Definition at line 163 of file i2s\_5411x.h.

**8.25.2.2** `enum I2S_FIFO_CMD_T`

Enumerator

***I2S\_FIFO\_ENABLE***

***I2S\_FIFO\_DISABLE***

***I2S\_FIFO\_DMA\_ENABLE***

***I2S\_FIFO\_DMA\_DISABLE***

***I2S\_FIFO\_CLEAR***

***I2S\_FIFO\_TXZ\_ENABLE***

***I2S\_FIFO\_TXZ\_DISABLE***

Definition at line 153 of file i2s\_5411x.h.

#### 8.25.2.3 enum I2S\_MODE\_T

Enumerator

***I2S\_CLASSIC***

***DSP\_WS\_50***

***DSP\_WS\_SHORT***

***DSP\_WS\_LONG***

Definition at line 175 of file i2s\_5411x.h.

#### 8.25.2.4 enum I2S\_MSTSLVCFG\_T

Enumerator

***NORMAL\_SLAVE***

***WS\_SYNC\_MASTER***

***EXT\_SCLCK\_MASTER***

***NORMAL\_MASTER***

Definition at line 168 of file i2s\_5411x.h.

### 8.25.3 Function Documentation

#### 8.25.3.1 \_\_STATIC\_INLINE void Chip\_I2S\_ClearStatus ( LPC\_I2S\_T \* *pl2S*, uint32\_t *stsMask* )

Clear the I2S status register.

Parameters

|                |                                       |
|----------------|---------------------------------------|
| <i>pl2S</i>    | : Pointer to selected I2Sx peripheral |
| <i>stsMask</i> | : OR'ed statuses to disable           |

Returns

Nothing

Note

Multiple interrupts may be pending. Mask the return value with one or more I2S\_INTEN\_\* definitions to determine pending interrupts.

Definition at line 272 of file i2s\_5411x.h.

#### 8.25.3.2 \_\_STATIC\_INLINE void Chip\_I2S\_ClrFIFOStatus ( LPC\_I2S\_T \* *pl2S*, uint32\_t *mask* )

Clear FIFO status.



**Parameters**

|             |                                                    |
|-------------|----------------------------------------------------|
| <i>pl2S</i> | : The base of I2S peripheral on the chip           |
| <i>mask</i> | : Mask of the status bits that needs to be cleared |

**Returns**

nothing

Definition at line 388 of file i2s\_5411x.h.

**8.25.3.3 Status Chip\_I2S\_Config ( LPC\_I2S\_T \* *pl2S*, I2S\_AUDIO\_FORMAT\_T \* *fmt* )**

Configure I2S Port.

**Parameters**

|             |                                       |
|-------------|---------------------------------------|
| <i>pl2S</i> | : The base I2S peripheral on the chip |
| <i>fmt</i>  | : Audio Format                        |

**Returns**

SUCCESS or ERROR

Definition at line 67 of file i2s\_5411x.c.

**8.25.3.4 \_\_STATIC\_INLINE void Chip\_I2S\_DeInit ( LPC\_I2S\_T \* *pl2S* )**

Shutdown I2S driver.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <i>pl2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

**Returns**

Nothing

**Note**

Reset all relative registers (DMA, transmit/receive control, interrupt) to default value

Definition at line 237 of file i2s\_5411x.h.

**8.25.3.5 void Chip\_I2S\_ErrorHandler ( LPC\_I2S\_T \* *pl2S*, I2S\_STATISTICS\_T \* *stat* )**

I2S error handler.

**Parameters**

|             |                                              |
|-------------|----------------------------------------------|
| <i>I2S</i>  | : The base of the I2S peripheral on the chip |
| <i>stat</i> | : Statistics structure                       |

**Returns**

Nothing

Definition at line 210 of file i2s\_5411x.c.

8.25.3.6 `__STATIC_INLINE void Chip_I2S_FIFO_ClearStatus ( LPC_I2S_T * pl2S, uint32_t mask )`

Clear the FIFO status register.

## Parameters

|             |                                                    |
|-------------|----------------------------------------------------|
| <i>I2S</i>  | : The base of the I2S peripheral on the chip       |
| <i>mask</i> | : Mask of the status bits that needs to be cleared |

## Returns

Nothing

Definition at line 487 of file i2s\_5411x.h.

### 8.25.3.7 `__STATIC_INLINE void Chip_I2S_FIFO_ClrInterrupt ( LPC_I2S_T * pl2S, uint32_t int_val )`

Clear I2S FIFO interrupts.

## Parameters

|                |                                       |
|----------------|---------------------------------------|
| <i>pl2S</i>    | : The base I2S peripheral on the chip |
| <i>int_val</i> | : Interrupts to clear                 |

## Returns

Nothing

Definition at line 466 of file i2s\_5411x.h.

### 8.25.3.8 `void Chip_I2S_FIFO_Config ( LPC_I2S_T * pl2S, I2S_AUDIO_FORMAT_T * fmt )`

Configure I2S FIFO.

## Parameters

|             |                                       |
|-------------|---------------------------------------|
| <i>pl2S</i> | : The base I2S peripheral on the chip |
| <i>fmt</i>  | : Audio format information            |

## Returns

Nothing

Definition at line 120 of file i2s\_5411x.c.

### 8.25.3.9 `void Chip_I2S_FIFO_Control ( LPC_I2S_T * pl2S, I2S_AUDIO_FORMAT_T * fmt, I2S_FIFO_CMD_T cmd )`

Execute I2S FIFO control commands.

## Parameters

|             |                                       |
|-------------|---------------------------------------|
| <i>pl2S</i> | : The base I2S peripheral on the chip |
| <i>fmt</i>  | : Audio format information            |
| <i>cmd</i>  | : FIFO command                        |

## Returns

Nothing

Definition at line 191 of file i2s\_5411x.c.

### 8.25.3.10 `__STATIC_INLINE uint32_t Chip_I2S_FIFO_GetPendingInts ( LPC_I2S_T * pl2S )`

Get I2S FIFO interrupts status.

**Parameters**

|             |                                       |
|-------------|---------------------------------------|
| <i>pI2S</i> | : The base I2S peripheral on the chip |
|-------------|---------------------------------------|

**Returns**

interrupt status

Definition at line 476 of file i2s\_5411x.h.

8.25.3.11 `__STATIC_INLINE void Chip_I2S_FIFO_SetInterrupt ( LPC_I2S_T * pI2S, uint32_t int_val )`

Set I2S FIFO interrupts.

**Parameters**

|                |                                       |
|----------------|---------------------------------------|
| <i>pI2S</i>    | : The base I2S peripheral on the chip |
| <i>int_val</i> | : Interrupts to set                   |

**Returns**

Nothing

Definition at line 455 of file i2s\_5411x.h.

8.25.3.12 `__STATIC_INLINE uint8_t Chip_I2S_GetFIFORxLevel ( LPC_I2S_T * pI2S )`

Get the current level of the Receive FIFO.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <i>pI2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

**Returns**

Current level of the Receive FIFO

Definition at line 441 of file i2s\_5411x.h.

8.25.3.13 `__STATIC_INLINE uint32_t Chip_I2S_GetFIFOStatus ( LPC_I2S_T * pI2S )`

Get FIFO status.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <i>pI2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

**Returns**

Current FIFO status

Definition at line 377 of file i2s\_5411x.h.

8.25.3.14 `__STATIC_INLINE uint32_t Chip_I2S_GetFIFOTrigLevel ( LPC_I2S_T * pI2S )`

Get I2S FIFO trigger-level.

## Parameters

|             |                                  |
|-------------|----------------------------------|
| <i>pI2S</i> | : Base of on-chip I2S peripheral |
|-------------|----------------------------------|

## Returns

Returns the complete raw trigger register.

Definition at line 418 of file i2s\_5411x.h.

8.25.3.15 `__STATIC_INLINE uint8_t Chip_I2S_GetFIFOTxLevel ( LPC_I2S_T * pI2S )`

Get the current level of the Transmit FIFO.

## Parameters

|             |                                          |
|-------------|------------------------------------------|
| <i>pI2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

## Returns

Current level of the Transmit FIFO

Definition at line 431 of file i2s\_5411x.h.

8.25.3.16 `__STATIC_INLINE uint32_t Chip_I2S_GetStatus ( LPC_I2S_T * pI2S )`

Get the I2S status register.

## Parameters

|             |                                       |
|-------------|---------------------------------------|
| <i>pI2S</i> | : Pointer to selected I2Sx peripheral |
|-------------|---------------------------------------|

## Returns

I2S status register

## Note

Multiple statuses may be pending. Mask the return value with one or more I2S\_STAT\_\* definitions to determine statuses.

Definition at line 258 of file i2s\_5411x.h.

8.25.3.17 `int Chip_I2S_Init ( LPC_I2S_T * pI2S, I2S_AUDIO_FORMAT_T * fmt )`

Initialize I2S driver.

## Parameters

|             |                                          |
|-------------|------------------------------------------|
| <i>pI2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

## Returns

0 - on success; [ERR\\_FLEXCOMM\\_FUNCNOTSUPPORTED](#) or [ERR\\_FLEXCOMM\\_NOTFREE](#) on failure

Definition at line 49 of file i2s\_5411x.c.

8.25.3.18 `__STATIC_INLINE void Chip_I2S_Pause ( LPC_I2S_T * pI2S )`

Pause the I2S port.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <i>pI2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

**Returns**

Nothing

Definition at line 326 of file i2s\_5411x.h.

**8.25.3.19 \_\_STATIC\_INLINE void Chip\_I2S\_Play ( LPC\_I2S\_T \* *pI2S* )**

Play (un-Pause) the I2S port.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <i>pI2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

**Returns**

Nothing

Definition at line 336 of file i2s\_5411x.h.

**8.25.3.20 \_\_STATIC\_INLINE uint32\_t Chip\_I2S\_Receive ( LPC\_I2S\_T \* *pI2S* )**

Get received data from RXFIFO.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <i>pI2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

**Returns**

Data received in RXFIFO

**Note**

The function reads from RXFIFO without checking any condition.

Definition at line 295 of file i2s\_5411x.h.

**8.25.3.21 \_\_STATIC\_INLINE int Chip\_I2S\_RX\_Init ( LPC\_I2S\_T \* *pI2S* )**

Definition at line 226 of file i2s\_5411x.h.

**8.25.3.22 \_\_STATIC\_INLINE void Chip\_I2S\_Send ( LPC\_I2S\_T \* *pI2S*, uint32\_t *data* )**

Send a 32-bit data to TXFIFO for transmission.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <i>pI2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

|             |                          |
|-------------|--------------------------|
| <i>data</i> | : Data to be transmitted |
|-------------|--------------------------|

**Returns**

Nothing

**Note**

The function writes to TXFIFO without checking any condition.

Definition at line 284 of file i2s\_5411x.h.

#### 8.25.3.23 `__STATIC_INLINE void Chip_I2S_SetFIFOTrigLevel ( LPC_I2S_T * pI2S, uint8_t tx_lvl, uint8_t rx_lvl )`

Setup I2S FIFO trigger-level.

**Parameters**

|               |                                          |
|---------------|------------------------------------------|
| <i>pI2S</i>   | : Base on-chip I2S peripheral address    |
| <i>tx_lvl</i> | : TX Trigger level [Valid values 0 to 7] |
| <i>rx_lvl</i> | : RX Trigger level [Valid values 0 to 7] |

**Returns**

Nothing

**Note**

When *tx\_lvl* = 0; trigger will happen when TX FIFO is empty if *tx\_lvl* = 7; trigger will happen when TX FIFO has at least one free space

When *rx\_lvl* = 0; trigger will happen when RX FIFO has at least one data in it, if *rx\_lvl* = 7; trigger will happen when RX FIFO is full and cannot receive anymore data.

Definition at line 408 of file i2s\_5411x.h.

#### 8.25.3.24 `__STATIC_INLINE void Chip_I2S_Start ( LPC_I2S_T * pI2S )`

Start I2S port.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <i>pI2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

**Returns**

Nothing

Definition at line 305 of file i2s\_5411x.h.

#### 8.25.3.25 `__STATIC_INLINE void Chip_I2S_Stop ( LPC_I2S_T * pI2S )`

Stop I2S asynchronously.

## Parameters

|             |                                          |
|-------------|------------------------------------------|
| <i>pl2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

## Returns

Nothing

## Note

Pause, resets the transmit channel and FIFO asynchronously

Definition at line 316 of file i2s\_5411x.h.

8.25.3.26 `__STATIC_INLINE int Chip_I2S_TX_Init ( LPC_I2S_T * pl2S )`

Definition at line 222 of file i2s\_5411x.h.

## 8.26 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/iap.h File Reference

## Macros

- `#define IAP_PREWRITE_CMD` 50
- `#define IAP_WRITECTOR_CMD` 51
- `#define IAP_ERSECTOR_CMD` 52
- `#define IAP_BLANK_CHECK_SECTOR_CMD` 53
- `#define IAP_READID_CMD` 54
- `#define IAP_READ_BOOT_CODE_CMD` 55
- `#define IAP_COMPARE_CMD` 56
- `#define IAP_REINVOKE_ISP_CMD` 57
- `#define IAP_READ_UID_CMD` 58
- `#define IAP_ERASE_PAGE_CMD` 59
- `#define IAP_EEPROM_WRITE` 61
- `#define IAP_EEPROM_READ` 62
- `#define IAP_CMD_SUCCESS` 0
- `#define IAP_INVALID_COMMAND` 1
- `#define IAP_SRC_ADDR_ERROR` 2
- `#define IAP_DST_ADDR_ERROR` 3
- `#define IAP_SRC_ADDR_NOT_MAPPED` 4
- `#define IAP_DST_ADDR_NOT_MAPPED` 5
- `#define IAP_COUNT_ERROR` 6
- `#define IAP_INVALID_SECTOR` 7
- `#define IAP_SECTOR_NOT_BLANK` 8
- `#define IAP_SECTOR_NOT_PREPARED` 9
- `#define IAP_COMPARE_ERROR` 10
- `#define IAP_BUSY` 11
- `#define IAP_PARAM_ERROR` 12
- `#define IAP_ADDR_ERROR` 13
- `#define IAP_ADDR_NOT_MAPPED` 14
- `#define IAP_CMD_LOCKED` 15
- `#define IAP_INVALID_CODE` 16
- `#define IAP_INVALID_BAUD_RATE` 17
- `#define IAP_INVALID_STOP_BIT` 18
- `#define IAP_CRP_ENABLED` 19



## Typedefs

- typedef void(\* [IAP\\_ENTRY\\_T](#)) (unsigned int[5], unsigned int[4])

## Functions

- uint8\_t [Chip\\_IAP\\_PreSectorForReadWrite](#) (uint32\_t strSector, uint32\_t endSector)  
*Prepare sector for write operation.*
- uint8\_t [Chip\\_IAP\\_CopyRamToFlash](#) (uint32\_t dstAdd, uint32\_t \*srcAdd, uint32\_t byteswrt)  
*Copy RAM to flash.*
- uint8\_t [Chip\\_IAP\\_EraseSector](#) (uint32\_t strSector, uint32\_t endSector)  
*Erase sector.*
- uint8\_t [Chip\\_IAP\\_BlankCheckSector](#) (uint32\_t strSector, uint32\_t endSector)  
*Blank check a sector or multiples sector of on-chip flash memory.*
- uint32\_t [Chip\\_IAP\\_ReadPID](#) (void)  
*Read part identification number.*
- uint8\_t [Chip\\_IAP\\_ReadBootCode](#) (void)  
*Read boot code version number.*
- uint8\_t [Chip\\_IAP\\_Compare](#) (uint32\_t dstAdd, uint32\_t srcAdd, uint32\_t bytescmp)  
*Compare the memory contents at two locations.*
- uint8\_t [Chip\\_IAP\\_ReinvokeISP](#) (void)  
*IAP reinvoke ISP to invoke the bootloader in ISP mode.*
- uint32\_t [Chip\\_IAP\\_ReadUID](#) (void)  
*Read the unique ID.*
- uint8\_t [Chip\\_IAP\\_ErasePage](#) (uint32\_t strPage, uint32\_t endPage)  
*Erase a page or multiple papers of on-chip flash memory.*

## 8.27 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/inmux\_5411x.h File Reference

### Data Structures

- struct [LPC\\_INMUX\\_T](#)  
*LPC5411X Input Mux Register Block Structure.*

### Enumerations

- enum [DMA\\_TRIGSRC\\_T](#) {  
DMATRIG\_ADC0\_SEQA\_IRQ = 0, DMATRIG\_ADC0\_SEQB\_IRQ, DMATRIG\_SCT0\_DMA0, DMATRIG\_↵  
SCT0\_DMA1,  
DMATRIG\_TIMER0\_MATCH0, DMATRIG\_TIMER0\_MATCH1, DMATRIG\_TIMER1\_MATCH0, DMATRIG\_↵  
TIMER2\_MATCH0,  
DMATRIG\_TIMER2\_MATCH1, DMATRIG\_TIMER3\_MATCH0, DMATRIG\_TIMER4\_MATCH0, DMATRIG\_↵  
TIMER4\_MATCH1,  
DMATRIG\_PININT0, DMATRIG\_PININT1, DMATRIG\_PININT2, DMATRIG\_PININT3,  
DMATRIG\_OUTMUX0, DMATRIG\_OUTMUX1, DMATRIG\_OUTMUX2, DMATRIG\_OUTMUX3 }  
• enum [FREQMSR\\_SRC\\_T](#) {  
FREQMSR\_CLKIN = 0, FREQMSR\_FRO12MHZ, FREQMSR\_WDOSC, FREQMSR\_32KHZOSC,  
FREQ\_MEAS\_MAIN\_CLK, FREQMSR\_PIO0\_4, FREQMSR\_PIO0\_20, FREQMSR\_PIO0\_24,  
FREQMSR\_PIO1\_4 }

## Functions

- `__STATIC_INLINE void Chip\_INMUX\_PinIntSel (uint8_t pintSel, uint8_t portNum, uint8_t pinNum)`  
*GPIO Pin Interrupt Pin Select (sets PINTSEL register)*
- `__STATIC_INLINE void Chip\_INMUX\_SetDMATrigger (uint8_t ch, DMA\_TRIGSRC\_T trig)`  
*Select a trigger source for a DMA channel.*
- `__STATIC_INLINE void Chip\_INMUX\_SetDMAOutMux (uint8_t index, uint8_t dmaCh)`  
*Selects a DMA trigger source for the DMATRIG\_OUTMUXn IDs.*
- `__STATIC_INLINE void Chip\_INMUX\_SetFreqMeasRefClock (FREQMSR\_SRC\_T ref)`  
*Selects a reference clock used with the frequency measure function.*
- `__STATIC_INLINE void Chip\_INMUX\_SetFreqMeasTargClock (FREQMSR\_SRC\_T targ)`  
*Selects a target clock used with the frequency measure function.*

## 8.28 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/iocon\_5411x.h File Reference

### Data Structures

- struct [LPC\\_IOCON\\_T](#)  
*LPC5411X IO Configuration Unit register block structure.*
- struct [PINMUX\\_GRP\\_T](#)  
*Array of IOCON pin definitions passed to [Chip\\_IOCON\\_SetPinMuxing\(\)](#) must be in this format.*

### Macros

- `#define IOCON\_FUNC0 0x0`
- `#define IOCON\_FUNC1 0x1`
- `#define IOCON\_FUNC2 0x2`
- `#define IOCON\_FUNC3 0x3`
- `#define IOCON\_FUNC4 0x4`
- `#define IOCON\_FUNC5 0x5`
- `#define IOCON\_FUNC6 0x6`
- `#define IOCON\_FUNC7 0x7`
- `#define IOCON\_MODE\_INACT (0x0 << 3)`
- `#define IOCON\_MODE\_PULLDOWN (0x1 << 3)`
- `#define IOCON\_MODE\_PULLUP (0x2 << 3)`
- `#define IOCON\_MODE\_REPEATER (0x3 << 3)`
- `#define IOCON\_HYS\_EN (0x1 << 5)`
- `#define IOCON\_GPIO\_MODE (0x1 << 5)`
- `#define IOCON\_I2C\_SLEW (0x1 << 5)`
- `#define IOCON\_INV\_EN (0x1 << 6)`
- `#define IOCON\_ANALOG\_EN (0x0 << 7)`
- `#define IOCON\_DIGITAL\_EN (0x1 << 7)`
- `#define IOCON\_STDI2C\_EN (0x1 << 8)`
- `#define IOCON\_FASTI2C\_EN (0x3 << 8)`
- `#define IOCON\_INPFILT\_OFF (0x1 << 8)`
- `#define IOCON\_INPFILT\_ON (0x0 << 8)`
- `#define IOCON\_OPENDRAIN\_EN (0x1 << 10)`
- `#define IOCON\_S\_MODE\_0CLK (0x0 << 11)`
- `#define IOCON\_S\_MODE\_1CLK (0x1 << 11)`
- `#define IOCON\_S\_MODE\_2CLK (0x2 << 11)`
- `#define IOCON\_S\_MODE\_3CLK (0x3 << 11)`
- `#define IOCON\_S\_MODE(clks) ((clks) << 11)`
- `#define IOCON\_CLKDIV(div) ((div) << 13)`

## Functions

- `__STATIC_INLINE void Chip_IOCON_PinMuxSet (LPC_IOCON_T *pIOCON, uint8_t port, uint8_t pin, uint32_t modefunc)`  
*Sets I/O Control pin mux.*
- `__STATIC_INLINE void Chip_IOCON_PinMux (LPC_IOCON_T *pIOCON, uint8_t port, uint8_t pin, uint16_t mode, uint8_t func)`  
*I/O Control pin mux.*
- `__STATIC_INLINE void Chip_IOCON_SetPinMuxing (LPC_IOCON_T *pIOCON, const PINMUX_GRP_T *pinArray, uint32_t arrayLength)`  
*Set all I/O Control pin muxing.*

## 8.29 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/lpc\_assert.h File Reference

### Macros

- `#define LPC_ASSERT(cond, file, line) if(!(cond)){ /* Required to avoid unused variable warning */`

#### 8.29.1 Macro Definition Documentation

8.29.1.1 `#define LPC_ASSERT( cond, file, line ) if(!(cond)){ /* Required to avoid unused variable warning */`

Definition at line 54 of file `lpc_assert.h`.

## 8.30 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/lpc\_types.h File Reference

```
#include <stdint.h>
#include <stdbool.h>
```

### Macros

- `#define PARAM_SETSTATE(State) ((State == RESET) || (State == SET))`
- `#define PARAM_FUNCTIONALSTATE(State) ((State == DISABLE) || (State == ENABLE))`
- `#define _BIT(n) (1 << (n))`
- `#define _SBF(f, v) ((v) << (f))`
- `#define _BITMASK(field_width) ( _BIT(field_width) - 1)`
- `#define NULL ((void *) 0)`
- `#define NELEMENTS(array) (sizeof(array) / sizeof(array[0]))`
- `#define STATIC static`
- `#define EXTERN extern`
- `#define MAX(a, b) (((a) > (b)) ? (a) : (b))`
- `#define MIN(a, b) (((a) < (b)) ? (a) : (b))`
- `#define INLINE inline`
- `#define ALIGN(x) __attribute__((aligned(x)))`
- `#define WEAK __attribute__((weak))`

## Typedefs

- typedef enum [FlagStatus](#) [IntStatus](#)
- typedef enum [FlagStatus](#) [SetState](#)
- typedef void(\* [PFV](#)) ()
- typedef int32\_t(\* [PFI](#)) ()
- typedef char [CHAR](#)
- typedef uint8\_t [UNS\\_8](#)
- typedef int8\_t [INT\\_8](#)
- typedef uint16\_t [UNS\\_16](#)
- typedef int16\_t [INT\\_16](#)
- typedef uint32\_t [UNS\\_32](#)
- typedef int32\_t [INT\\_32](#)
- typedef int64\_t [INT\\_64](#)
- typedef uint64\_t [UNS\\_64](#)
- typedef bool [BOOL\\_32](#)
- typedef bool [BOOL\\_16](#)
- typedef bool [BOOL\\_8](#)

## Enumerations

- enum [Bool](#) { [FALSE](#) = 0, [TRUE](#) = ![FALSE](#) }  
*Boolean Type definition.*
- enum [FlagStatus](#) { [RESET](#) = 0, [SET](#) = ![RESET](#) }  
*Boolean Type definition.*
- enum [FunctionalState](#) { [DISABLE](#) = 0, [ENABLE](#) = ![DISABLE](#) }  
*Functional State Definition.*
- enum [Status](#) { [ERROR](#) = 0, [SUCCESS](#) = ![ERROR](#) }
- enum [TRANSFER\\_BLOCK\\_T](#) { [NONE\\_BLOCKING](#) = 0, [BLOCKING](#) }

## 8.31 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/mailbox\_5411x.h File Reference

### Data Structures

- struct [LPC\\_MBOXIRQ\\_T](#)
- struct [LPC\\_MBOX\\_T](#)

### Macros

- #define [MAILBOX\\_AVAIL](#) ([MAILBOX\\_CM4](#) + 1) /\* Number of available mailboxes \*/

### Enumerations

- enum [MBOX\\_IDX\\_T](#) { [MAILBOX\\_CM0PLUS](#) = 0, [MAILBOX\\_CM4](#) }

## Functions

- `__STATIC_INLINE void Chip_MBOX_Init (LPC_MBOX_T *pMBOX)`  
*Initialize mailbox.*
- `__STATIC_INLINE void Chip_MBOX_DeInit (LPC_MBOX_T *pMBOX)`  
*Shutdown mailbox.*
- `__STATIC_INLINE void Chip_MBOX_SetValue (LPC_MBOX_T *pMBOX, uint32_t cpu_id, uint32_t mbox←Data)`  
*Set data value in the mailbox based on the CPU ID.*
- `__STATIC_INLINE void Chip_MBOX_SetValueBits (LPC_MBOX_T *pMBOX, uint32_t cpu_id, uint32_←t mboxSetBits)`  
*Set data bits in the mailbox based on the CPU ID.*
- `__STATIC_INLINE void Chip_MBOX_ClearValueBits (LPC_MBOX_T *pMBOX, uint32_t cpu_id, uint32_←t mboxClrBits)`  
*Clear data bits in the mailbox based on the CPU ID.*
- `__STATIC_INLINE uint32_t Chip_MBOX_GetValue (LPC_MBOX_T *pMBOX, uint32_t cpu_id)`  
*Get data in the mailbox based on the cpu\_id.*
- `__STATIC_INLINE uint32_t Chip_MBOX_GetMutex (LPC_MBOX_T *pMBOX)`  
*Get MUTEX state and lock mutex.*
- `__STATIC_INLINE void Chip_MBOX_SetMutex (LPC_MBOX_T *pMBOX)`  
*Set MUTEX state.*

## 8.32 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/mrt\_5411x.h File Reference

### Data Structures

- struct `LPC_MRT_CH_T`  
*MRT register block structure.*
- struct `LPC_MRT_T`  
*MRT register block structure.*

### Macros

- `#define MRT_CHANNELS_NUM (4)`  
*LPC5411X MRT chip configuration.*
- `#define MRT_NO_IDLE_CHANNEL (0x40)`
- `#define MRT_INTVAL_IVALUE (0xFFFFF) /* Maximum interval load value and mask */`  
*MRT register bit fields & masks.*
- `#define MRT_INTVAL_LOAD (0x80000000UL) /* Force immediate load of timer interval register bit */`
- `#define MRT_CTRL_INTEN_MASK (0x01)`
- `#define MRT_CTRL_MODE_MASK (0x06)`
- `#define MRT_STAT_INTFLAG (0x01)`
- `#define MRT_STAT_RUNNING (0x02)`
- `#define LPC_MRT_CH0 ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[0])`
- `#define LPC_MRT_CH1 ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[1])`
- `#define LPC_MRT_CH2 ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[2])`
- `#define LPC_MRT_CH3 ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[3])`
- `#define LPC_MRT_CH(ch) ((LPC_MRT_CH_T *) &LPC_MRT->CHANNEL[(ch)])`
- `#define MRT0_INTFLAG (1)`
- `#define MRT1_INTFLAG (2)`
- `#define MRT2_INTFLAG (4)`
- `#define MRT3_INTFLAG (8)`
- `#define MRTn_INTFLAG(ch) (1 << (ch))`

## Enumerations

- enum `MRT_MODE_T` { `MRT_MODE_REPEAT` = (0 << 1), `MRT_MODE_ONESHOT` = (1 << 1) }  
*MRT Interrupt Modes enum.*

## Functions

- \_\_STATIC\_INLINE void `Chip_MRT_Init` (void)  
*Initializes the MRT.*
- \_\_STATIC\_INLINE void `Chip_MRT_DeInit` (void)  
*De-initializes the MRT Channel.*
- \_\_STATIC\_INLINE `LPC_MRT_CH_T` \* `Chip_MRT_GetRegPtr` (uint8\_t ch)  
*Returns a pointer to the register block for a MRT channel.*
- \_\_STATIC\_INLINE uint32\_t `Chip_MRT_GetInterval` (`LPC_MRT_CH_T` \*pMRT)  
*Returns the timer time interval value.*
- \_\_STATIC\_INLINE void `Chip_MRT_SetInterval` (`LPC_MRT_CH_T` \*pMRT, uint32\_t interval)  
*Sets the timer time interval value.*
- \_\_STATIC\_INLINE uint32\_t `Chip_MRT_GetTimer` (`LPC_MRT_CH_T` \*pMRT)  
*Returns the current timer value.*
- \_\_STATIC\_INLINE bool `Chip_MRT_GetEnabled` (`LPC_MRT_CH_T` \*pMRT)  
*Returns true if the timer is enabled.*
- \_\_STATIC\_INLINE void `Chip_MRT_SetEnabled` (`LPC_MRT_CH_T` \*pMRT)  
*Enables the timer.*
- \_\_STATIC\_INLINE void `Chip_MRT_SetDisabled` (`LPC_MRT_CH_T` \*pMRT)  
*Disables the timer.*
- \_\_STATIC\_INLINE `MRT_MODE_T` `Chip_MRT_GetMode` (`LPC_MRT_CH_T` \*pMRT)  
*Returns the timer mode (repeat or one-shot)*
- \_\_STATIC\_INLINE void `Chip_MRT_SetMode` (`LPC_MRT_CH_T` \*pMRT, `MRT_MODE_T` mode)  
*Sets the timer mode (repeat or one-shot)*
- \_\_STATIC\_INLINE bool `Chip_MRT_IsRepeatMode` (`LPC_MRT_CH_T` \*pMRT)  
*Check if the timer is configured in repeat mode.*
- \_\_STATIC\_INLINE bool `Chip_MRT_IsOneShotMode` (`LPC_MRT_CH_T` \*pMRT)  
*Check if the timer is configured in one-shot mode.*
- \_\_STATIC\_INLINE bool `Chip_MRT_IntPending` (`LPC_MRT_CH_T` \*pMRT)  
*Check if the timer has an interrupt pending.*
- \_\_STATIC\_INLINE void `Chip_MRT_IntClear` (`LPC_MRT_CH_T` \*pMRT)  
*Clears the pending interrupt (if any)*
- \_\_STATIC\_INLINE bool `Chip_MRT_Running` (`LPC_MRT_CH_T` \*pMRT)  
*Check if the timer is running.*
- \_\_STATIC\_INLINE uint8\_t `Chip_MRT_GetIdleChannel` (void)  
*Returns the IDLE channel value.*
- \_\_STATIC\_INLINE uint8\_t `Chip_MRT_GetIdleChannelShifted` (void)  
*Returns the IDLE channel value.*
- \_\_STATIC\_INLINE uint32\_t `Chip_MRT_GetIntPending` (void)  
*Returns the interrupt pending status for all MRT channels.*
- \_\_STATIC\_INLINE bool `Chip_MRT_GetIntPendingByChannel` (uint8\_t ch)  
*Returns the interrupt pending status for a singel MRT channel.*
- \_\_STATIC\_INLINE void `Chip_MRT_ClearIntPending` (uint32\_t mask)  
*Clears the interrupt pending status for one or more MRT channels.*

## 8.33 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/packing.h File Reference

### Macros

- #define [ALIGNED](#)(n) /\* Nothing \*/

#### 8.33.1 Macro Definition Documentation

##### 8.33.1.1 #define [ALIGNED](#)( n ) /\* Nothing \*/

Definition at line 37 of file packing.h.

## 8.34 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/pinint\_5411x.h File Reference

### Data Structures

- struct [LPC\\_PIN\\_INT\\_T](#)  
*LPC5411X Pin Interrupt and Pattern Match register block structure.*

### Macros

- #define [PININT\\_ISEL\\_PMODE\\_MASK](#) ((uint32\_t) 0x00FF)
- #define [PININT\\_PMCTRL\\_MASK](#) ((uint32\_t) 0xFF000003)
- #define [PININT\\_PMCTRL\\_PMATCH\\_SEL](#) (1 << 0)
- #define [PININT\\_PMCTRL\\_RXEV\\_ENA](#) (1 << 1)
- #define [PININT\\_SRC\\_BITSOURCE\\_START](#) 8
- #define [PININT\\_SRC\\_BITSOURCE\\_MASK](#) 7
- #define [PININT\\_SRC\\_BITCFG\\_START](#) 8
- #define [PININT\\_SRC\\_BITCFG\\_MASK](#) 7
- #define [PININTCH0](#) (1 << 0)
- #define [PININTCH1](#) (1 << 1)
- #define [PININTCH2](#) (1 << 2)
- #define [PININTCH3](#) (1 << 3)
- #define [PININTCH4](#) (1 << 4)
- #define [PININTCH5](#) (1 << 5)
- #define [PININTCH6](#) (1 << 6)
- #define [PININTCH7](#) (1 << 7)
- #define [PININTCH](#)(ch) (1 << (ch))

### Enumerations

- enum [Chip\\_PININT\\_SELECT\\_T](#) {  
[PININTSELECT0](#) = 0, [PININTSELECT1](#) = 1, [PININTSELECT2](#) = 2, [PININTSELECT3](#) = 3,  
[PININTSELECT4](#) = 4, [PININTSELECT5](#) = 5, [PININTSELECT6](#) = 6, [PININTSELECT7](#) = 7 }
- enum [Chip\\_PININT\\_BITSLICE\\_T](#) {  
[PININTBITSLICE0](#) = 0, [PININTBITSLICE1](#) = 1, [PININTBITSLICE2](#) = 2, [PININTBITSLICE3](#) = 3,  
[PININTBITSLICE4](#) = 4, [PININTBITSLICE5](#) = 5, [PININTBITSLICE6](#) = 6, [PININTBITSLICE7](#) = 7 }
- enum [Chip\\_PININT\\_BITSLICE\\_CFG\\_T](#) {  
[PININT\\_PATTERNCONST1](#) = 0x0, [PININT\\_PATTERNRISING](#) = 0x1, [PININT\\_PATTERNFALLING](#) = 0x2,  
[PININT\\_PATTERNRISINGORFALLING](#) = 0x3,  
[PININT\\_PATTERNHIGH](#) = 0x4, [PININT\\_PATTERNLOW](#) = 0x5, [PININT\\_PATTERNCONST0](#) = 0x6, [PININT\\_PATTERNEVENT](#) = 0x7 }

## Functions

- `__STATIC_INLINE void Chip_PININT_Init (LPC_PIN_INT_T *pPININT)`  
*Initialize Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_DeInit (LPC_PIN_INT_T *pPININT)`  
*De-Initialize Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_SetPinModeEdge (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Configure the pins as edge sensitive in Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_SetPinModeLevel (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Configure the pins as level sensitive in Pin interrupt block.*
- `__STATIC_INLINE uint32_t Chip_PININT_GetPinMode (LPC_PIN_INT_T *pPININT)`  
*Return current PININT edge or level sensitive interrupt selection state.*
- `__STATIC_INLINE uint32_t Chip_PININT_GetHighEnabled (LPC_PIN_INT_T *pPININT)`  
*Return current PININT rising edge or level interrupt enable state.*
- `__STATIC_INLINE void Chip_PININT_EnableIntHigh (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Enable rising edge/level PININT interrupts for pins.*
- `__STATIC_INLINE void Chip_PININT_DisableIntHigh (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Disable rising edge/level PININT interrupts for pins.*
- `__STATIC_INLINE uint32_t Chip_PININT_GetLowEnabled (LPC_PIN_INT_T *pPININT)`  
*Return current PININT falling edge or level interrupt active level enable state.*
- `__STATIC_INLINE void Chip_PININT_EnableIntLow (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Enable falling edge/level active level PININT interrupts for pins.*
- `__STATIC_INLINE void Chip_PININT_DisableIntLow (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Disable low edge/level active level PININT interrupts for pins.*
- `__STATIC_INLINE uint32_t Chip_PININT_GetRiseStates (LPC_PIN_INT_T *pPININT)`  
*Return pin states that have a detected latched rising edge (RISE) state.*
- `__STATIC_INLINE void Chip_PININT_ClearRiseStates (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Clears pin states that had a latched rising edge (RISE) state.*
- `__STATIC_INLINE uint32_t Chip_PININT_GetFallStates (LPC_PIN_INT_T *pPININT)`  
*Return pin states that have a detected latched falling edge (FALL) state.*
- `__STATIC_INLINE void Chip_PININT_ClearFallStates (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Clears pin states that had a latched falling edge (FALL) state.*
- `__STATIC_INLINE uint32_t Chip_PININT_GetIntStatus (LPC_PIN_INT_T *pPININT)`  
*Get interrupt status from Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_ClearIntStatus (LPC_PIN_INT_T *pPININT, uint32_t pins)`  
*Clear interrupt status in Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_SetPatternMatchSrc (LPC_PIN_INT_T *pPININT, Chip_PININT_SELECT_T channelNum, Chip_PININT_BITSLICE_T sliceNum)`  
*Set source for pattern match in Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_SetPatternMatchConfig (LPC_PIN_INT_T *pPININT, Chip_PININT_BITSLICE_T sliceNum, Chip_PININT_BITSLICE_CFG_T slice_cfg, bool end_point)`  
*Configure the pattern match in Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_EnablePatternMatch (LPC_PIN_INT_T *pPININT)`  
*Enable pattern match interrupts in Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_DisablePatternMatch (LPC_PIN_INT_T *pPININT)`  
*Disable pattern match interrupts in Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_EnablePatternMatchRxEv (LPC_PIN_INT_T *pPININT)`  
*Enable RXEV output in Pin interrupt block.*
- `__STATIC_INLINE void Chip_PININT_DisablePatternMatchRxEv (LPC_PIN_INT_T *pPININT)`  
*Disable RXEV output in Pin interrupt block.*
- `__STATIC_INLINE uint32_t Chip_PININT_GetPatternMatchState (LPC_PIN_INT_T *pPININT)`  
*Return pattern match state.*



## 8.35 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/pintable\_5411x.h File Reference

### Data Structures

- struct [PINTABLE\\_T](#)

*LPC5411X Pin table structure used for enhanced boot block support.*

### Macros

- #define [IMAGE\\_ENH\\_MARKER\\_OFF](#) 0x24
- #define [IMAGE\\_SINGLE\\_ENH\\_SIG](#) 0xEDDC9494
- #define [IMAGE\\_DUAL\\_ENH\\_SIG](#) 0x0FFEB6B6
- #define [IMAGE\\_BOOT\\_BLOCK\\_OFF](#) 0x28
- #define [IMAGE\\_ENH\\_BLOCK\\_MARKER](#) 0xFEEDA5A5
- #define [SETPORTPIN](#)(port, pin) (((port) & 0x7) << 5) | ((pin) & 0x1F)

### Enumerations

- enum [IMAGE\\_T](#) {  
[IMG\\_NORMAL](#) = 0, [IMG\\_ISP\\_WAIT](#), [IMG\\_NO\\_WAIT](#), [IMG\\_NO\\_CRC](#),  
[IMG\\_JUST\\_BOOT](#) = 0xFF }
- enum [IFSEL\\_T](#) {  
[SL\\_AUTO](#) = 0, [SL\\_I2C0](#), [SL\\_I2C1](#), [SL\\_I2C2](#),  
[SL\\_SPI0](#), [SL\\_SPI1](#) }

## 8.36 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/pll\_5411x.h File Reference

### Data Structures

- struct [PLL\\_CONFIG\\_T](#)

*PLL configuration structure This structure can be used to configure the settings for a PLL setup structure. Fill in the desired configuration for the PLL and call the PLL setup function to fill in a PLL setup structure.*

- struct [PLL\\_SETUP\\_T](#)

*PLL setup structure This structure can be used to pre-build a PLL setup configuration at run-time and quickly set the PLL to the configuration. It can be populated with the PLL setup function. If powering up or waiting for PLL lock, the PLL input clock source should be configured prior to PLL setup.*

### Macros

- #define [PLL\\_CONFIGFLAG\\_USEINRATE](#) (1 << 0)  
*PLL configuration structure flags for 'flags' field These flags control how the PLL configuration function sets up the PLL setup structure.*
- #define [PLL\\_CONFIGFLAG\\_FORCENOFRACT](#) (1 << 2)
- #define [PLL\\_SETUPFLAG\\_POWERUP](#) (1 << 0)  
*PLL setup structure flags for 'flags' field These flags control how the PLL setup function sets up the PLL.*
- #define [PLL\\_SETUPFLAG\\_WAITLOCK](#) (1 << 1)
- #define [PLL\\_SETUPFLAG\\_ADGVOLT](#) (1 << 2)

## Enumerations

- enum [CHIP\\_SYSCON\\_PLLCLKSRC\\_T](#) {  
[SYSCON\\_PLLCLKSRC\\_FRO12MHZ](#) = 0, [SYSCON\\_PLLCLKSRC\\_CLKIN](#), [SYSCON\\_PLLCLKSRC\\_WD](#),  
[T](#), [SYSCON\\_PLLCLKSRC\\_RTC](#),  
[SYSCON\\_PLLCLKSRC\\_DISABLED](#) = 7 }
- enum [SS\\_PROGMODFM\\_T](#) {  
[SS\\_MF\\_512](#) = (0 << 20), [SS\\_MF\\_384](#) = (1 << 20), [SS\\_MF\\_256](#) = (2 << 20), [SS\\_MF\\_128](#) = (3 << 20),  
[SS\\_MF\\_64](#) = (4 << 20), [SS\\_MF\\_32](#) = (5 << 20), [SS\\_MF\\_24](#) = (6 << 20), [SS\\_MF\\_16](#) = (7 << 20) }  
*PLL Spread Spectrum (SS) Programmable modulation frequency See (MF) field in the SYSPLLSSCTRL1 register in the UM.*
- enum [SS\\_PROGMODDP\\_T](#) {  
[SS\\_MR\\_K0](#) = (0 << 23), [SS\\_MR\\_K1](#) = (1 << 23), [SS\\_MR\\_K1\\_5](#) = (2 << 23), [SS\\_MR\\_K2](#) = (3 << 23),  
[SS\\_MR\\_K3](#) = (4 << 23), [SS\\_MR\\_K4](#) = (5 << 23), [SS\\_MR\\_K6](#) = (6 << 23), [SS\\_MR\\_K8](#) = (7 << 23) }  
*PLL Spread Spectrum (SS) Programmable frequency modulation depth See (MR) field in the SYSPLLSSCTRL1 register in the UM.*
- enum [SS\\_MODWVCTRL\\_T](#) { [SS\\_MC\\_NOC](#) = (0 << 26), [SS\\_MC\\_RECC](#) = (2 << 26), [SS\\_MC\\_MAXC](#) = (3 << 26) }  
*PLL Spread Spectrum (SS) Modulation waveform control See (MC) field in the SYSPLLSSCTRL1 register in the UM. Compensation for low pass filtering of the PLL to get a triangular modulation at the output of the PLL, giving a flat frequency spectrum.*
- enum [PLL\\_ERROR\\_T](#) {  
[PLL\\_ERROR\\_SUCCESS](#) = 0, [PLL\\_ERROR\\_OUTPUT\\_TOO\\_LOW](#), [PLL\\_ERROR\\_OUTPUT\\_TOO\\_HIGH](#),  
[PLL\\_ERROR\\_INPUT\\_TOO\\_LOW](#),  
[PLL\\_ERROR\\_INPUT\\_TOO\\_HIGH](#), [PLL\\_ERROR\\_OUTSIDE\\_INTLIMIT](#) }  
*PLL status definitions.*

## Functions

- `__STATIC_INLINE void Chip\_Clock\_SetSystemPLLSource (CHIP\_SYSCON\_PLLCLKSRC\_T src)`  
*Set System PLL clock source.*
- `uint32_t Chip\_Clock\_GetSystemPLLInClockRate (void)`  
*Return System PLL input clock rate.*
- `uint32_t Chip\_Clock\_GetSystemPLLOutClockRate (bool recompute)`  
*Return System PLL output clock rate.*
- `void Chip\_Clock\_SetBypassPLL (bool bypass)`  
*Enables and disables PLL bypass mode.*
- `__STATIC_INLINE bool Chip\_Clock\_IsSystemPLLLocked (void)`  
*Check if PLL is locked or not.*
- `uint32_t Chip\_Clock\_GetStoredPLLClockRate (void)`  
*Get the rate of pll from the stored value.*
- `void Chip\_Clock\_SetStoredPLLClockRate (uint32_t rate)`  
*Store the current PLL rate.*
- `uint32_t Chip\_Clock\_GetSystemPLLOutFromSetup (PLL\_SETUP\_T *pSetup)`  
*Return System PLL output clock rate from setup structure.*
- `PLL\_ERROR\_T Chip\_Clock\_SetupPLLData (PLL\_CONFIG\_T *pControl, PLL\_SETUP\_T *pSetup)`  
*Set PLL output based on the passed PLL setup data.*
- `PLL\_ERROR\_T Chip\_Clock\_SetupSystemPLLPrec (PLL\_SETUP\_T *pSetup)`  
*Set PLL output from PLL setup structure (precise frequency)*
- `PLL\_ERROR\_T Chip\_Clock\_SetPLLFreq (const PLL\_SETUP\_T *pSetup)`  
*Set PLL output from PLL setup structure (precise frequency)*
- `void Chip\_Clock\_SetupSystemPLL (uint32_t multiply_by, uint32_t input_freq)`  
*Set PLL output based on the multiplier and input frequency.*

## 8.37 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/pmu\_5411x.h File Reference

### Data Structures

- struct [LPC\\_PMU\\_T](#)  
*PMU register block structure.*

### Macros

- #define [PMU\\_BOD\\_RST](#) (1 << 6)
- #define [PMU\\_BOD\\_INT](#) (1 << 7)

### Enumerations

- enum [CHIP\\_PMU\\_BODRSTLVL\\_T](#) {  
[PMU\\_BODRSTLVL\\_0](#), [PMU\\_BODRSTLVL\\_1\\_50V](#) = [PMU\\_BODRSTLVL\\_0](#), [PMU\\_BODRSTLVL\\_1](#), [PMU\\_BODRSTLVL\\_1\\_85V](#) = [PMU\\_BODRSTLVL\\_1](#),  
[PMU\\_BODRSTLVL\\_2](#), [PMU\\_BODRSTLVL\\_2\\_00V](#) = [PMU\\_BODRSTLVL\\_2](#), [PMU\\_BODRSTLVL\\_3](#), [PMU\\_BODRSTLVL\\_2\\_30V](#) = [PMU\\_BODRSTLVL\\_3](#) }
- enum [CHIP\\_PMU\\_BODRINTVAL\\_T](#) {  
[PMU\\_BODINTVAL\\_LVL0](#), [PMU\\_BODINTVAL\\_2\\_05v](#) = [PMU\\_BODINTVAL\\_LVL0](#), [PMU\\_BODINTVAL\\_LVL1](#),  
[PMU\\_BODINTVAL\\_2\\_45v](#) = [PMU\\_BODINTVAL\\_LVL1](#),  
[PMU\\_BODINTVAL\\_LVL2](#), [PMU\\_BODINTVAL\\_2\\_75v](#) = [PMU\\_BODINTVAL\\_LVL2](#), [PMU\\_BODINTVAL\\_LVL3](#), [PMU\\_BODINTVAL\\_3\\_05v](#) = [PMU\\_BODINTVAL\\_LVL3](#) }

### Functions

- `__STATIC_INLINE void Chip\_PMU\_SetBODLevels (CHIP\_PMU\_BODRSTLVL\_T rstlvl, CHIP\_PMU\_BODRINTVAL\_T intlvl)`  
*Set brown-out detection interrupt and reset levels.*
- `__STATIC_INLINE void Chip\_PMU\_EnableBODReset (void)`  
*Enable brown-out detection reset.*
- `__STATIC_INLINE void Chip\_PMU\_DisableBODReset (void)`  
*Disable brown-out detection reset.*
- `__STATIC_INLINE void Chip\_PMU\_EnableBODInt (void)`  
*Enable brown-out detection interrupt.*
- `__STATIC_INLINE void Chip\_PMU\_DisableBODInt (void)`  
*Disable brown-out detection interrupt.*

## 8.38 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/power\_lib\_5411x.h File Reference

### Macros

- #define [LPC5411X\\_ROMVER\\_0](#) (0x1100)
- #define [LPC5411X\\_ROMVER\\_1](#) (0x1101)
- #define [LPC5411X\\_ROMVER\\_2](#) (0x1102)

### Enumerations

- enum [POWER\\_MODE\\_T](#) { [POWER\\_SLEEP](#) = 0, [POWER\\_DEEP\\_SLEEP](#), [POWER\\_DEEP\\_POWER\\_DOWN](#) }

## Functions

- void [Chip\\_POWER\\_SetFROHFRate](#) (uint32\_t freq)  
*Sets the High Frequency FRO rate to (48MHz or 96MHz)*
- uint32\_t [Chip\\_POWER\\_SetPLL](#) (uint32\_t multiply\_by, uint32\_t input\_freq)  
*Sets up the System PLL given the PLL input frequency and feedback multiplier.*
- uint32\_t [Chip\\_POWER\\_SetVoltage](#) (uint32\_t desired\_freq)  
*Set optimal system voltage based on passed system frequency.*
- void [Chip\\_POWER\\_SetLowPowerVoltage](#) (uint32\_t freq)  
*Set low-power voltage levels for LP mode.*
- void [Chip\\_POWER\\_EnterPowerMode](#) (POWER\_MODE\_T mode, uint32\_t peripheral\_ctrl)  
*Enters the selected power state.*
- uint32\_t [Chip\\_POWER\\_GetROMVersion](#) (void)  
*Return ROM version.*

## 8.39 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/ring\_buffer.h File Reference

```
#include "lpc_types.h"
#include "cmsis.h"
```

## Data Structures

- struct [RINGBUFF\\_T](#)  
*Ring buffer structure.*

## Macros

- #define [RB\\_VHEAD](#)(rb) (\*(volatile uint32\_t \*) &(rb)->head)
- #define [RB\\_VTAIL](#)(rb) (\*(volatile uint32\_t \*) &(rb)->tail)

## Functions

- int [RingBuffer\\_Init](#) (RINGBUFF\_T \*RingBuff, void \*buffer, int itemSize, int count, void \*(\*cpyFunc)(void \*dst, const void \*src, uint32\_t len))  
*Initialize ring buffer.*
- \_\_STATIC\_INLINE void [RingBuffer\\_Flush](#) (RINGBUFF\_T \*RingBuff)  
*Resets the ring buffer to empty.*
- \_\_STATIC\_INLINE int [RingBuffer\\_GetSize](#) (RINGBUFF\_T \*RingBuff)  
*Return size the ring buffer.*
- \_\_STATIC\_INLINE int [RingBuffer\\_GetCount](#) (RINGBUFF\_T \*RingBuff)  
*Return number of items in the ring buffer.*
- \_\_STATIC\_INLINE int [RingBuffer\\_GetFree](#) (RINGBUFF\_T \*RingBuff)  
*Return number of free items in the ring buffer.*
- \_\_STATIC\_INLINE int [RingBuffer\\_IsFull](#) (RINGBUFF\_T \*RingBuff)  
*Return number of items in the ring buffer.*
- \_\_STATIC\_INLINE int [RingBuffer\\_IsEmpty](#) (RINGBUFF\_T \*RingBuff)  
*Return empty status of ring buffer.*
- int [RingBuffer\\_Insert](#) (RINGBUFF\_T \*RingBuff, const void \*data)  
*Insert a single item into ring buffer.*

- int [RingBuffer\\_InsertMult](#) (RINGBUFF\_T \*RingBuff, const void \*data, int num)  
*Insert an array of items into ring buffer.*
- int [RingBuffer\\_Pop](#) (RINGBUFF\_T \*RingBuff, void \*data)  
*Pop an item from the ring buffer.*
- int [RingBuffer\\_PopMult](#) (RINGBUFF\_T \*RingBuff, void \*data, int num)  
*Pop an array of items from the ring buffer.*

## 8.40 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/romapi\_5411x.h File Reference

```
#include <stdint.h>
#include "iap.h"
#include "error.h"
#include "cmsis.h"
```

### Data Structures

- struct [LPC\\_ROM\\_API\\_T](#)  
*High level ROM API structure.*

### Macros

- #define [LPC\\_ROM\\_API\\_BASE\\_LOC](#) 0x03000200UL
- #define [LPC\\_ROM\\_API](#) (\*(LPC\_ROM\_API\_T \*) [LPC\\_ROM\\_API\\_BASE\\_LOC](#))
- #define [IAP\\_ENTRY\\_LOCATION](#) 0x03000205

### Functions

- static [INLINE](#) void [iap\\_entry](#) (unsigned int cmd\_param[5], unsigned int status\_result[4])  
*LPC5410x IAP\_ENTRY API function type.*

## 8.41 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/rtc\_5411x.h File Reference

### Data Structures

- struct [LPC\\_RTC\\_T](#)  
*LPC5411X Real Time clock register block structure.*

### Macros

- #define [RTC\\_CTRL\\_SWRESET](#) (1 << 0)
- #define [RTC\\_CTRL\\_ALARM1HZ](#) (1 << 2)
- #define [RTC\\_CTRL\\_WAKE1KHZ](#) (1 << 3)
- #define [RTC\\_CTRL\\_ALARMDPD\\_EN](#) (1 << 4)
- #define [RTC\\_CTRL\\_WAKEDPD\\_EN](#) (1 << 5)
- #define [RTC\\_CTRL\\_RTC1KHZ\\_EN](#) (1 << 6)
- #define [RTC\\_CTRL\\_RTC\\_EN](#) (1 << 7)
- #define [RTC\\_CTRL\\_RTC\\_OSC\\_PD](#) (1 << 8)
- #define [RTC\\_CTRL\\_RTC\\_OSC\\_BYPASS](#) (1 << 9)
- #define [RTC\\_CTRL\\_MASK](#) ((uint32\_t) 0x0000003FD)

## Functions

- `__STATIC_INLINE void Chip_RTC_Init (LPC_RTC_T *pRTC)`  
*Initialize the RTC peripheral.*
- `__STATIC_INLINE void Chip_RTC_DeInit (LPC_RTC_T *pRTC)`  
*De-initialize the RTC peripheral.*
- `__STATIC_INLINE void Chip_RTC_EnableOptions (LPC_RTC_T *pRTC, uint32_t flags)`  
*Enable RTC options.*
- `__STATIC_INLINE void Chip_RTC_DisableOptions (LPC_RTC_T *pRTC, uint32_t flags)`  
*Disable RTC options.*
- `__STATIC_INLINE void Chip_RTC_Reset (LPC_RTC_T *pRTC)`  
*Reset RTC.*
- `__STATIC_INLINE void Chip_RTC_Enable (LPC_RTC_T *pRTC)`  
*Enables the RTC.*
- `__STATIC_INLINE void Chip_RTC_Disable (LPC_RTC_T *pRTC)`  
*Disables the RTC.*
- `__STATIC_INLINE void Chip_RTC_PowerUp (LPC_RTC_T *pRTC)`  
*Power up the RTC.*
- `__STATIC_INLINE void Chip_RTC_PowerDown (LPC_RTC_T *pRTC)`  
*Disables the RTC.*
- `__STATIC_INLINE void Chip_RTC_Enable1KHZ (LPC_RTC_T *pRTC)`  
*Enables the RTC 1KHz high resolution timer.*
- `__STATIC_INLINE void Chip_RTC_Disable1KHZ (LPC_RTC_T *pRTC)`  
*Disables the RTC 1KHz high resolution timer.*
- `__STATIC_INLINE void Chip_RTC_EnableWakeup (LPC_RTC_T *pRTC, uint32_t ints)`  
*Enables selected RTC wakeup events.*
- `__STATIC_INLINE void Chip_RTC_DisableWakeup (LPC_RTC_T *pRTC, uint32_t ints)`  
*Disables selected RTC wakeup events.*
- `__STATIC_INLINE void Chip_RTC_ClearStatus (LPC_RTC_T *pRTC, uint32_t stsMask)`  
*Clears latched RTC statuses.*
- `__STATIC_INLINE uint32_t Chip_RTC_GetStatus (LPC_RTC_T *pRTC)`  
*Return RTC control/status register.*
- `__STATIC_INLINE void Chip_RTC_SetAlarm (LPC_RTC_T *pRTC, uint32_t count)`  
*Set RTC match value for alarm status/interrupt.*
- `__STATIC_INLINE uint32_t Chip_RTC_GetAlarm (LPC_RTC_T *pRTC)`  
*Return the RTC match value used for alarm status/interrupt.*
- `__STATIC_INLINE void Chip_RTC_SetCount (LPC_RTC_T *pRTC, uint32_t count)`  
*Set RTC match count for 1 second timer count.*
- `__STATIC_INLINE uint32_t Chip_RTC_GetCount (LPC_RTC_T *pRTC)`  
*Get current RTC 1 second timer count.*
- `__STATIC_INLINE void Chip_RTC_SetWake (LPC_RTC_T *pRTC, uint16_t count)`  
*Set RTC wake count countdown value (in mS ticks)*
- `__STATIC_INLINE uint16_t Chip_RTC_GetWake (LPC_RTC_T *pRTC)`  
*Get RTC wake count countdown value.*

## 8.42 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/rtc\_ut.h File Reference

```
#include "chip.h"
#include <stdlib.h>
#include <time.h>
```

## Macros

- #define [TM\\_YEAR\\_BASE](#) (1970)
- #define [TM\\_DAYOFWEEK](#) (4)

## Functions

- void [ConvertRtcTime](#) (uint32\_t rtcTick, struct tm \*pTime)  
*Converts a RTC tick time to Universal time.*
- void [ConvertTimeRtc](#) (struct tm \*pTime, uint32\_t \*rtcTick)  
*Converts a Universal time to RTC tick time.*

## 8.43 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/sct\_5411x.h File Reference

### Data Structures

- struct [LPC\\_SCT\\_T](#)  
*State Configurable Timer register block structure.*

## Macros

- #define [CONFIG\\_SCT\\_nEV](#) (13)
- #define [CONFIG\\_SCT\\_nRG](#) (13)
- #define [CONFIG\\_SCT\\_nOU](#) (8)
- #define [CONFIG\\_SCT\\_nIN](#) (8)
- #define [SCT\\_CONFIG\\_16BIT\\_COUNTER](#) 0x00000000  
*Macro defines for SCT configuration register.*
- #define [SCT\\_CONFIG\\_32BIT\\_COUNTER](#) 0x00000001
- #define [SCT\\_CONFIG\\_CLKMODE\\_BUSCLK](#) (0x0 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_SCTCLK](#) (0x1 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_INCLK](#) (0x2 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_INEDGECLK](#) (0x3 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_SYSCLK](#) (0x0 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_PRESCALED\\_SYSCLK](#) (0x1 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_SCT\\_INPUT](#) (0x2 << 1)
- #define [SCT\\_CONFIG\\_CLKMODE\\_PRESCALED\\_SCT\\_INPUT](#) (0x3 << 1)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_0](#) (0x0UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_0](#) (0x1UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_1](#) (0x2UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_1](#) (0x3UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_2](#) (0x4UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_2](#) (0x5UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_3](#) (0x6UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_3](#) (0x7UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_4](#) (0x8UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_4](#) (0x9UL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_5](#) (0xAUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_5](#) (0xBUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_6](#) (0xCUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_6](#) (0xDUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_RISING\\_IN\\_7](#) (0xEUL << 3)
- #define [SCT\\_CONFIG\\_CKSEL\\_FALLING\\_IN\\_7](#) (0xFUL << 3)

- #define `SCT_CONFIG_NOLOADL_U` (0x1 << 7)
- #define `SCT_CONFIG_NOLOADH` (0x1 << 8)
- #define `SCT_CONFIG_AUTOLIMIT_U` (0x1UL << 17)
- #define `SCT_CONFIG_AUTOLIMIT_L` (0x1UL << 17)
- #define `SCT_CONFIG_AUTOLIMIT_H` (0x1UL << 18)
- #define `COUNTUP_TO_LIMIT_THEN_CLEAR_TO_ZERO` 0

*Macro defines for SCT control register.*

- #define `COUNTUP_TO_LIMIT_THEN_COUNTDOWN_TO_ZERO` 1
- #define `SCT_CTRL_STOP_L` (1 << 1)
- #define `SCT_CTRL_HALT_L` (1 << 2)
- #define `SCT_CTRL_CLRCTR_L` (1 << 3)
- #define `SCT_CTRL_BIDIR_L(x)` (((x) & 0x01) << 4)
- #define `SCT_CTRL_PRE_L(x)` (((x) & 0xFF) << 5)
- #define `COUNTUP_TO_LIMIT_THEN_CLEAR_TO_ZERO` 0

*Macro defines for SCT control register.*

- #define `COUNTUP_TO_LIMIT_THEN_COUNTDOWN_TO_ZERO` 1
- #define `SCT_CTRL_STOP_H` (1 << 17)
- #define `SCT_CTRL_HALT_H` (1 << 18)
- #define `SCT_CTRL_CLRCTR_H` (1 << 19)
- #define `SCT_CTRL_BIDIR_H(x)` (((x) & 0x01) << 20)
- #define `SCT_CTRL_PRE_H(x)` (((x) & 0xFF) << 21)
- #define `SCT_EV_CTRL_MATCHSEL(reg)` (reg << 0)
- #define `SCT_EV_CTRL_HEVENT_L` (0UL << 4)
- #define `SCT_EV_CTRL_HEVENT_H` (1UL << 4)
- #define `SCT_EV_CTRL_OUTSEL_INPUT` (0UL << 5)
- #define `SCT_EV_CTRL_OUTSEL_OUTPUT` (0UL << 5)
- #define `SCT_EV_CTRL_IOSEL(signal)` (signal << 6)
- #define `SCT_EV_CTRL_IOCOND_LOW` (0UL << 10)
- #define `SCT_EV_CTRL_IOCOND_RISE` (0x1UL << 10)
- #define `SCT_EV_CTRL_IOCOND_FALL` (0x2UL << 10)
- #define `SCT_EV_CTRL_IOCOND_HIGH` (0x3UL << 10)
- #define `SCT_EV_CTRL_COMBMODE_OR` (0x0UL << 12)
- #define `SCT_EV_CTRL_COMBMODE_MATCH` (0x1UL << 12)
- #define `SCT_EV_CTRL_COMBMODE_IO` (0x2UL << 12)
- #define `SCT_EV_CTRL_COMBMODE_AND` (0x3UL << 12)
- #define `SCT_EV_CTRL_STATELD` (0x1UL << 14)
- #define `SCT_EV_CTRL_STATEV(x)` (x << 15)
- #define `SCT_EV_CTRL_MATCHMEM` (0x1UL << 20)
- #define `SCT_EV_CTRL_DIRECTION_INDEPENDENT` (0x0UL << 21)
- #define `SCT_EV_CTRL_DIRECTION_UP` (0x1UL << 21)
- #define `SCT_EV_CTRL_DIRECTION_DOWN` (0x2UL << 21)
- #define `SCT_RES_NOCHANGE` (0)

*Macro defines for SCT Conflict resolution register.*

- #define `SCT_RES_SET_OUTPUT` (1)
- #define `SCT_RES_CLEAR_OUTPUT` (2)
- #define `SCT_RES_TOGGLE_OUTPUT` (3)



## Enumerations

- enum [CHIP\\_SCT\\_MATCH\\_REG\\_T](#) {  
[SCT\\_MATCH\\_0](#) = 0, [SCT\\_MATCH\\_1](#), [SCT\\_MATCH\\_2](#), [SCT\\_MATCH\\_3](#),  
[SCT\\_MATCH\\_4](#), [SCT\\_MATCH\\_5](#), [SCT\\_MATCH\\_6](#), [SCT\\_MATCH\\_7](#),  
[SCT\\_MATCH\\_8](#), [SCT\\_MATCH\\_9](#), [SCT\\_MATCH\\_10](#), [SCT\\_MATCH\\_11](#),  
[SCT\\_MATCH\\_12](#), [SCT\\_MATCH\\_13](#), [SCT\\_MATCH\\_14](#), [SCT\\_MATCH\\_15](#) }
- enum [CHIP\\_SCT\\_EVENT\\_T](#) {  
[SCT\\_EVT\\_0](#) = (1 << 0), [SCT\\_EVT\\_1](#) = (1 << 1), [SCT\\_EVT\\_2](#) = (1 << 2), [SCT\\_EVT\\_3](#) = (1 << 3),  
[SCT\\_EVT\\_4](#) = (1 << 4), [SCT\\_EVT\\_5](#) = (1 << 5), [SCT\\_EVT\\_6](#) = (1 << 6), [SCT\\_EVT\\_7](#) = (1 << 7),  
[SCT\\_EVT\\_8](#) = (1 << 8), [SCT\\_EVT\\_9](#) = (1 << 9), [SCT\\_EVT\\_10](#) = (1 << 10), [SCT\\_EVT\\_11](#) = (1 << 11),  
[SCT\\_EVT\\_12](#) = (1 << 12), [SCT\\_EVT\\_13](#) = (1 << 13), [SCT\\_EVT\\_14](#) = (1 << 14), [SCT\\_EVT\\_15](#) = (1 << 15) }

## Functions

- \_\_STATIC\_INLINE void [Chip\\_SCT\\_EventControl](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint32\_t event\_number, uint32\_t value)  
Set event control register.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_EventStateMask](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint32\_t event\_number, uint32\_t event\_state\_mask)  
Set event state mask register.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_Config](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint32\_t cfg)  
Set configuration register.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_Limit](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint32\_t value)  
Configures the Limit register.
- void [Chip\\_SCT\\_SetClrControl](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint32\_t value, [FunctionalState](#) ena)  
Set or Clear the Control register.
- void [Chip\\_SCT\\_SetConflictResolution](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint8\_t outnum, uint8\_t value)  
Set the conflict resolution.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_SetCount](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint32\_t count)  
Set unified count value in State Configurable Timer.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_SetCountL](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint16\_t count)  
Set lower count value in State Configurable Timer.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_SetCountH](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint16\_t count)  
Set higher count value in State Configurable Timer.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_SetMatchCount](#) ([LPC\\_SCT\\_T](#) \*pSCT, [CHIP\\_SCT\\_MATCH\\_REG\\_T](#) n, uint32\_t value)  
Set unified match count value in State Configurable Timer.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_SetMatchReload](#) ([LPC\\_SCT\\_T](#) \*pSCT, [CHIP\\_SCT\\_MATCH\\_REG\\_T](#) n, uint32\_t value)  
Set unified match reload count value in State Configurable Timer.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_EnableEventInt](#) ([LPC\\_SCT\\_T](#) \*pSCT, [CHIP\\_SCT\\_EVENT\\_T](#) evt)  
Enable the interrupt for the specified event in State Configurable Timer.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_DisableEventInt](#) ([LPC\\_SCT\\_T](#) \*pSCT, [CHIP\\_SCT\\_EVENT\\_T](#) evt)  
Disable the interrupt for the specified event in State Configurable Timer.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_ClearEventFlag](#) ([LPC\\_SCT\\_T](#) \*pSCT, [CHIP\\_SCT\\_EVENT\\_T](#) evt)  
Clear the specified event flag in State Configurable Timer.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_SetControl](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint32\_t value)  
Set control register in State Configurable Timer.
- \_\_STATIC\_INLINE void [Chip\\_SCT\\_ClearControl](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint32\_t value)  
Clear control register in State Configurable Timer.
- void [Chip\\_SCT\\_Init](#) ([LPC\\_SCT\\_T](#) \*pSCT)

*Initializes the State Configurable Timer.*

- void [Chip\\_SCT\\_DeInit](#) ([LPC\\_SCT\\_T](#) \*pSCT)

*Deinitializes the State Configurable Timer.*

## 8.44 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/sct\_pwm\_5411x.h File Reference

### Functions

- `__STATIC_INLINE uint32_t Chip\_SCTPWM\_GetTicksPerCycle (LPC\_SCT\_T *pSCT)`  
*Get number of ticks per PWM cycle.*
- `__STATIC_INLINE uint32_t Chip\_SCTPWM\_PercentageToTicks (LPC\_SCT\_T *pSCT, uint8_t percent)`  
*Converts a percentage to ticks.*
- `__STATIC_INLINE uint32_t Chip\_SCTPWM\_GetDutyCycle (LPC\_SCT\_T *pSCT, uint8_t index)`  
*Get number of ticks on per PWM cycle.*
- `__STATIC_INLINE void Chip\_SCTPWM\_SetDutyCycle (LPC\_SCT\_T *pSCT, uint8_t index, uint32_t ticks)`  
*Get number of ticks on per PWM cycle.*
- `__STATIC_INLINE void Chip\_SCTPWM\_Init (LPC\_SCT\_T *pSCT)`  
*Initialize the SCT/PWM clock and reset.*
- `__STATIC_INLINE void Chip\_SCTPWM\_Start (LPC\_SCT\_T *pSCT)`  
*Start the SCT PWM.*
- `__STATIC_INLINE void Chip\_SCTPWM\_Stop (LPC\_SCT\_T *pSCT)`  
*Stop the SCT PWM.*
- void [Chip\\_SCTPWM\\_SetRate](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint32\_t freq)  
*Sets the frequency of the generated PWM wave.*
- void [Chip\\_SCTPWM\\_SetOutPin](#) ([LPC\\_SCT\\_T](#) \*pSCT, uint8\_t index, uint8\_t pin)  
*Setup the OUTPUT pin and associate it with an index.*

## 8.45 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/spi\_common\_5411x.h File Reference

### Data Structures

- struct [LPC\\_SPI\\_T](#)  
*SPI register block structure.*
- struct [SPI\\_CFGSETUP\\_T](#)

### Macros

- `#define SPI\_CFG\_BITMASK (0x0FBD) /** SPI register bit mask */`
- `#define SPI\_CFG\_SPI\_EN (1 << 0) /** SPI Slave Mode Select */`
- `#define SPI\_CFG\_SLAVE\_EN (0 << 0) /** SPI Master Mode Select */`
- `#define SPI\_CFG\_MASTER\_EN (1 << 2) /** SPI MSB First mode enable */`
- `#define SPI\_CFG\_MSB\_FIRST\_EN (0 << 3) /** SPI LSB First mode enable */`
- `#define SPI\_CFG\_LSB\_FIRST\_EN (1 << 3) /** SPI Clock Phase Select */`
- `#define SPI\_CFG\_CPHA\_FIRST (0 << 4) /** Capture data on the first edge, Change data on the following edge */`
- `#define SPI\_CFG\_CPHA\_SECOND (1 << 4) /** SPI Clock Polarity Select */`
- `#define SPI\_CFG\_CPOL\_LO (0 << 5) /** The rest state of the clock (between frames) is low. */`
- `#define SPI\_CFG\_CPOL\_HI (1 << 5) /** The rest state of the clock (between frames) is high. */`
- `#define SPI\_CFG\_LBM\_EN (1 << 7) /** SPI control 1 loopback mode enable */`
- `#define SPI\_CFG\_SPOL\_LO (0 << 8) /** SPI SSEL0 Polarity Select */`

- #define `SPI_CFG_SPOL_HI` (1 << 8) /\*\* SSEL0 is active High \*/
- #define `SPI_CFG_SPOLNUM_HI`(n) (1 << ((n) + 8)) /\*\* SSELN is active High, selects 0 - 3 \*/
- #define `SPI_DLY_BITMASK` (0xFFFF) /\*\* SPI DLY Register Mask \*/
- #define `SPI_DLY_PRE_DELAY`(n) (((n) & 0x0F) << 0) /\*\* Time in SPI clocks between SSEL assertion and the beginning of a data frame \*/
- #define `SPI_DLY_POST_DELAY`(n) (((n) & 0x0F) << 4) /\*\* Time in SPI clocks between the end of a data frame and SSEL deassertion. \*/
- #define `SPI_DLY_FRAME_DELAY`(n) (((n) & 0x0F) << 8) /\*\* Minimum time in SPI clocks between adjacent data frames. \*/
- #define `SPI_DLY_TRANSFER_DELAY`(n) (((n) & 0x0F) << 12) /\*\* Minimum time in SPI clocks that the SSEL is deasserted between transfers. \*/
- #define `SPI_STAT_BITMASK` (0x01F0) /\*\* SPI STAT Register BitMask \*/
- #define `SPI_STAT_SSA` (1 << 4) /\*\* Slave Select Assert \*/
- #define `SPI_STAT_SSD` (1 << 5) /\*\* Slave Select Deassert \*/
- #define `SPI_STAT_STALLED` (1 << 6) /\*\* Stalled status flag \*/
- #define `SPI_STAT_EOT` (1 << 7) /\*\* End Transfer flag \*/
- #define `SPI_STAT_MSTIDLE` (1 << 8) /\*\* Idle status flag \*/
- #define `SPI_INT_BITMASK` (0x0130) /\*\* SPI interrupt Enable/Disable bits \*/
- #define `SPI_INT_SSAEN` (1 << 4) /\*\* Slave Select is asserted interrupt [BIT-4 of `INTENSET/INTENCLR/INTSTAT` register] \*/
- #define `SPI_INT_SSDEN` (1 << 5) /\*\* Slave Select is deasserted interrupt [BIT-5 of `INTENSET/INTENCLR/INTSTAT` register] \*/
- #define `SPI_INT_MSTIDLE` (1 << 8) /\*\* SPI master is Idle [BIT-8 of `INTENSET/INTENCLR/INTSTAT` register] \*/
- #define `SPI_FIFOCFG_ENABLETX` (1 << 0)  
*SPI FIFO Configuration register bits.*
- #define `SPI_FIFOCFG_ENABLERX` (1 << 1)
- #define `SPI_FIFOCFG_DMATX` (1 << 12)
- #define `SPI_FIFOCFG_DMARX` (1 << 13)
- #define `SPI_FIFOCFG_WAKETX` (1 << 14)
- #define `SPI_FIFOCFG_WAKERX` (1 << 15)
- #define `SPI_FIFOCFG_EMPTYTX` (1 << 16)
- #define `SPI_FIFOCFG_EMPTYRX` (1 << 17)
- #define `SPI_FIFO_DEPTH` (8) /\*\* SPI-FIFO How many entries are in the FIFO \*/  
*Macro defines for FIFO Status register.*
- #define `SPI_FIFOSTAT_BITMASK` (0x1F1FFB) /\*\* SPI-FIFO STAT Register BitMask \*/
- #define `SPI_FIFOSTAT_TXERR` (1 << 0) /\*\* SPI-FIFO transmit error \*/
- #define `SPI_FIFOSTAT_RXERR` (1 << 1) /\*\* SPI-FIFO receive error \*/
- #define `SPI_FIFOSTAT_PERINT` (1 << 3) /\*\* SPI-FIFO peripheral (SPI) interrupt \*/
- #define `SPI_FIFOSTAT_TXEMPTY` (1 << 4) /\*\* SPI-FIFO transmitter empty \*/
- #define `SPI_FIFOSTAT_TXNOTFULL` (1 << 5) /\*\* SPI-FIFO transmitter not full \*/
- #define `SPI_FIFOSTAT_RXNOTEMPTY` (1 << 6) /\*\* SPI-FIFO receiver not empty \*/
- #define `SPI_FIFOSTAT_RXFULL` (1 << 7) /\*\* SPI-FIFO receiver not full \*/
- #define `SPI_FIFOSTAT_TXLVL`(val) (((val) >> 8) & 0x1F) /\*\* SPI-FIFO extract transmit level \*/
- #define `SPI_FIFOSTAT_RXLVL`(val) (((val) >> 16) & 0x1F) /\*\* SPI-FIFO extract receive level \*/
- #define `SPI_FIFOTRIG_BITMASK` (0x000f0f03) /\*\* SPI FIFO trigger settings Register BitMask \*/  
*UART FIFO trigger settings register defines.*
- #define `SPI_FIFOTRIG_TXLVLENA` (1 << 0)
- #define `SPI_FIFOTRIG_RXLVLENA` (1 << 1)
- #define `SPI_FIFOTRIG_TXLVL`(lvl) ((lvl & 0x0f) << 8)
- #define `SPI_FIFOTRIG_RXLVL`(lvl) ((lvl & 0x0f) << 16)
- #define `SPI_FIFOTRIG_TXLVL_DEFAULT` 4
- #define `SPI_FIFOTRIG_RXLVL_DEFAULT` 0
- #define `SPI_FIFOINT_BITMASK` (0x001F) /\*\* FIFO interrupt Bit mask \*/

Macro defines for SPI Interrupt enable/disable/status [INTSET/INTCLR/INTSTAT and FIFOINTSET/FIFOINTCLR/←  
FIFOINTSTAT registers].

- #define SPI\_FIFOINT\_TXERR (1 << 0) /\*\* TX error interrupt [BIT-0 of FIFOINTENSET/FIFOINTENCL←  
R/FIFOINTSTAT register] \*/
- #define SPI\_FIFOINT\_RXERR (1 << 1) /\*\* RX error interrupt [BIT-1 of FIFOINTENSET/FIFOINTENCL←  
R/FIFOINTSTAT register] \*/
- #define SPI\_FIFOINT\_TXLVL (1 << 2) /\*\* TX FIFO ready interrupt [BIT-2 of FIFOINTENSET/FIFOINTE←  
NCLR/FIFOINTSTAT register] \*/
- #define SPI\_FIFOINT\_RXLVL (1 << 3) /\*\* RX Data ready interrupt [BIT-3 of FIFOINTENSET/FIFOINTE←  
NCLR/FIFOINTSTAT register] \*/
- #define SPI\_FIFOINT\_PERINT (1 << 4) /\*\* SPI peripheral interrupt [BIT-4 of FIFOINTSTAT register] \*/
- #define SPI\_RXDAT\_BITMASK (0x1FFFFFF) /\*\* SPI RXDAT Register BitMask \*/
- #define SPI\_RXDAT\_DATA(n) ((n) & 0xFFFF) /\*\* Receiver Data \*/
- #define SPI\_RXDAT\_RXSSELN(n) ((~((n >> 16) & 0x0f)) & 0x0f) /\*\* Determine the active SSEL pin \*/
- #define SPI\_RXDAT\_RXSSELN\_ACTIVE (0 << 16) /\*\* The state of SSEL pin is active \*/
- #define SPI\_RXDAT\_SOT (1 << 20) /\*\* Start of Transfer flag \*/
- #define SPI\_TXDAT\_BITMASK (0xF7FFFFFF) /\*\* SPI TXDATCTL Register BitMask \*/
- #define SPI\_TXDAT\_DATA(n) ((n) & 0xFFFF) /\*\* SPI Transmit Data \*/
- #define SPI\_TXDAT\_CTRLMASK (0xF7F) /\*\* SPI TXDATCTL Register BitMask for control bits only \*/
- #define SPI\_TXDAT\_ASSERT\_SSEL (0) /\*\* Assert SSEL0 pin \*/
- #define SPI\_TXDAT\_ASSERTNUM\_SSEL(n) ((~(1 << (n))) & 0x0f) /\*\* Assert SSELN pin \*/
- #define SPI\_TXDAT\_DEASSERT\_SSEL (1) /\*\* Deassert SSEL0 pin \*/
- #define SPI\_TXDAT\_DEASSERTNUM\_SSEL(n) (1 << (n)) /\*\* Deassert SSELN pin \*/
- #define SPI\_TXDAT\_DEASSERT\_ALL (0xF) /\*\* Deassert all SSEL pins \*/
- #define SPI\_TXDAT\_EOT (1 << 4) /\*\* End of Transfer flag (TRANSFER\_DELAY is applied after sending  
the current frame) \*/
- #define SPI\_TXDAT\_EOF (1 << 5) /\*\* End of Frame flag (FRAME\_DELAY is applied after sending the  
current part) \*/
- #define SPI\_TXDAT\_RXIGNORE (1 << 6) /\*\* Receive Ignore Flag \*/
- #define SPI\_TXDAT\_FLEN(n) (((n) & 0x0F) << 8) /\*\* Frame length - 1 \*/
- #define SPI\_TXDAT\_FLENMASK (0xF << 8) /\*\* Frame length mask \*/
- #define SPI\_TXDAT\_DATA(n) ((n) & 0xFFFF) /\*\* SPI Transmit Data \*/
- #define SPI\_DIV\_VAL(n) ((n) & 0xFFFF) /\*\* Rate divider value mask (In Master Mode only)\*/
- #define Chip\_SPI\_ReadFIFO Chip\_SPI\_ReadRawRXFifo
- #define Chip\_SPI\_ReadFIFOdata Chip\_SPI\_ReadRXData
- #define Chip\_SPI\_WriteFIFOcd Chip\_SPI\_SetTXCTRLData
- #define Chip\_SPI\_WriteFIFOdata Chip\_SPI\_WriteTXData
- #define Chip\_SPI\_FlushFIFOs Chip\_SPI\_FlushFifos

## Enumerations

- enum ROM\_SPI\_CLOCK\_MODE\_T {  
ROM\_SPI\_CLOCK\_CPHA0\_CPOL0 = 0, ROM\_SPI\_CLOCK\_MODE0 = ROM\_SPI\_CLOCK\_CPHA0\_CP←  
OL0, ROM\_SPI\_CLOCK\_CPHA1\_CPOL0 = 1, ROM\_SPI\_CLOCK\_MODE1 = ROM\_SPI\_CLOCK\_CPHA1←  
\_CPOL0,  
ROM\_SPI\_CLOCK\_CPHA0\_CPOL1 = 2, ROM\_SPI\_CLOCK\_MODE2 = ROM\_SPI\_CLOCK\_CPHA0\_CP←  
OL1, ROM\_SPI\_CLOCK\_CPHA1\_CPOL1 = 3, ROM\_SPI\_CLOCK\_MODE3 = ROM\_SPI\_CLOCK\_CPHA1←  
\_CPOL1 }

*SPI Clock Mode.*

- enum SPI\_CLOCK\_MODE\_T {  
SPI\_CLOCK\_CPHA0\_CPOL0 = SPI\_CFG\_CPOL\_LO | SPI\_CFG\_CPHA\_FIRST, SPI\_CLOCK\_MODE0 =  
SPI\_CLOCK\_CPHA0\_CPOL0, SPI\_CLOCK\_CPHA1\_CPOL0 = SPI\_CFG\_CPOL\_LO | SPI\_CFG\_CPHA←  
SECOND, SPI\_CLOCK\_MODE1 = SPI\_CLOCK\_CPHA1\_CPOL0,  
SPI\_CLOCK\_CPHA0\_CPOL1 = SPI\_CFG\_CPOL\_HI | SPI\_CFG\_CPHA\_FIRST, SPI\_CLOCK\_MODE2 =  
SPI\_CLOCK\_CPHA0\_CPOL1, SPI\_CLOCK\_CPHA1\_CPOL1 = SPI\_CFG\_CPOL\_HI | SPI\_CFG\_CPHA←  
\_SECOND, SPI\_CLOCK\_MODE3 = SPI\_CLOCK\_CPHA1\_CPOL1 }

*SPI Clock Mode.*

## Functions

- int [Chip\\_SPI\\_Init](#) (LPC\_SPI\_T \*pSPI)  
*Initialize the SPI.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_DeInit](#) (LPC\_SPI\_T \*pSPI)  
*Disable SPI operation.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_SetCFGRegBits](#) (LPC\_SPI\_T \*pSPI, uint32\_t bits)  
*Set SPI CFG register values.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_ClearCFGRegBits](#) (LPC\_SPI\_T \*pSPI, uint32\_t bits)  
*Clear SPI CFG register values.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_Enable](#) (LPC\_SPI\_T \*pSPI)  
*Enable SPI peripheral.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_Disable](#) (LPC\_SPI\_T \*pSPI)  
*Disable SPI peripheral.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_EnableSlaveMode](#) (LPC\_SPI\_T \*pSPI)  
*Enable SPI slave mode.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_EnableLSBFirst](#) (LPC\_SPI\_T \*pSPI)  
*Enable LSB First transfers.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_EnableMSBFirst](#) (LPC\_SPI\_T \*pSPI)  
*Enable MSB First transfers.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_SetSPIMode](#) (LPC\_SPI\_T \*pSPI, SPI\_CLOCK\_MODE\_T mode)  
*Set SPI mode.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_SetCSPolHigh](#) (LPC\_SPI\_T \*pSPI, uint8\_t csNum)  
*Set polarity on the SPI chip select high.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_SetCSPolLow](#) (LPC\_SPI\_T \*pSPI, uint8\_t csNum)  
*Set polarity on the SPI chip select low.*
- void [Chip\\_SPI\\_ConfigureSPI](#) (LPC\_SPI\_T \*pSPI, SPI\_CFGSETUP\_T \*pCFG)  
*Setup SPI configuration.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SPI\\_GetStatus](#) (LPC\_SPI\_T \*pSPI)  
*Get the current status of SPI controller.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_ClearStatus](#) (LPC\_SPI\_T \*pSPI, uint32\_t Flag)  
*Clear SPI status.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_EnableInts](#) (LPC\_SPI\_T \*pSPI, uint32\_t intMask)  
*Enable a SPI interrupt.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_DisableInts](#) (LPC\_SPI\_T \*pSPI, uint32\_t intMask)  
*Disable a SPI interrupt.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SPI\\_GetEnabledInts](#) (LPC\_SPI\_T \*pSPI)  
*Return enabled SPI interrupts.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SPI\\_GetPendingInts](#) (LPC\_SPI\_T \*pSPI)  
*Return pending SPI interrupts.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_SetFIFOCfg](#) (LPC\_SPI\_T \*pSPI, uint32\_t cfg)  
*Set FIFO Configuration register.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_ClearFIFOCfg](#) (LPC\_SPI\_T \*pSPI, uint32\_t cfg)  
*Clear FIFO Configuration register.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SPI\\_GetFIFOStatus](#) (LPC\_SPI\_T \*pSPI)  
*Get the current status of SPI controller FIFO.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_ClearFIFOStatus](#) (LPC\_SPI\_T \*pSPI, uint32\_t mask)  
*Clear the FIFO status register.*
- `__STATIC_INLINE` void [Chip\\_SPI\\_SetFIFOTrigLevel](#) (LPC\_SPI\_T \*pSPI, uint8\_t tx\_lvl, uint8\_t rx\_lvl)  
*Setup SPI FIFO trigger-level.*
- `__STATIC_INLINE` uint32\_t [Chip\\_SPI\\_GetFIFOTrigLevel](#) (LPC\_SPI\_T \*pSPI)

- *Get SPI FIFO trigger-level.*
- `__STATIC_INLINE void Chip_SPI_EnableFIFOInts (LPC_SPI_T *pSPI, uint32_t intMask)`  
*Enable a SPI FIFO interrupt.*
- `__STATIC_INLINE void Chip_SPI_DisableFIFOInts (LPC_SPI_T *pSPI, uint32_t intMask)`  
*Disable a SPI FIFO interrupt.*
- `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOEnabledInts (LPC_SPI_T *pSPI)`  
*Return enabled SPI FIFO interrupts.*
- `__STATIC_INLINE uint32_t Chip_SPI_GetFIFOPendingInts (LPC_SPI_T *pSPI)`  
*Return pending SPI FIFO interrupts.*
- `__STATIC_INLINE uint32_t Chip_SPI_ReadRawRXFifo (LPC_SPI_T *pSPI)`  
*Read raw data from receive FIFO with status bits.*
- `__STATIC_INLINE uint16_t Chip_SPI_ReadRXData (LPC_SPI_T *pSPI)`  
*Read data from receive FIFO masking off status bits.*
- `__STATIC_INLINE void Chip_SPI_WriteFIFO (LPC_SPI_T *pSPI, uint32_t data)`  
*Write FIFOWR register: writes 32-bit value to FIFO.*
- `__STATIC_INLINE void Chip_SPI_SetTXCTRLData (LPC_SPI_T *pSPI, uint16_t ctrl, uint16_t data)`  
*Write FIFOWR register: writes control options and data.*
- `__STATIC_INLINE void Chip_SPI_WriteTXData (LPC_SPI_T *pSPI, uint16_t data)`  
*Write data to transmit FIFO.*
- `__STATIC_INLINE void Chip_SPI_FlushFifos (LPC_SPI_T *pSPI)`  
*Flush FIFOs.*

## 8.46 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/spim\_5411x.h File Reference

```
#include "spi_common_5411x.h"
```

### Data Structures

- struct `SPIM_DELAY_CONFIG_T`  
*SPI Delay Configure Struct.*
- struct `SPIM_XFER_T`  
*SPI Master transfer data context.*

### Macros

- `#define SPIM_XFER_OPT_FRAME_DLY (1 << 0) /* frame delay between frames */`
- `#define SPIM_XFER_OPT_FRAME_ASSERT (1 << 1) /* assert/de-assert ssel for each frame */`
- `#define SPIM_XFER_OPT_DMA (1 << 2) /* use DMA */`

### Enumerations

- enum `SPIM_EVENT_T` {  
  `SPIM_EVENT_WAIT`, `SPIM_EVENT_ERRORRX`, `SPIM_EVENT_ERRORRX`, `SPIM_EVENT_ERROR`,  
  `SPIM_EVENT_DONE` }  
*SPI master Xfer events.*
- enum `SPIM_XFER_STATE_T` {  
  `SPI_XFER_STATE_IDLE`, `SPI_XFER_STATE_BUSY`, `SPI_XFER_STATE_DONE`, `SPI_XFER_STATE_S←`  
  `TALL`,  
  `SPI_XFER_STATE_ERROR` }  
*States of SPI Master Xfer.*

## Functions

- `__STATIC_INLINE uint32_t Chip_SPIM_GetClockRate (LPC_SPI_T *pSPI)`  
*Get SPI master bit rate.*
- `uint32_t Chip_SPIM_SetClockRate (LPC_SPI_T *pSPI, uint32_t rate)`  
*Set SPI master bit rate.*
- `void Chip_SPIM_DelayConfig (LPC_SPI_T *pSPI, SPIM_DELAY_CONFIG_T *pConfig)`  
*Config SPI Delay parameters.*
- `__STATIC_INLINE void Chip_SPIM_ForceEndOfTransfer (LPC_SPI_T *pSPI)`  
*Forces an end of transfer for the current master transfer.*
- `__STATIC_INLINE void Chip_SPIM_EnableLoopBack (LPC_SPI_T *pSPI)`  
*Enable loopback mode.*
- `__STATIC_INLINE void Chip_SPIM_DisableLoopBack (LPC_SPI_T *pSPI)`  
*Disable loopback mode.*
- `void Chip_SPIM_XferHandler (LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)`  
*SPI master transfer state change handler.*
- `void Chip_SPIM_Xfer (LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)`  
*Start non-blocking SPI master transfer.*
- `void Chip_SPIM_XferFIFO (LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)`  
*Start polled SPI master transfer.*
- `void Chip_SPIM_XferBlocking (LPC_SPI_T *pSPI, SPIM_XFER_T *xfer)`  
*Perform blocking SPI master transfer.*

## 8.47 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/spis\_5411x.h File Reference

```
#include "spi_common_5411x.h"
```

## Data Structures

- struct `SPIS_XFER_T`

## Enumerations

- enum `SPIS_EVENT_T` {  
    `SPIS_EVENT_SASSERT`, `SPIS_EVENT_SDEASSERT`, `SPIS_EVENT_DONE`, `SPIS_EVENT_ERRORTX`,  
    `SPIS_EVENT_ERRORRX`, `SPIS_EVENT_THRESHOLD` }  
*Slave callback events.*

## Functions

- `void Chip_SPIS_Init (LPC_SPI_T *pSPI)`  
*SPI slave initialization.*
- `void Chip_SPIS_EnableInts (LPC_SPI_T *pSPI)`  
*SPI slave interrupt enable.*
- `void Chip_SPIS_DisableInts (LPC_SPI_T *pSPI)`  
*SPI slave interrupt disable.*
- `void Chip_SPIS_LoadFIFO (LPC_SPI_T *pSPI, SPIS_XFER_T *xfer)`  
*Load slave transmit FIFO.*
- `void Chip_SPIS_ReadFIFO (LPC_SPI_T *pSPI, SPIS_XFER_T *xfer)`



*SPI slave FIFO read.*

- void [Chip\\_SPIS\\_XferHandler](#) (LPC\_SPI\_T \*pSPI, SPIS\_XFER\_T \*xfer)

*SPI slave transfer state change handler.*

## 8.48 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/stopwatch.h File Reference

```
#include "cmsis.h"
```

### Functions

- void [StopWatch\\_Init](#) (void)  
*Initialize stopwatch.*
- uint32\_t [StopWatch\\_Start](#) (void)  
*Start a stopwatch.*
- \_\_STATIC\_INLINE uint32\_t [StopWatch\\_Elapsed](#) (uint32\_t startTime)  
*Returns number of ticks elapsed since stopwatch was started.*
- uint32\_t [StopWatch\\_TicksPerSecond](#) (void)  
*Returns number of ticks per second of the stopwatch timer.*
- uint32\_t [StopWatch\\_TicksToMs](#) (uint32\_t ticks)  
*Converts from stopwatch ticks to mS.*
- uint32\_t [StopWatch\\_TicksToUs](#) (uint32\_t ticks)  
*Converts from stopwatch ticks to uS.*
- uint32\_t [StopWatch\\_MsToTicks](#) (uint32\_t mS)  
*Converts from mS to stopwatch ticks.*
- uint32\_t [StopWatch\\_UsToTicks](#) (uint32\_t uS)  
*Converts from uS to stopwatch ticks.*
- \_\_STATIC\_INLINE void [StopWatch\\_DelayTicks](#) (uint32\_t ticks)  
*Delays the given number of ticks using stopwatch primitives.*
- \_\_STATIC\_INLINE void [StopWatch\\_DelayMs](#) (uint32\_t mS)  
*Delays the given number of mS using stopwatch primitives.*
- \_\_STATIC\_INLINE void [StopWatch\\_DelayUs](#) (uint32\_t uS)  
*Delays the given number of uS using stopwatch primitives.*

## 8.49 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/syscon\_5411x.h File Reference

### Data Structures

- struct [LPC\\_SYSCON\\_T](#)  
*LPC5411X Main system configuration register block structure.*
- struct [LPC\\_ASYNC\\_SYSCON\\_T](#)  
*LPC5411X Asynchronous system configuration register block structure.*



## Macros

- #define `SYSCON_FROCTRL_MASK`  $((1 \ll 15) \mid (0xF \ll 26))$   
*FROCTRL register bits.*
- #define `SYSCON_FROCTRL_WRTRIM`  $(1UL \ll 31)$
- #define `SYSCON_FROCTRL_HSPDCLK`  $(1UL \ll 30)$
- #define `SYSCON_FROCTRL_USBMODCHG`  $(1UL \ll 25)$
- #define `SYSCON_FROCTRL_USBCLKADJ`  $(1UL \ll 24)$
- #define `SYSCON_FROCTRL_SEL96MHZ`  $(1UL \ll 14)$
- #define `SYSCON_NMISRC_M0_ENABLE`  $((uint32_t) 1 \ll 30)$
- #define `SYSCON_NMISRC_M4_ENABLE`  $((uint32_t) 1 \ll 31)$
- #define `SYSCON_RST_POR`  $(1 \ll 0)$
- #define `SYSCON_RST_EXTRST`  $(1 \ll 1)$
- #define `SYSCON_RST_WDT`  $(1 \ll 2)$
- #define `SYSCON_RST_BOD`  $(1 \ll 3)$
- #define `SYSCON_RST_SYSRST`  $(1 \ll 4)$
- #define `SYSCON_PDRUNCFG_PD_FRO`  $(1 \ll 4)$
- #define `SYSCON_PDRUNCFG_PD_FLASH`  $(1 \ll 5)$
- #define `SYSCON_PDRUNCFG_PD_TS`  $(1 \ll 6)$
- #define `SYSCON_PDRUNCFG_PD_BOD_RST`  $(1 \ll 7)$
- #define `SYSCON_PDRUNCFG_PD_BOD_INTR`  $(1 \ll 8)$
- #define `SYSCON_PDRUNCFG_PD_ADC0`  $(1 \ll 10)$
- #define `SYSCON_PDRUNCFG_PD_VDDFLASH`  $(1 \ll 11)$
- #define `SYSCON_PDRUNCFG_LP_VDDFLASH`  $(1 \ll 12)$
- #define `SYSCON_PDRUNCFG_PD_SRAM0`  $(1 \ll 13)$
- #define `SYSCON_PDRUNCFG_PD_SRAM1`  $(1 \ll 14)$
- #define `SYSCON_PDRUNCFG_PD_SRAM2`  $(1 \ll 15)$
- #define `SYSCON_PDRUNCFG_PD_SRAMX`  $(1 \ll 16)$
- #define `SYSCON_PDRUNCFG_PD_ROM`  $(1 \ll 17)$
- #define `SYSCON_PDRUNCFG_PD_VDDHV_ENA`  $(1 \ll 18)$
- #define `SYSCON_PDRUNCFG_PD_VDDA_ENA`  $(1 \ll 19)$
- #define `SYSCON_PDRUNCFG_PD_WDT_OSC`  $(1 \ll 20)$
- #define `SYSCON_PDRUNCFG_PD_USB_PHY`  $(1 \ll 21)$
- #define `SYSCON_PDRUNCFG_PD_SYS_PLL`  $(1 \ll 22)$
- #define `SYSCON_PDRUNCFG_PD_VREFP`  $(1 \ll 23)$
- #define `SYSCON_AUTOCGOR_RAM0X`  $(1 \ll 1)$
- #define `SYSCON_AUTOCGOR_RAM1`  $(1 \ll 2)$
- #define `SYSCON_AUTOCGOR_RAM2`  $(1 \ll 3)$
- #define `SYSCON_AUTOCGOR_MASK`  $(SYSCON_AUTOCGOR_RAM0X \mid SYSCON_AUTOCGOR_RAM1 \mid SYSCON_AUTOCGOR_RAM2)$

## Enumerations

- enum `CHIP_SYSCON_BOOT_MODE_REMAP_T` { `REMAP_BOOT_LOADER_MODE`, `REMAP_USER_RAM_MODE`, `REMAP_USER_FLASH_MODE` }
- enum `CHIP_SYSCON_PERIPH_RESET_T` {  
`RESET_FLASH` = 7, `RESET_FMC`, `RESET_SPIFI` = 10, `RESET_MUX`,  
`RESET_IOCON` = 13, `RESET_GPIO0`, `RESET_GPIO1`, `RESET_PINT` = 18,  
`RESET_GINT`, `RESET_DMA`, `RESET_CRC`, `RESET_WWDT`,  
`RESET_ADC` = 27, `RESET_ADC0` = 27, `RESET_MRT` = 32, `RESET_SCT0` = 32 + 2,  
`RESET_SCT` = 32 + 2, `RESET_UTICK` = 32 + 10, `RESET_FLEXCOMM0`, `RESET_FLEXCOMM1`,  
`RESET_FLEXCOMM2`, `RESET_FLEXCOMM3`, `RESET_FLEXCOMM4`, `RESET_FLEXCOMM5`,  
`RESET_FLEXCOMM6`, `RESET_FLEXCOMM7`, `RESET_DMIC`, `RESET_TIMER2` = 32 + 22,  
`RESET_USB`, `RESET_TIMER0`, `RESET_TIMER1`, `RESET_TIMER3` = 128 + 13,  
`RESET_TIMER4` }

- enum `SYSCON_FLASHTIM_T` {  
`SYSCON_FLASH_1CYCLE` = 0, `FLASHTIM_20MHZ_CPU` = `SYSCON_FLASH_1CYCLE`, `SYSCON_FLASH_2CYCLE`, `SYSCON_FLASH_3CYCLE`,  
`SYSCON_FLASH_4CYCLE`, `SYSCON_FLASH_5CYCLE`, `SYSCON_FLASH_6CYCLE`, `SYSCON_FLASH_7CYCLE`,  
`SYSCON_FLASH_8CYCLE` }  
*FLASH Access time definitions.*
- enum `CHIP_SYSCON_WAKEUP_T` {  
`SYSCON_STARTER_WWDT_BOD` = 0, `SYSCON_STARTER_DMA`, `SYSCON_STARTER_GINT0`, `SYSCON_STARTER_GINT1`,  
`SYSCON_STARTER_PINT0`, `SYSCON_STARTER_PINT1`, `SYSCON_STARTER_PINT2`, `SYSCON_STARTER_PINT3`,  
`SYSCON_STARTER_UTICK`, `SYSCON_STARTER_MRT`, `SYSCON_STARTER_TIMER0`, `SYSCON_STARTER_TIMER1`,  
`SYSCON_STARTER_SCT0`, `SYSCON_STARTER_TIMER3`, `SYSCON_STARTER_FLEXCOMM0`, `SYSCON_STARTER_FLEXCOMM1`,  
`SYSCON_STARTER_FLEXCOMM2`, `SYSCON_STARTER_FLEXCOMM3`, `SYSCON_STARTER_FLEXCOMM4`, `SYSCON_STARTER_FLEXCOMM5`,  
`SYSCON_STARTER_FLEXCOMM6`, `SYSCON_STARTER_FLEXCOMM7`, `SYSCON_STARTER_ADC0_SEQA`, `SYSCON_STARTER_ADC0_SEQB`,  
`SYSCON_STARTER_ADC0_THCMP`, `SYSCON_STARTER_DMIC`, `SYSCON_STARTER_HWVAD`, `SYSCON_STARTER_USBNEEDCLK`,  
`SYSCON_STARTER_USB`, `SYSCON_STARTER_RTC`, `SYSCON_STARTER_RESERVED0`, `SYSCON_STARTER_MAILBOX`,  
`SYSCON_STARTER_PINT4`, `SYSCON_STARTER_PINT5`, `SYSCON_STARTER_PINT6`, `SYSCON_STARTER_PINT7`,  
`SYSCON_STARTER_TIMER2`, `SYSCON_STARTER_TIMER4` }

## Functions

- `__STATIC_INLINE void Chip_SYSCON_Map (CHIP_SYSCON_BOOT_MODE_REMAP_T remap)`  
*Re-map interrupt vectors.*
- `__STATIC_INLINE CHIP_SYSCON_BOOT_MODE_REMAP_T Chip_SYSCON_GetMemoryMap (void)`  
*Get system remap setting.*
- `__STATIC_INLINE void Chip_SYSCON_SetSYSTCKCAL (uint32_t sysCalVal)`  
*Set System tick timer calibration value.*
- `void Chip_SYSCON_SetNMISource (uint32_t intsrc)`  
*Set source for non-maskable interrupt (NMI)*
- `void Chip_SYSCON_EnableNMISource (void)`  
*Enable interrupt used for NMI source.*
- `void Chip_SYSCON_DisableNMISource (void)`  
*Disable interrupt used for NMI source.*
- `void Chip_SYSCON_Enable_ASYNC_Syscon (bool enable)`  
*Enable or disable asynchronous APB bridge and subsystem.*
- `__STATIC_INLINE void Chip_SYSCON_SetUSARTFRGCtrl (uint8_t fmul, uint8_t fdiv)`  
*Set UART Fractional divider value.*
- `__STATIC_INLINE uint32_t Chip_SYSCON_GetSystemRSTStatus (void)`  
*Get system reset status.*
- `__STATIC_INLINE void Chip_SYSCON_ClearSystemRSTStatus (uint32_t reset)`  
*Clear system reset status.*
- `__STATIC_INLINE void Chip_SYSCON_PeriphReset (CHIP_SYSCON_PERIPH_RESET_T periph)`  
*Resets a peripheral.*
- `__STATIC_INLINE uint32_t Chip_SYSCON_GetPORPIOStatus (uint8_t port)`  
*Read POR captured PIO status.*

- `__STATIC_INLINE uint32_t Chip_SYSCON_GetResetPIOStatus` (uint8\_t port)  
*Read reset captured PIO status.*
- `__STATIC_INLINE void Chip_SYSCON_StartFreqMeas` (void)  
*Starts a frequency measurement cycle.*
- `__STATIC_INLINE bool Chip_SYSCON_IsFreqMeasComplete` (void)  
*Indicates when a frequency measurement cycle is complete.*
- `__STATIC_INLINE uint32_t Chip_SYSCON_GetRawFreqMeasCapval` (void)  
*Returns the raw capture value for a frequency measurement cycle.*
- `uint32_t Chip_SYSCON_GetCompFreqMeas` (uint32\_t refClockRate)  
*Returns the computed value for a frequency measurement cycle.*
- `__STATIC_INLINE void Chip_SYSCON_SetFLASHAccess` (SYSCON\_FLASHTIM\_T clks)  
*Set FLASH memory access time in clocks.*
- `__STATIC_INLINE uint32_t Chip_SYSCON_GetPowerStates` (void)  
*Power up one or more blocks or peripherals.*
- `__STATIC_INLINE void Chip_SYSCON_PowerDown` (uint32\_t powerdownmask)  
*Power down one or more blocks or peripherals.*
- `void Chip_SYSCON_PowerUp` (uint32\_t powerupmask)  
*Power up one or more blocks or peripherals.*
- `__STATIC_INLINE void Chip_SYSCON_EnableWakeup` (CHIP\_SYSCON\_WAKEUP\_T periphId)  
*Enables a pin's (PINT) wakeup logic.*
- `__STATIC_INLINE void Chip_SYSCON_DisableWakeup` (CHIP\_SYSCON\_WAKEUP\_T periphId)  
*Disables peripheral's wakeup logic.*
- `__STATIC_INLINE uint32_t Chip_SYSCON_GetDeviceID` (void)  
*Return the pointer to device ID registers.*
- `__STATIC_INLINE void Chip_SYSCON_DisableAutoClocking` (uint32\_t mask)  
*Disables Auto clock gating for SRAM's.*
- `__STATIC_INLINE void Chip_SYSCON_EnableAutoClocking` (uint32\_t mask)  
*Re-enables Auto clock gating for SRAM's.*

## 8.50 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/timer\_5411x.h File Reference

### Data Structures

- struct `LPC_TIMER_T`  
*32-bit Standard timer register block structure*

### Macros

- `#define TIMER_IR_CLR(n) _BIT(n)`
- `#define TIMER_MATCH_INT(n) (_BIT(((n) & 0x0F))`
- `#define TIMER_CAP_INT(n) (_BIT((((n) & 0x0F) + 4)))`
- `#define TIMER_ENABLE ((uint32_t) (1 << 0))`
- `#define TIMER_RESET ((uint32_t) (1 << 1))`
- `#define TIMER_CTRL_MASK ((uint32_t) 0x03)`
- `#define TIMER_INT_ON_MATCH(n) (_BIT(((n) * 3)))`
- `#define TIMER_RESET_ON_MATCH(n) (_BIT((((n) * 3) + 1)))`
- `#define TIMER_STOP_ON_MATCH(n) (_BIT((((n) * 3) + 2)))`
- `#define TIMER_MCR_MASK ((uint32_t) 0x0FFF)`
- `#define TIMER_CAP_RISING(n) (_BIT(((n) * 3)))`
- `#define TIMER_CAP_FALLING(n) (_BIT((((n) * 3) + 1)))`

- #define `TIMER_INT_ON_CAP`(n) (`_BIT`((n) \* 3) + 2))
- #define `TIMER_CCR_MASK` ((uint32\_t) 0x0FFF)
- #define `TIMER_EMR_MASK` ((uint32\_t) 0x0FFF)
- #define `TIMER_CTCR_MASK` ((uint32\_t) 0x0F)

## Enumerations

- enum `TIMER_PIN_MATCH_STATE_T` { `TIMER_EXTMATCH_DO_NOTHING` = 0, `TIMER_EXTMATCH_CLEAR` = 1, `TIMER_EXTMATCH_SET` = 2, `TIMER_EXTMATCH_TOGGLE` = 3 }
- *Standard timer initial match pin state and change state.*
- enum `TIMER_CAP_SRC_STATE_T` { `TIMER_CAPSRC_RISING_PCLK` = 0, `TIMER_CAPSRC_RISING_CAPN` = 1, `TIMER_CAPSRC_FALLING_CAPN` = 2, `TIMER_CAPSRC_BOTH_CAPN` = 3 }
- *Standard timer clock and edge for count source.*

## Functions

- `__STATIC_INLINE void Chip_TIMER_Init (LPC_TIMER_T *pTMR)`  
*Initialize a timer.*
- `__STATIC_INLINE void Chip_TIMER_DeInit (LPC_TIMER_T *pTMR)`  
*Shutdown a timer.*
- `__STATIC_INLINE bool Chip_TIMER_MatchPending (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Determine if a match interrupt is pending.*
- `__STATIC_INLINE bool Chip_TIMER_CapturePending (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Determine if a capture interrupt is pending.*
- `__STATIC_INLINE void Chip_TIMER_ClearMatch (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Clears a (pending) match interrupt.*
- `__STATIC_INLINE void Chip_TIMER_ClearCapture (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Clears a (pending) capture interrupt.*
- `__STATIC_INLINE void Chip_TIMER_Enable (LPC_TIMER_T *pTMR)`  
*Enables the timer (starts count)*
- `__STATIC_INLINE void Chip_TIMER_Disable (LPC_TIMER_T *pTMR)`  
*Disables the timer (stops count)*
- `__STATIC_INLINE uint32_t Chip_TIMER_ReadCount (LPC_TIMER_T *pTMR)`  
*Returns the current timer count.*
- `__STATIC_INLINE uint32_t Chip_TIMER_ReadPrescale (LPC_TIMER_T *pTMR)`  
*Returns the current prescale count.*
- `__STATIC_INLINE void Chip_TIMER_PrescaleSet (LPC_TIMER_T *pTMR, uint32_t prescale)`  
*Sets the prescaler value.*
- `__STATIC_INLINE void Chip_TIMER_SetMatch (LPC_TIMER_T *pTMR, int8_t matchnum, uint32_t matchval)`  
*Sets a timer match value.*
- `__STATIC_INLINE uint32_t Chip_TIMER_ReadCapture (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Reads a capture register.*
- `void Chip_TIMER_Reset (LPC_TIMER_T *pTMR)`  
*Resets the timer terminal and prescale counts to 0.*
- `__STATIC_INLINE void Chip_TIMER_MatchEnableInt (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Enables a match interrupt that fires when the terminal count matches the match counter value.*
- `__STATIC_INLINE void Chip_TIMER_MatchDisableInt (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Disables a match interrupt for a match counter.*
- `__STATIC_INLINE void Chip_TIMER_ResetOnMatchEnable (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*For the specific match counter, enables reset of the terminal count register when a match occurs.*

- `__STATIC_INLINE void Chip_TIMER_ResetOnMatchDisable (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*For the specific match counter, disables reset of the terminal count register when a match occurs.*
- `__STATIC_INLINE void Chip_TIMER_StopOnMatchEnable (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Enable a match timer to stop the terminal count when a match count equals the terminal count.*
- `__STATIC_INLINE void Chip_TIMER_StopOnMatchDisable (LPC_TIMER_T *pTMR, int8_t matchnum)`  
*Disable stop on match for a match timer. Disables a match timer to stop the terminal count when a match count equals the terminal count.*
- `__STATIC_INLINE void Chip_TIMER_CaptureRisingEdgeEnable (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Enables capture on on rising edge of selected CAP signal for the selected capture register, enables the selected CAPn.capnum signal to load the capture register with the terminal coount on a rising edge.*
- `__STATIC_INLINE void Chip_TIMER_CaptureRisingEdgeDisable (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Disables capture on on rising edge of selected CAP signal. For the selected capture register, disables the selected CAPn.capnum signal to load the capture register with the terminal coount on a rising edge.*
- `__STATIC_INLINE void Chip_TIMER_CaptureFallingEdgeEnable (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Enables capture on on falling edge of selected CAP signal. For the selected capture register, enables the selected CAPn.capnum signal to load the capture register with the terminal coount on a falling edge.*
- `__STATIC_INLINE void Chip_TIMER_CaptureFallingEdgeDisable (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Disables capture on on falling edge of selected CAP signal. For the selected capture register, disables the selected CAPn.capnum signal to load the capture register with the terminal coount on a falling edge.*
- `__STATIC_INLINE void Chip_TIMER_CaptureEnableInt (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Enables interrupt on capture of selected CAP signal. For the selected capture register, an interrupt will be generated when the enabled rising or falling edge on CAPn.capnum is detected.*
- `__STATIC_INLINE void Chip_TIMER_CaptureDisableInt (LPC_TIMER_T *pTMR, int8_t capnum)`  
*Disables interrupt on capture of selected CAP signal.*
- `void Chip_TIMER_ExtMatchControlSet (LPC_TIMER_T *pTMR, int8_t initial_state, TIMER_PIN_MATCH↵  
STATE_T matchState, int8_t matchnum)`  
*Sets external match control (MATn.matchnum) pin control. For the pin selected with matchnum, sets the function of the pin that occurs on a terminal count match for the match count.*
- `__STATIC_INLINE void Chip_TIMER_TIMER_SetCountClockSrc (LPC_TIMER_T *pTMR, TIMER_CAP_↵  
SRC_STATE_T capSrc, int8_t capnum)`  
*Sets timer count source and edge with the selected passed from CapSrc. If CapSrc selected a CAPn pin, select the specific CAPn pin with the capnum value.*

## 8.51 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/uart\_5411x.h File Reference

```
#include "ring_buffer.h"
```

### Data Structures

- struct `LPC_USART_T`  
*USART Registers.*
- struct `UART_BAUD_T`  
*UART Baud rate calculation structure.*
- struct `UART_STATISTICS_T`  
*UART statistics structure.*

## Macros

- #define [ECHO\\_EN](#) 1
- #define [ECHO\\_DIS](#) 0
- #define [UART\\_CFG\\_BITMASK](#) (0x00fddbfd)
- UART CFG register definitions.*
- #define [UART\\_CFG\\_ENABLE](#) (0x01 << 0)
- #define [UART\\_CFG\\_DATALEN\\_7](#) (0x00 << 2)
- #define [UART\\_CFG\\_DATALEN\\_8](#) (0x01 << 2)
- #define [UART\\_CFG\\_DATALEN\\_9](#) (0x02 << 2)
- #define [UART\\_CFG\\_PARITY\\_NONE](#) (0x00 << 4)
- #define [UART\\_CFG\\_PARITY\\_EVEN](#) (0x02 << 4)
- #define [UART\\_CFG\\_PARITY\\_ODD](#) (0x03 << 4)
- #define [UART\\_CFG\\_STOPLEN\\_1](#) (0x00 << 6)
- #define [UART\\_CFG\\_STOPLEN\\_2](#) (0x01 << 6)
- #define [UART\\_CFG\\_MODE32K](#) (0x01 << 7)
- #define [UART\\_CFG\\_LINMODE](#) (0x01 << 8)
- #define [UART\\_CFG\\_CTSEN](#) (0x01 << 9)
- #define [UART\\_CFG\\_SYNCEN](#) (0x01 << 11)
- #define [UART\\_CFG\\_CLKPOL](#) (0x01 << 12)
- #define [UART\\_CFG\\_SYNCMST](#) (0x01 << 14)
- #define [UART\\_CFG\\_LOOP](#) (0x01 << 15)
- #define [UART\\_CFG\\_IOMODE](#) (0x01 << 16)
- #define [UART\\_CFG\\_OETA](#) (0x01 << 18)
- #define [UART\\_CFG\\_AUTOADDR](#) (0x01 << 19)
- #define [UART\\_CFG\\_OESEL](#) (0x01 << 20)
- #define [UART\\_CFG\\_OEPOL](#) (0x01 << 21)
- #define [UART\\_CFG\\_RXPOL](#) (0x01 << 22)
- #define [UART\\_CFG\\_TXPOL](#) (0x01 << 23)
- #define [UART\\_CTRL\\_TXBRKEN](#) (0x01 << 1)
- UART CTRL register definitions.*
- #define [UART\\_CTRL\\_ADDRDET](#) (0x01 << 2)
- #define [UART\\_CTRL\\_TXDIS](#) (0x01 << 6)
- #define [UART\\_CTRL\\_CC](#) (0x01 << 8)
- #define [UART\\_CTRL\\_CLRCCONRX](#) (0x01 << 9)
- #define [UART\\_CTRL\\_AUTOBAUD](#) (0x01 << 16)
- #define [UART\\_STAT\\_RXIDLE](#) (0x01 << 1)
- UART STAT register definitions.*
- #define [UART\\_STAT\\_TXIDLE](#) (0x01 << 3)
- #define [UART\\_STAT\\_CTS](#) (0x01 << 4)
- #define [UART\\_STAT\\_DELTACTS](#) (0x01 << 5)
- #define [UART\\_STAT\\_TXDISINT](#) (0x01 << 6)
- #define [UART\\_STAT\\_RXBRK](#) (0x01 << 10)
- #define [UART\\_STAT\\_DELTARXBRK](#) (0x01 << 11)
- #define [UART\\_STAT\\_START](#) (0x01 << 12)
- #define [UART\\_STAT\\_FRM\\_ERRINT](#) (0x01 << 13)
- #define [UART\\_STAT\\_PAR\\_ERRINT](#) (0x01 << 14)
- #define [UART\\_STAT\\_RXNOISEINT](#) (0x01 << 15)
- #define [UART\\_STAT\\_ABERR](#) (0x01 << 16)
- #define [UART\\_INT\\_TXIDLE](#) (0x01 << 3)
- UART INTENSET/INTENCLR/INTSTAT register definitions.*
- #define [UART\\_INT\\_DELTACTS](#) (0x01 << 5)
- #define [UART\\_INT\\_TXDIS](#) (0x01 << 6)
- #define [UART\\_INT\\_DELTARXBRK](#) (0x01 << 11)

- #define `UART_INT_START` (0x01 << 12)
- #define `UART_INT_FRAMERR` (0x01 << 13)
- #define `UART_INT_PARITYERR` (0x01 << 14)
- #define `UART_INT_RXNOISE` (0x01 << 15)
- #define `UART_INT_ABERR` (0x01 << 16)
- #define `UART_FIFOCFG_BITMASK` (0x7F033)

*UART FIFO Configuration register bits.*

- #define `UART_FIFOCFG_ENABLETX` (1 << 0)
- #define `UART_FIFOCFG_ENABLERX` (1 << 1)
- #define `UART_FIFOCFG_DMATX` (1 << 12)
- #define `UART_FIFOCFG_DMARX` (1 << 13)
- #define `UART_FIFOCFG_WAKETX` (1 << 14)
- #define `UART_FIFOCFG_WAKERX` (1 << 15)
- #define `UART_FIFOCFG_EMPTYTX` (1 << 16)
- #define `UART_FIFOCFG_EMPTYRX` (1 << 17)
- #define `UART_FIFO_DEPTH` (16) /\*\* UART-FIFO How many entries are in the FIFO \*/

*UART FIFO Status register defines.*

- #define `UART_FIFOSTAT_BITMASK` (0x1F1FFB) /\*\* UART-FIFO STAT Register BitMask \*/
- #define `UART_FIFOSTAT_TXERR` (1 << 0)
- #define `UART_FIFOSTAT_RXERR` (1 << 1)
- #define `UART_FIFOSTAT_PERIPH` (1 << 3)
- #define `UART_FIFOSTAT_TXEMPTY` (1 << 4)
- #define `UART_FIFOSTAT_TXNOTFULL` (1 << 5)
- #define `UART_FIFOSTAT_RXNOTEMPTY` (1 << 6)
- #define `UART_FIFOSTAT_RXFULL` (1 << 7)
- #define `UART_FIFOSTAT_TXLVL`(lvl) (((lvl) >> 8) & 0x1F)
- #define `UART_FIFOSTAT_RXLVL`(lvl) (((lvl) >> 16) & 0x1F)
- #define `UART_FIFOTRIG_BITMASK` (0x000f0f03) /\*\* UART FIFO trigger settings Register BitMask \*/

*UART FIFO trigger settings register defines.*

- #define `UART_FIFOTRIG_TXLVLENA` (1 << 0)
- #define `UART_FIFOTRIG_RXLVLENA` (1 << 1)
- #define `UART_FIFOTRIG_TXLVL`(lvl) ((lvl & 0x0f) << 8)
- #define `UART_FIFOTRIG_RXLVL`(lvl) ((lvl & 0x0f) << 16)
- #define `UART_FIFOINT_BITMASK` (0x001F) /\*\* FIFO interrupt Bit mask \*/

*UART FIFO Interrupt enable/disable/status [INTSET/INTCLR/INTSTAT and FIFOINTSET/FIFOINTCLR/FIFOINTSTAT registers].*

- #define `UART_FIFOINT_TXERR` (1 << 0) /\*\* TX error interrupt [BIT-0 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/
- #define `UART_FIFOINT_RXERR` (1 << 1) /\*\* RX error interrupt [BIT-1 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/
- #define `UART_FIFOINT_TXLVL` (1 << 2) /\*\* TX FIFO ready interrupt [BIT-2 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/
- #define `UART_FIFOINT_RXLVL` (1 << 3) /\*\* RX Data ready interrupt [BIT-3 of FIFOINTENSET/FIFOINTENCLR/FIFOINTSTAT register] \*/
- #define `UART_FIFOINT_PERINT` (1 << 4) /\*\* UART peripheral interrupt [BIT-4 of FIFOINTSTAT register] \*/

## Functions

- `__STATIC_INLINE void Chip_UART_Enable (LPC_USART_T *pUART)`  
*Enable the UART.*
- `__STATIC_INLINE void Chip_UART_Disable (LPC_USART_T *pUART)`  
*Disable the UART.*
- `__STATIC_INLINE void Chip_UART_TXEnable (LPC_USART_T *pUART)`



- Enable transmission on UART TxD pin.*

  - `__STATIC_INLINE void Chip_UART_TXDisable (LPC_USART_T *pUART)`
- Disable transmission on UART TxD pin.*

  - `__STATIC_INLINE uint32_t Chip_UART_AutoBaud (LPC_USART_T *pUART)`
- Set auto baud.*

  - `__STATIC_INLINE void Chip_UART_SendByte (LPC_USART_T *pUART, uint8_t data)`
- Transmit a single data byte through the UART peripheral.*

  - `__STATIC_INLINE uint32_t Chip_UART_ReadByte (LPC_USART_T *pUART)`
- Read a single byte data from the UART peripheral.*

  - `__STATIC_INLINE void Chip_UART_IntEnable (LPC_USART_T *pUART, uint32_t intMask)`
- Enable UART interrupts.*

  - `__STATIC_INLINE void Chip_UART_IntDisable (LPC_USART_T *pUART, uint32_t intMask)`
- Disable UART interrupts.*

  - `__STATIC_INLINE uint32_t Chip_UART_GetIntsEnabled (LPC_USART_T *pUART)`
- Returns UART interrupts that are enabled.*

  - `__STATIC_INLINE uint32_t Chip_UART_GetIntStatus (LPC_USART_T *pUART)`
- Get UART interrupt status.*

  - `__STATIC_INLINE void Chip_UART_ConfigData (LPC_USART_T *pUART, uint32_t config)`
- Configure data width, parity and stop bits.*

  - `__STATIC_INLINE uint32_t Chip_UART_GetStatus (LPC_USART_T *pUART)`
- Get the UART status register.*

  - `__STATIC_INLINE void Chip_UART_ClearStatus (LPC_USART_T *pUART, uint32_t stsMask)`
- Clear the UART status register.*

  - `__STATIC_INLINE uint32_t Chip_UART_GetFIFOStatus (LPC_USART_T *pUART)`
- Get the current status of UART controller FIFO.*

  - `__STATIC_INLINE void Chip_UART_ClearFIFOStatus (LPC_USART_T *pUART, uint32_t mask)`
- Clear the FIFO status register.*

  - `__STATIC_INLINE void Chip_UART_SetFIFOTrigLevel (LPC_USART_T *pUART, uint8_t tx_lvl, uint8_t rx_lvl)`
- Setup the trigger level for UART FIFO.*

  - `__STATIC_INLINE void Chip_UART_EnableFIFOInts (LPC_USART_T *pUART, uint32_t intMask)`
- Enable a UART FIFO interrupt.*

  - `__STATIC_INLINE void Chip_UART_DisableFIFOInts (LPC_USART_T *pUART, uint32_t intMask)`
- Disable a UART FIFO interrupt.*

  - `__STATIC_INLINE uint32_t Chip_UART_GetFIFOEnabledInts (LPC_USART_T *pUART)`
- Return enabled UART FIFO interrupts.*

  - `__STATIC_INLINE uint32_t Chip_UART_GetFIFOPendingInts (LPC_USART_T *pUART)`
- Return pending UART FIFO interrupts.*

  - `__STATIC_INLINE void Chip_UART_SetFIFOCfg (LPC_USART_T *pUART, uint32_t cfg)`
- Set FIFO Configuration register.*

  - `__STATIC_INLINE void Chip_UART_ClearFIFOCfg (LPC_USART_T *pUART, uint32_t cfg)`
- Clear FIFO Configuration register.*

  - `__STATIC_INLINE void Chip_UART_FlushFIFOs (LPC_USART_T *pUART)`
- Flush FIFOs.*

  - `int Chip_UART_Init (LPC_USART_T *pUART)`
- Initialize the UART peripheral.*

  - `void Chip_UART_DeInit (LPC_USART_T *pUART)`
- Deinitialize the UART peripheral.*

  - `void Chip_UART_ConfigDMA (LPC_USART_T *pUART)`
- Configure UART for DMA.*

  - `int Chip_UART_Send (LPC_USART_T *pUART, const void *data, int numBytes)`



- Transmit a byte array through the UART peripheral (non-blocking)*
- int [Chip\\_UART\\_Read](#) (LPC\_USART\_T \*pUART, void \*data, int numBytes)
- Read data through the UART peripheral (non-blocking)*
- uint32\_t [Chip\\_UART\\_SetBaud](#) (LPC\_USART\_T \*pUART, uint32\_t baudrate)
- Set baud rate for UART.*
- int [Chip\\_UART\\_SendBlocking](#) (LPC\_USART\_T \*pUART, const void \*data, int numBytes)
- Transmit a byte array through the UART peripheral (blocking)*
- int [Chip\\_UART\\_ReadBlocking](#) (LPC\_USART\_T \*pUART, void \*data, int numBytes)
- Read data through the UART peripheral (blocking)*
- void [Chip\\_UART\\_RXIntHandlerRB](#) (LPC\_USART\_T \*pUART, RINGBUFF\_T \*pRB)
- UART receive-only interrupt handler for ring buffers.*
- void [Chip\\_UART\\_TXIntHandlerRB](#) (LPC\_USART\_T \*pUART, RINGBUFF\_T \*pRB)
- UART transmit-only interrupt handler for ring buffers.*
- uint32\_t [Chip\\_UART\\_SendRB](#) (LPC\_USART\_T \*pUART, RINGBUFF\_T \*pRB, const void \*data, int count)
- Populate a transmit ring buffer and start UART transmit.*
- int [Chip\\_UART\\_ReadRB](#) (LPC\_USART\_T \*pUART, RINGBUFF\_T \*pRB, void \*data, int bytes)
- Copy data from a receive ring buffer.*
- void [Chip\\_UART\\_IRQHandlerRB](#) (LPC\_USART\_T \*pUART, UART\_STATISTICS\_T \*statistics, RINGBUFF\_T \*pRXRB, RINGBUFF\_T \*pTXRB)
- UART receive/transmit interrupt handler for ring buffers.*
- void [Chip\\_UART\\_IRQHandlerDMA](#) (LPC\_USART\_T \*pUART, UART\_STATISTICS\_T \*statistics)
- UART receive/transmit interrupt handler for DMA.*

## 8.52 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/usbd.h File Reference

### 8.52.1 Detailed Description

Common definitions and declarations for the USB stack.

Common definitions and declarations for the USB stack. `#include "lpc_types.h"`

### Data Structures

- struct [WB\\_T](#)
- union [WORD\\_BYTE](#)
- struct [BM\\_T](#)
- union [REQUEST\\_TYPE](#)
- struct [USB\\_SETUP\\_PACKET](#)
- struct [USB\\_DEVICE\\_DESCRIPTOR](#)
- struct [USB\\_DEVICE\\_QUALIFIER\\_DESCRIPTOR](#)
- struct [USB\\_CONFIGURATION\\_DESCRIPTOR](#)
- struct [USB\\_IAD\\_DESCRIPTOR](#)
- struct [USB\\_INTERFACE\\_DESCRIPTOR](#)
- struct [USB\\_ENDPOINT\\_DESCRIPTOR](#)
- struct [USB\\_STRING\\_DESCRIPTOR](#)
- struct [USB\\_COMMON\\_DESCRIPTOR](#)
- struct [USB\\_OTHER\\_SPEED\\_CONFIGURATION](#)

## Macros

- `#define USB_CONFIG_POWER_MA(mA) ((mA)/2)`
  - `#define USB_ENDPOINT_0_HS_MAXP 64`
  - `#define USB_ENDPOINT_0_LS_MAXP 8`
  - `#define USB_ENDPOINT_BULK_HS_MAXP 512`
  - `#define WBVAL(x) ((x) & 0xFF),(((x) >> 8) & 0xFF)`
  - `#define B3VAL(x) ((x) & 0xFF),(((x) >> 8) & 0xFF),(((x) >> 16) & 0xFF)`
  - `#define USB_DEVICE_DESC_SIZE (sizeof(USB_DEVICE_DESCRIPTOR))`
  - `#define USB_CONFIGURATION_DESC_SIZE (sizeof(USB_CONFIGURATION_DESCRIPTOR))`
  - `#define USB_INTERFACE_DESC_SIZE (sizeof(USB_INTERFACE_DESCRIPTOR))`
  - `#define USB_INTERFACE_ASSOC_DESC_SIZE (sizeof(USB_IAD_DESCRIPTOR))`
  - `#define USB_ENDPOINT_DESC_SIZE (sizeof(USB_ENDPOINT_DESCRIPTOR))`
  - `#define USB_DEVICE_QUALI_SIZE (sizeof(USB_DEVICE_QUALIFIER_DESCRIPTOR))`
  - `#define USB_OTHER_SPEED_CONF_SIZE (sizeof(USB_OTHER_SPEED_CONFIGURATION))`
- 
- `#define REQUEST_HOST_TO_DEVICE 0`
  - `#define REQUEST_DEVICE_TO_HOST 1`
- 
- `#define REQUEST_STANDARD 0`
  - `#define REQUEST_CLASS 1`
  - `#define REQUEST_VENDOR 2`
  - `#define REQUEST_RESERVED 3`
- 
- `#define REQUEST_TO_DEVICE 0`
  - `#define REQUEST_TO_INTERFACE 1`
  - `#define REQUEST_TO_ENDPOINT 2`
  - `#define REQUEST_TO_OTHER 3`
- 
- `#define USB_REQUEST_GET_STATUS 0`
  - `#define USB_REQUEST_CLEAR_FEATURE 1`
  - `#define USB_REQUEST_SET_FEATURE 3`
  - `#define USB_REQUEST_SET_ADDRESS 5`
  - `#define USB_REQUEST_GET_DESCRIPTOR 6`
  - `#define USB_REQUEST_SET_DESCRIPTOR 7`
  - `#define USB_REQUEST_GET_CONFIGURATION 8`
  - `#define USB_REQUEST_SET_CONFIGURATION 9`
  - `#define USB_REQUEST_GET_INTERFACE 10`
  - `#define USB_REQUEST_SET_INTERFACE 11`
  - `#define USB_REQUEST_SYNC_FRAME 12`
- 
- `#define USB_GETSTATUS_SELF_POWERED 0x01`
  - `#define USB_GETSTATUS_REMOTE_WAKEUP 0x02`
  - `#define USB_GETSTATUS_ENDPOINT_STALL 0x01`
- 
- `#define USB_FEATURE_ENDPOINT_STALL 0`
  - `#define USB_FEATURE_REMOTE_WAKEUP 1`
  - `#define USB_FEATURE_TEST_MODE 2`
- 
- `#define USB_DEVICE_DESCRIPTOR_TYPE 1`
  - `#define USB_CONFIGURATION_DESCRIPTOR_TYPE 2`
  - `#define USB_STRING_DESCRIPTOR_TYPE 3`
  - `#define USB_INTERFACE_DESCRIPTOR_TYPE 4`
  - `#define USB_ENDPOINT_DESCRIPTOR_TYPE 5`

- #define [USB\\_DEVICE\\_QUALIFIER\\_DESCRIPTOR\\_TYPE](#) 6
  - #define [USB\\_OTHER\\_SPEED\\_CONFIG\\_DESCRIPTOR\\_TYPE](#) 7
  - #define [USB\\_INTERFACE\\_POWER\\_DESCRIPTOR\\_TYPE](#) 8
  - #define [USB\\_OTG\\_DESCRIPTOR\\_TYPE](#) 9
  - #define [USB\\_DEBUG\\_DESCRIPTOR\\_TYPE](#) 10
  - #define [USB\\_INTERFACE\\_ASSOCIATION\\_DESCRIPTOR\\_TYPE](#) 11
- 
- #define [USB\\_DEVICE\\_CLASS\\_RESERVED](#) 0x00
  - #define [USB\\_DEVICE\\_CLASS\\_AUDIO](#) 0x01
  - #define [USB\\_DEVICE\\_CLASS\\_COMMUNICATIONS](#) 0x02
  - #define [USB\\_DEVICE\\_CLASS\\_HUMAN\\_INTERFACE](#) 0x03
  - #define [USB\\_DEVICE\\_CLASS\\_MONITOR](#) 0x04
  - #define [USB\\_DEVICE\\_CLASS\\_PHYSICAL\\_INTERFACE](#) 0x05
  - #define [USB\\_DEVICE\\_CLASS\\_POWER](#) 0x06
  - #define [USB\\_DEVICE\\_CLASS\\_PRINTER](#) 0x07
  - #define [USB\\_DEVICE\\_CLASS\\_STORAGE](#) 0x08
  - #define [USB\\_DEVICE\\_CLASS\\_HUB](#) 0x09
  - #define [USB\\_DEVICE\\_CLASS\\_MISCELLANEOUS](#) 0xEF
  - #define [USB\\_DEVICE\\_CLASS\\_APP](#) 0xFE
  - #define [USB\\_DEVICE\\_CLASS\\_VENDOR\\_SPECIFIC](#) 0xFF
- 
- #define [USB\\_CONFIG\\_POWERED\\_MASK](#) 0x40
  - #define [USB\\_CONFIG\\_BUS\\_POWERED](#) 0x80
  - #define [USB\\_CONFIG\\_SELF\\_POWERED](#) 0xC0
  - #define [USB\\_CONFIG\\_REMOTE\\_WAKEUP](#) 0x20
- 
- #define [USB\\_ENDPOINT\\_DIRECTION\\_MASK](#) 0x80
  - #define [USB\\_ENDPOINT\\_OUT\(addr\)](#) ((addr) | 0x00)
  - #define [USB\\_ENDPOINT\\_IN\(addr\)](#) ((addr) | 0x80)
- 
- #define [USB\\_ENDPOINT\\_TYPE\\_MASK](#) 0x03
  - #define [USB\\_ENDPOINT\\_TYPE\\_CONTROL](#) 0x00
  - #define [USB\\_ENDPOINT\\_TYPE\\_ISOCHRONOUS](#) 0x01
  - #define [USB\\_ENDPOINT\\_TYPE\\_BULK](#) 0x02
  - #define [USB\\_ENDPOINT\\_TYPE\\_INTERRUPT](#) 0x03
  - #define [USB\\_ENDPOINT\\_SYNC\\_MASK](#) 0x0C
  - #define [USB\\_ENDPOINT\\_SYNC\\_NO\\_SYNCHRONIZATION](#) 0x00
  - #define [USB\\_ENDPOINT\\_SYNC\\_ASYNCHRONOUS](#) 0x04
  - #define [USB\\_ENDPOINT\\_SYNC\\_ADAPTIVE](#) 0x08
  - #define [USB\\_ENDPOINT\\_SYNC\\_SYNCHRONOUS](#) 0x0C
  - #define [USB\\_ENDPOINT\\_USAGE\\_MASK](#) 0x30
  - #define [USB\\_ENDPOINT\\_USAGE\\_DATA](#) 0x00
  - #define [USB\\_ENDPOINT\\_USAGE\\_FEEDBACK](#) 0x10
  - #define [USB\\_ENDPOINT\\_USAGE\\_IMPLICIT\\_FEEDBACK](#) 0x20
  - #define [USB\\_ENDPOINT\\_USAGE\\_RESERVED](#) 0x30

## Typedefs

- typedef void \* [USBD\\_HANDLE\\_T](#)

## 8.53 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/utick\_5411x.h File Reference

### Data Structures

- struct [LPC\\_UTICK\\_T](#)  
*Micro Tick register block structure.*

### Macros

- #define [UTICK\\_CTRL\\_REPEAT](#) ((uint32\_t) 1UL << 31)  
*UTick register definitions.*
- #define [UTICK\\_CTRL\\_DELAY\\_MASK](#) ((uint32\_t) 0x7FFFFFFF)
- #define [UTICK\\_STATUS\\_INTR](#) ((uint32\_t) 1 << 0)
- #define [UTICK\\_STATUS\\_ACTIVE](#) ((uint32\_t) 1 << 1)
- #define [UTICK\\_STATUS\\_MASK](#) ((uint32\_t) 0x03)

### Functions

- `__STATIC_INLINE void Chip\_UTICK\_Init (LPC_UTICK_T *pUTICK)`  
*Initialize the UTICK peripheral.*
- `__STATIC_INLINE void Chip\_UTICK\_DeInit (LPC_UTICK_T *pUTICK)`  
*De-initialize the UTICK peripheral.*
- `__STATIC_INLINE void Chip\_UTICK\_SetTick (LPC_UTICK_T *pUTICK, uint32_t tick_value, bool repeat)`  
*Setup UTICK.*
- `__STATIC_INLINE void Chip\_UTICK\_SetDelayMs (LPC_UTICK_T *pUTICK, uint32_t delayMs, bool repeat)`  
*Setup UTICK for the passed delay (in mS)*
- `__STATIC_INLINE uint32_t Chip\_UTICK\_GetTick (LPC_UTICK_T *pUTICK)`  
*Read UTICK Value.*
- `__STATIC_INLINE void Chip\_UTICK\_Halt (LPC_UTICK_T *pUTICK)`  
*Halt UTICK timer.*
- `__STATIC_INLINE uint32_t Chip\_UTICK\_GetStatus (LPC_UTICK_T *pUTICK)`  
*Returns the status of UTICK.*
- `__STATIC_INLINE void Chip\_UTICK\_ClearInterrupt (LPC_UTICK_T *pUTICK)`  
*Clears UTICK Interrupt flag.*

## 8.54 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/wwdt\_5411x.h File Reference

### Data Structures

- struct [LPC\\_WWDT\\_T](#)  
*Windowed Watchdog register block structure.*

### Macros

- #define [WWDT\\_WDMOD\\_BITMASK](#) ((uint32\_t) 0x3F)  
*Watchdog Mode register definitions.*
- #define [WWDT\\_WDMOD\\_WDEN](#) ((uint32\_t) (1 << 0))
- #define [WWDT\\_WDMOD\\_WDRESET](#) ((uint32\_t) (1 << 1))
- #define [WWDT\\_WDMOD\\_WDTOF](#) ((uint32\_t) (1 << 2))
- #define [WWDT\\_WDMOD\\_WDINT](#) ((uint32\_t) (1 << 3))
- #define [WWDT\\_WDMOD\\_WDPROTECT](#) ((uint32\_t) (1 << 4))
- #define [WWDT\\_WDMOD\\_LOCK](#) ((uint32\_t) (1 << 5))

## Functions

- `__STATIC_INLINE void Chip_WWDT_Init (LPC_WWDT_T *pWWDT)`  
*Initialize the Watchdog timer.*
- `__STATIC_INLINE void Chip_WWDT_DeInit (LPC_WWDT_T *pWWDT)`  
*Shutdown the Watchdog timer.*
- `__STATIC_INLINE void Chip_WWDT_SetTimeOut (LPC_WWDT_T *pWWDT, uint32_t timeout)`  
*Set WDT timeout constant value used for feed.*
- `__STATIC_INLINE void Chip_WWDT_Feed (LPC_WWDT_T *pWWDT)`  
*Feed watchdog timer.*
- `__STATIC_INLINE void Chip_WWDT_SetWarning (LPC_WWDT_T *pWWDT, uint32_t timeout)`  
*Set WWDT warning interrupt.*
- `__STATIC_INLINE uint32_t Chip_WWDT_GetWarning (LPC_WWDT_T *pWWDT)`  
*Get WWDT warning interrupt.*
- `__STATIC_INLINE void Chip_WWDT_SetWindow (LPC_WWDT_T *pWWDT, uint32_t timeout)`  
*Set WWDT window time.*
- `__STATIC_INLINE uint32_t Chip_WWDT_GetWindow (LPC_WWDT_T *pWWDT)`  
*Get WWDT window time.*
- `__STATIC_INLINE void Chip_WWDT_SetOption (LPC_WWDT_T *pWWDT, uint32_t options)`  
*Enable watchdog timer options.*
- `__STATIC_INLINE void Chip_WWDT_UnsetOption (LPC_WWDT_T *pWWDT, uint32_t options)`  
*Disable/clear watchdog timer options.*
- `__STATIC_INLINE void Chip_WWDT_Start (LPC_WWDT_T *pWWDT)`  
*Enable WWDT activity.*
- `__STATIC_INLINE uint32_t Chip_WWDT_GetStatus (LPC_WWDT_T *pWWDT)`  
*Read WWDT status flag.*
- `__STATIC_INLINE void Chip_WWDT_ClearStatusFlag (LPC_WWDT_T *pWWDT, uint32_t status)`  
*Clear WWDT interrupt status flags.*
- `__STATIC_INLINE uint32_t Chip_WWDT_GetCurrentCount (LPC_WWDT_T *pWWDT)`  
*Get the current value of WDT.*

## 8.55 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/adc\_5411x.c File Reference

```
#include "chip.h"
```

## Functions

- `void Chip_ADC_Init (LPC_ADC_T *pADC, uint32_t flags)`  
*Initialize the ADC peripheral.*
- `void Chip_ADC_DeInit (LPC_ADC_T *pADC)`  
*Shutdown ADC.*
- `uint32_t Chip_ADC_Calibration (LPC_ADC_T *pADC)`  
*Perform ADC calibration.*
- `void Chip_ADC_SetClockRate (LPC_ADC_T *pADC, uint32_t rate)`  
*Set ADC clock rate.*

## 8.56 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/chip\_5411x.c File Reference

```
#include "chip.h"
```

### Functions

- void [SystemCoreClockUpdate](#) (void)  
*Update system core and ASYNC syscon clock rate, should be called if the system has a clock rate change.*
- void [Chip\\_USB\\_Init](#) (void)  
*Initialize the USB bus.*

### Variables

- uint32\_t [SystemCoreClock](#)  
*Current system clock rate, mainly used for peripherals in SYSCON.*

## 8.57 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/clock\_5411x.c File Reference

```
#include "chip.h"
```

### Macros

- #define [WDT\\_FREQ\\_LOOKUP](#)

### Functions

- static uint32\_t [Chip\\_Clock\\_GetAsyncSyscon\\_ClockRate\\_NoDiv](#) (void)
- uint32\_t [Chip\\_Clock\\_GetMain\\_A\\_ClockRate](#) (void)  
*Return main A clock rate.*
- uint32\_t [Chip\\_Clock\\_GetMain\\_B\\_ClockRate](#) (void)  
*Return main B clock rate.*
- void [Chip\\_Clock\\_EnablePeriphClock](#) (CHIP\_SYSCON\_CLOCK\_T clk)  
*Enable a system or peripheral clock.*
- void [Chip\\_Clock\\_DisablePeriphClock](#) (CHIP\_SYSCON\_CLOCK\_T clk)  
*Disable a system or peripheral clock.*
- uint32\_t [Chip\\_Clock\\_GetSysTickClockRate](#) (void)  
*Returns the system tick rate as used with the system tick divider.*
- uint32\_t [Chip\\_Clock\\_GetADCClockRate](#) (void)  
*Return ADC clock rate.*
- uint32\_t [Chip\\_Clock\\_GetAsyncSyscon\\_ClockRate](#) (void)  
*Return asynchronous APB clock rate.*
- [CHIP\\_SYSCON\\_MAINCLKSRC\\_T](#) [Chip\\_Clock\\_GetMainClockSource](#) (void)  
*Get main system clock source.*
- uint32\_t [Chip\\_Clock\\_GetMainClockRate](#) (void)  
*Return main clock rate.*
- uint32\_t [Chip\\_Clock\\_GetSystemClockRate](#) (void)  
*Return system clock rate.*

- uint32\_t [Chip\\_Clock\\_GetFRGClockRate](#) (void)  
*Get Fraction Rate Generator (FRG) clock rate.*
- uint32\_t [Chip\\_Clock\\_GetFRGInClockRate](#) (void)  
*Get the input clock frequency of FRG.*
- uint32\_t [Chip\\_Clock\\_SetFRGClockRate](#) (uint32\_t rate)  
*Set FRG rate to given rate.*
- uint32\_t [Chip\\_Clock\\_GetMCLKClockRate](#) (void)
- uint32\_t [Chip\\_Clock\\_GetFLEXCOMMClockRate](#) (uint32\_t id)  
*Return FlexCOMM clock rate.*
- uint32\_t [Chip\\_Clock\\_GetWDTOSCRate](#) (void)  
*Return estimated watchdog oscillator rate.*

## Variables

- **WEAK** uint32\_t [mclk\\_in\\_rate](#) = 0

## 8.57.1 Macro Definition Documentation

### 8.57.1.1 #define WDT\_FREQ\_LOOKUP

#### Value:

```
"\x0\x0F\x1E\x28\x3C\x46\x50\x5A\x64\x6E"\  
"\x78\x82\x8C\x91\x96\xA0\xA5\xAA\xB4\xB9"\  
"\xBE\xC8\xCD\xD2\xDC\xE1\xE6\xEB\xF0\xF5"
```

Definition at line 39 of file clock\_5411x.c.

## 8.57.2 Function Documentation

### 8.57.2.1 static uint32\_t Chip\_Clock\_GetAsyncSyscon\_ClockRate\_NoDiv ( void ) [static]

Definition at line 54 of file clock\_5411x.c.

### 8.57.2.2 uint32\_t Chip\_Clock\_GetMCLKClockRate ( void )

< HF-FRO 48MHz or 96MHz

< Main pll

< MCLK INPUT Clock pin set by IOCON

< Disable clock source to MCLK

Definition at line 350 of file clock\_5411x.c.

## 8.57.3 Variable Documentation

### 8.57.3.1 **WEAK** uint32\_t mclk\_in\_rate = 0

Definition at line 47 of file clock\_5411x.c.

## 8.58 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/dma\_5411x.c File Reference

```
#include "chip.h"
```

## 8.59 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/dma\_service\_5411x.c File Reference

```
#include "chip.h"
```

### Macros

- `#define DMA_XFERCFG_WIDTH 8`
- `#define DMA_XFERCFG_SRCINC 12`
- `#define DMA_XFERCFG_DSTINC 14`

### Functions

- void `Chip_DMASERVICE_Init` (`DMA_CHDESC_T *base`)  
*Initialize DMA service.*
- void `Chip_DMASERVICE_ErrorHandler` (void)
- void `Chip_DMASERVICE_Isr` (void)  
*DMA service interrupt handler.*
- void `Chip_DMASERVICE_RegisterCb` (const `DMA_PERIPHERAL_CONTEXT_T *pContext`, `DMA_CALLBACK_T pCallback`)  
*Register callback function.*
- void `Chip_DMASERVICE_SingleBuffer` (const `DMA_PERIPHERAL_CONTEXT_T *pContext`, `uint32_t pMem`, `uint32_t length`)  
*Use Single buffer mechanism.*
- void `Chip_DMASERVICE_DoubleBuffer` (const `DMA_PERIPHERAL_CONTEXT_T *pContext`, `uint32_t pMem`, `uint32_t length`, `DMA_DUAL_DESCRIPTOR_T *pD`)  
*Use double buffer mechanism.*

### Variables

- `DMA_CALLBACK_T dma_pCallback_array` [`MAX_DMA_CHANNEL`]
- `DMA_CALLBACK_T dma_service_error_cb`

### 8.59.1 Macro Definition Documentation

#### 8.59.1.1 `#define DMA_XFERCFG_DSTINC 14`

Definition at line 97 of file `dma_service_5411x.c`.

#### 8.59.1.2 `#define DMA_XFERCFG_SRCINC 12`

Definition at line 96 of file `dma_service_5411x.c`.



8.59.1.3 `#define DMA_XFERCFG_WIDTH 8`

Definition at line 95 of file dma\_service\_5411x.c.

## 8.59.2 Function Documentation

8.59.2.1 `void Chip_DMASERVICE_ErrorHandler ( void )`

Definition at line 50 of file dma\_service\_5411x.c.

## 8.59.3 Variable Documentation

8.59.3.1 `DMA_CALLBACK_T dma_pCallback_array[MAX_DMA_CHANNEL]`

Definition at line 34 of file dma\_service\_5411x.c.

8.59.3.2 `DMA_CALLBACK_T dma_service_error_cb`

Definition at line 35 of file dma\_service\_5411x.c.

## 8.60 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/dmic\_5411x.c File Reference

```
#include "chip.h"
```

## Functions

- void `Chip_DMIC_Init` (const `CHIP_SYSCON_CLOCK_T` clock, const `CHIP_SYSCON_PERIPH_RESET_T` reset)  
*Initialize DMIC interface.*
- void `Chip_DMIC_CfgIO` (`LPC_DMIC_T` \*pDMIC, `DMIC_IO_T` cfg)  
*Configure DMIC io.*
- void `Chip_DMIC_SetOpMode` (`LPC_DMIC_T` \*pDMIC, `OP_MODE_T` mode)  
*Set DMIC operating mode.*
- void `Chip_DMIC_CfgChannel` (`LPC_DMIC_T` \*pDMIC, uint32\_t channel, `DMIC_CHANNEL_CONFIG_T` \*channel\_cfg)  
*Configure DMIC channel.*
- void `Chip_DMIC_CfgChannelDc` (`LPC_DMIC_T` \*pDMIC, uint32\_t channel, `DC_REMOVAL_T` dc\_cut\_level, uint32\_t post\_dc\_gain\_reduce, bool saturate16bit)  
*Configure DMIC channel DC removal setting.*
- void `Chip_DMIC_Use2fs` (`LPC_DMIC_T` \*pDMIC, bool use2fs)  
*Configure Clock scaling.*
- void `Chip_DMIC_EnableChannel` (`LPC_DMIC_T` \*pDMIC, uint32\_t channelmask)  
*Configure Clock scaling.*
- void `Chip_DMIC_FifoChannel` (`LPC_DMIC_T` \*pDMIC, uint32\_t channel, uint32\_t trig\_level, uint32\_t enable, uint32\_t resetn)  
*Configure fifo settings for DMIC channel.*

## Variables

- [DMA\\_PERIPHERAL\\_CONTEXT\\_T dmic\\_ch0\\_dma\\_context](#)
- [DMA\\_PERIPHERAL\\_CONTEXT\\_T dmic\\_ch1\\_dma\\_context](#)
- [DMA\\_PERIPHERAL\\_CONTEXT\\_T dmic\\_ch0\\_dma\\_interleaved\\_context](#)
- [DMA\\_PERIPHERAL\\_CONTEXT\\_T dmic\\_ch1\\_dma\\_interleaved\\_context](#)

## 8.61 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/flexcomm\_5411x.c File Reference

```
#include "chip.h"
```

## Functions

- int [Chip\\_FLEXCOMM\\_GetIndex](#) (LPC\_FLEXCOMM\_T \*pFCOMM)  
*Get index of the FLEXCOMM corresponding to the given base address.*
- int [Chip\\_FLEXCOMM\\_Init](#) (LPC\_FLEXCOMM\_T \*pFCOMM, FLEXCOMM\_PERIPH\_T periph)  
*Initialize FlexCOMM and associate it with a given peripheral.*
- void [Chip\\_FLEXCOMM\\_DeInit](#) (LPC\_FLEXCOMM\_T \*pFCOMM)  
*Uninitialize the FlexCOMM.*
- int [Chip\\_FLEXCOMM\\_SetPeriph](#) (LPC\_FLEXCOMM\_T \*pFCOMM, FLEXCOMM\_PERIPH\_T periph, int lock)  
*Set FLEXCOMM to a peripheral function.*

## 8.62 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/fpu\_init.c File Reference

## 8.63 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/i2cm\_5411x.c File Reference

```
#include "chip.h"
```

## Functions

- void [Chip\\_I2CM\\_SetBusSpeed](#) (LPC\_I2C\_T \*pI2C, uint32\_t busSpeed)  
*Set up bus speed for LPC\_I2C controller.*
- uint32\_t [Chip\\_I2CM\\_XferHandler](#) (LPC\_I2C\_T \*pI2C, I2CM\_XFER\_T \*xfer)  
*Transfer state change handler.*
- void [Chip\\_I2CM\\_Xfer](#) (LPC\_I2C\_T \*pI2C, I2CM\_XFER\_T \*xfer)  
*Transmit and Receive data in master mode.*
- uint32\_t [Chip\\_I2CM\\_XferBlocking](#) (LPC\_I2C\_T \*pI2C, I2CM\_XFER\_T \*xfer)  
*Transmit and Receive data in master mode.*

## 8.64 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/i2cs\_5411x.c File Reference

```
#include "chip.h"
```

## Functions

- uint32\_t [Chip\\_I2CS\\_XferHandler](#) (LPC\_I2C\_T \*pl2C, const I2CS\_XFER\_T \*xfers)  
*Slave transfer state change handler.*

## 8.65 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/i2s\_5411x.c File Reference

```
#include "chip.h"
```

## Functions

- int [Chip\\_I2S\\_Init](#) (LPC\_I2S\_T \*pl2S, I2S\_AUDIO\_FORMAT\_T \*fmt)  
*Initialize I2S driver.*
- [Status Chip\\_I2S\\_Config](#) (LPC\_I2S\_T \*pl2S, I2S\_AUDIO\_FORMAT\_T \*fmt)  
*Configure I2S Port.*
- void [Chip\\_I2S\\_FIFO\\_Config](#) (LPC\_I2S\_T \*pl2S, I2S\_AUDIO\_FORMAT\_T \*fmt)  
*Configure I2S FIFO.*
- static void [fifo\\_ctrl\\_tx](#) (LPC\_I2S\_T \*pl2S, I2S\_FIFO\_CMD\_T cmd)
- static void [fifo\\_ctrl\\_rx](#) (LPC\_I2S\_T \*pl2S, I2S\_FIFO\_CMD\_T cmd)
- void [Chip\\_I2S\\_FIFO\\_Control](#) (LPC\_I2S\_T \*pl2S, I2S\_AUDIO\_FORMAT\_T \*fmt, I2S\_FIFO\_CMD\_T cmd)  
*Execute I2S FIFO control commands.*
- void [Chip\\_I2S\\_ErrorHandler](#) (LPC\_I2S\_T \*pl2S, I2S\_STATISTICS\_T \*stat)  
*I2S error handler.*

### 8.65.1 Function Documentation

#### 8.65.1.1 Status Chip\_I2S\_Config ( LPC\_I2S\_T \* pl2S, I2S\_AUDIO\_FORMAT\_T \* fmt )

Configure I2S Port.

##### Parameters

|             |                                       |
|-------------|---------------------------------------|
| <i>pl2S</i> | : The base I2S peripheral on the chip |
| <i>fmt</i>  | : Audio Format                        |

##### Returns

SUCCESS or ERROR

Definition at line 67 of file i2s\_5411x.c.

#### 8.65.1.2 void Chip\_I2S\_ErrorHandler ( LPC\_I2S\_T \* pl2S, I2S\_STATISTICS\_T \* stat )

I2S error handler.

##### Parameters

|            |                                              |
|------------|----------------------------------------------|
| <i>I2S</i> | : The base of the I2S peripheral on the chip |
|------------|----------------------------------------------|

|             |                        |
|-------------|------------------------|
| <i>stat</i> | : Statistics structure |
|-------------|------------------------|

**Returns**

Nothing

Definition at line 210 of file i2s\_5411x.c.

**8.65.1.3** void Chip\_I2S\_FIFO\_Config ( LPC\_I2S\_T \* *pl2S*, I2S\_AUDIO\_FORMAT\_T \* *fmt* )

Configure I2S FIFO.

**Parameters**

|             |                                       |
|-------------|---------------------------------------|
| <i>pl2S</i> | : The base I2S peripheral on the chip |
| <i>fmt</i>  | : Audio format information            |

**Returns**

Nothing

Definition at line 120 of file i2s\_5411x.c.

**8.65.1.4** void Chip\_I2S\_FIFO\_Control ( LPC\_I2S\_T \* *pl2S*, I2S\_AUDIO\_FORMAT\_T \* *fmt*, I2S\_FIFO\_CMD\_T *cmd* )

Execute I2S FIFO control commands.

**Parameters**

|             |                                       |
|-------------|---------------------------------------|
| <i>pl2S</i> | : The base I2S peripheral on the chip |
| <i>fmt</i>  | : Audio format information            |
| <i>cmd</i>  | : FIFO command                        |

**Returns**

Nothing

Definition at line 191 of file i2s\_5411x.c.

**8.65.1.5** int Chip\_I2S\_Init ( LPC\_I2S\_T \* *pl2S*, I2S\_AUDIO\_FORMAT\_T \* *fmt* )

Initialize I2S driver.

**Parameters**

|             |                                          |
|-------------|------------------------------------------|
| <i>pl2S</i> | : The base of I2S peripheral on the chip |
|-------------|------------------------------------------|

**Returns**

0 - on success; [ERR\\_FLEXCOMM\\_FUNCNOTSUPPORTED](#) or [ERR\\_FLEXCOMM\\_NOTFREE](#) on failure

Definition at line 49 of file i2s\_5411x.c.

**8.65.1.6** static void fifo\_ctrl\_rx ( LPC\_I2S\_T \* *pl2S*, I2S\_FIFO\_CMD\_T *cmd* ) [static]

Definition at line 167 of file i2s\_5411x.c.

8.65.1.7 `static void fifo_ctrl_tx ( LPC_I2S_T * pI2S, I2S_FIFO_CMD_T cmd ) [static]`

Definition at line 140 of file i2s\_5411x.c.

## 8.66 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/iap.c File Reference

```
#include "chip.h"
```

### Functions

- `uint8_t Chip_IAP_PreSectorForReadWrite (uint32_t strSector, uint32_t endSector)`  
*Prepare sector for write operation.*
- `uint8_t Chip_IAP_CopyRamToFlash (uint32_t dstAdd, uint32_t *srcAdd, uint32_t byteswrt)`  
*Copy RAM to flash.*
- `uint8_t Chip_IAP_EraseSector (uint32_t strSector, uint32_t endSector)`  
*Erase sector.*
- `uint8_t Chip_IAP_BlankCheckSector (uint32_t strSector, uint32_t endSector)`  
*Blank check a sector or multiples sector of on-chip flash memory.*
- `uint32_t Chip_IAP_ReadPID ()`  
*Read part identification number.*
- `uint8_t Chip_IAP_ReadBootCode ()`  
*Read boot code version number.*
- `uint8_t Chip_IAP_Compare (uint32_t dstAdd, uint32_t srcAdd, uint32_t bytescmp)`  
*Compare the memory contents at two locations.*
- `uint8_t Chip_IAP_ReinvokeISP ()`  
*IAP reinvokes ISP to invoke the bootloader in ISP mode.*
- `uint32_t Chip_IAP_ReadUID ()`  
*Read the unique ID.*
- `uint8_t Chip_IAP_ErasePage (uint32_t strPage, uint32_t endPage)`  
*Erase a page or multiple pages of on-chip flash memory.*

## 8.67 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/pll\_5411x.c File Reference

```
#include "chip.h"
```

### Macros

- `#define NVALMAX (0x100)`
- `#define PVALMAX (0x20)`
- `#define MVALMAX (0x8000)`
- `#define SYS_PLL_SEL_R(d) (((d) & 0xf) << 0)`
- `#define SYS_PLL_SEL_I(d) (((d) & 0x3f) << 4)`
- `#define SYS_PLL_SEL_P(d) (((d) & 0x1f) << 10)`
- `#define SYS_PLL_BYPASS (1 << 15)`
- `#define SYS_PLL_BYPASS_CODIV2 (1 << 16)`
- `#define SYS_PLL_UPLIMOFF (1 << 17)`
- `#define SYS_PLL_BANDSEL (1 << 18)`

- #define [SYS\\_PLL\\_DIRECTI](#) (1 << 19)
- #define [SYS\\_PLL\\_DIRECTO](#) (1 << 20)
- #define [PLL\\_SSCG0\\_MDEC\\_VAL\\_P](#) (0)
- #define [PLL\\_SSCG0\\_MDEC\\_VAL\\_M](#) (0x1FFFFUL << PLL\_SSCG0\_MDEC\_VAL\_P)
- #define [PLL\\_NDEC\\_VAL\\_P](#) (0)
- #define [PLL\\_NDEC\\_VAL\\_M](#) (0x3FFUL << PLL\_NDEC\_VAL\_P)
- #define [PLL\\_PDEC\\_VAL\\_P](#) (0)
- #define [PLL\\_PDEC\\_VAL\\_M](#) (0x3FFUL << PLL\_PDEC\_VAL\_P)
- #define [PLL\\_MIN\\_CCO\\_FREQ\\_MHZ](#) (75000000)
- #define [PLL\\_MAX\\_CCO\\_FREQ\\_MHZ](#) (150000000)
- #define [PLL\\_LOWER\\_IN\\_LIMIT](#) (4000)
- #define [PLL\\_MIN\\_IN\\_SSMODE](#) (2000000)
- #define [PLL\\_MAX\\_IN\\_SSMODE](#) (4000000)
- #define [PLL\\_SSCG\\_MF\\_FREQ\\_VALUE](#) 4
- #define [PLL\\_SSCG\\_MC\\_COMP\\_VALUE](#) 2
- #define [PLL\\_SSCG\\_MR\\_DEPTH\\_VALUE](#) 4
- #define [PLL\\_SSCG\\_DITHER\\_VALUE](#) 0
- #define [SYSCON\\_SYSPLLCTRL\\_SEL\\_R\\_P](#) 0
- #define [SYSCON\\_SYSPLLCTRL\\_SEL\\_R\\_M](#) (0xFUL << SYSCON\_SYSPLLCTRL\_SEL\_R\_P)
- #define [SYSCON\\_SYSPLLCTRL\\_SEL\\_I\\_P](#) 4
- #define [SYSCON\\_SYSPLLCTRL\\_SEL\\_I\\_M](#) (0x3FUL << SYSCON\_SYSPLLCTRL\_SEL\_I\_P)
- #define [SYSCON\\_SYSPLLCTRL\\_SEL\\_P\\_P](#) 10
- #define [SYSCON\\_SYSPLLCTRL\\_SEL\\_P\\_M](#) (0x1FUL << SYSCON\_SYSPLLCTRL\_SEL\_P\_P)
- #define [SYSCON\\_SYSPLLCTRL\\_BYPASS\\_P](#) 15
- #define [SYSCON\\_SYSPLLCTRL\\_BYPASS](#) (1UL << SYSCON\_SYSPLLCTRL\_BYPASS\_P)
- #define [SYSCON\\_SYSPLLCTRL\\_BYPASS\\_FBDIV2\\_P](#) 16
- #define [SYSCON\\_SYSPLLCTRL\\_BYPASS\\_FBDIV2](#) (1UL << SYSCON\_SYSPLLCTRL\_BYPASS\_FBDIV2\_P)
- #define [SYSCON\\_SYSPLLCTRL\\_UPLIMOFF\\_P](#) 17
- #define [SYSCON\\_SYSPLLCTRL\\_UPLIMOFF](#) (1UL << SYSCON\_SYSPLLCTRL\_UPLIMOFF\_P)
- #define [SYSCON\\_SYSPLLCTRL\\_BANDSEL\\_SSCGREG\\_N\\_P](#) 18
- #define [SYSCON\\_SYSPLLCTRL\\_BANDSEL\\_SSCGREG\\_N](#) (1UL << SYSCON\_SYSPLLCTRL\_BANDSEL\_SSCGREG\_N\_P)
- #define [SYSCON\\_SYSPLLCTRL\\_DIRECTI\\_P](#) 19
- #define [SYSCON\\_SYSPLLCTRL\\_DIRECTI](#) (1UL << SYSCON\_SYSPLLCTRL\_DIRECTI\_P)
- #define [SYSCON\\_SYSPLLCTRL\\_DIRECTO\\_P](#) 20
- #define [SYSCON\\_SYSPLLCTRL\\_DIRECTO](#) (1UL << SYSCON\_SYSPLLCTRL\_DIRECTO\_P)
- #define [SYSCON\\_SYSPLLSTAT\\_LOCK\\_P](#) 0
- #define [SYSCON\\_SYSPLLSTAT\\_LOCK](#) (1UL << SYSCON\_SYSPLLSTAT\_LOCK\_P)
- #define [PLL\\_CTRL\\_BYPASS\\_P](#) 15
- #define [PLL\\_CTRL\\_BYPASS\\_FBDIV2\\_P](#) 16
- #define [PLL\\_CTRL\\_UPLIMOFF\\_P](#) 17
- #define [PLL\\_CTRL\\_BANDSEL\\_SSCGREG\\_N\\_P](#) 18
- #define [PLL\\_CTRL\\_DIRECTI\\_P](#) 19
- #define [PLL\\_CTRL\\_DIRECTO\\_P](#) 20
- #define [PLL\\_CTRL\\_BYPASS](#) (1 << PLL\_CTRL\_BYPASS\_P)
- #define [PLL\\_CTRL\\_DIRECTI](#) (1 << PLL\_CTRL\_DIRECTI\_P)
- #define [PLL\\_CTRL\\_DIRECTO](#) (1 << PLL\_CTRL\_DIRECTO\_P)
- #define [PLL\\_CTRL\\_UPLIMOFF](#) (1 << PLL\_CTRL\_UPLIMOFF\_P)
- #define [PLL\\_CTRL\\_BANDSEL\\_SSCGREG\\_N](#) (1 << PLL\_CTRL\_BANDSEL\_SSCGREG\_N\_P)
- #define [PLL\\_CTRL\\_BYPASS\\_FBDIV2](#) (1 << PLL\_CTRL\_BYPASS\_FBDIV2\_P)
- #define [PLL\\_SSCG0\\_MREQ\\_P](#) 17
- #define [PLL\\_SSCG0\\_SEL\\_EXT\\_SSCG\\_N\\_P](#) 18
- #define [PLL\\_SSCG0\\_SEL\\_EXT\\_SSCG\\_N](#) (1 << PLL\_SSCG0\_SEL\_EXT\_SSCG\_N\_P)
- #define [PLL\\_SSCG0\\_MREQ](#) (1 << PLL\_SSCG0\_MREQ\_P)

- #define [PLL\\_SSCG1\\_MD\\_REQ\\_P](#) 19
- #define [PLL\\_SSCG1\\_MOD\\_PD\\_SSCGCLK\\_N\\_P](#) 28
- #define [PLL\\_SSCG1\\_DITHER\\_P](#) 29
- #define [PLL\\_SSCG1\\_MOD\\_PD\\_SSCGCLK\\_N](#) (1 << [PLL\\_SSCG1\\_MOD\\_PD\\_SSCGCLK\\_N\\_P](#))
- #define [PLL\\_SSCG1\\_DITHER](#) (1 << [PLL\\_SSCG1\\_DITHER\\_P](#))
- #define [PLL\\_SSCG1\\_MD\\_REQ](#) (1 << [PLL\\_SSCG1\\_MD\\_REQ\\_P](#))
- #define [PLL\\_NDEC\\_VAL\\_SET](#)(value) (((unsigned long) (value) << [PLL\\_NDEC\\_VAL\\_P](#)) & [PLL\\_NDEC\\_VAL\\_M](#))
- #define [PLL\\_NDEC\\_NREQ\\_P](#) 10
- #define [PLL\\_NDEC\\_NREQ](#) (1 << [PLL\\_NDEC\\_NREQ\\_P](#))
- #define [PLL\\_PDEC\\_VAL\\_SET](#)(value) (((unsigned long) (value) << [PLL\\_PDEC\\_VAL\\_P](#)) & [PLL\\_PDEC\\_VAL\\_M](#))
- #define [PLL\\_PDEC\\_PREQ\\_P](#) 7
- #define [PLL\\_PDEC\\_PREQ](#) (1 << [PLL\\_PDEC\\_PREQ\\_P](#))
- #define [PLL\\_SSCG0\\_MDEC\\_VAL\\_SET](#)(value) (((unsigned long) (value) << [PLL\\_SSCG0\\_MDEC\\_VAL\\_P](#)) & [PLL\\_SSCG0\\_MDEC\\_VAL\\_M](#))
- #define [PLL\\_SSCG0\\_MREQ\\_P](#) 17
- #define [PLL\\_SSCG0\\_MREQ](#) (1 << [PLL\\_SSCG0\\_MREQ\\_P](#))
- #define [PLL\\_SSCG0\\_SEL\\_EXT\\_SSCG\\_N\\_P](#) 18
- #define [PLL\\_SSCG0\\_SEL\\_EXT\\_SSCG\\_N](#) (1 << [PLL\\_SSCG0\\_SEL\\_EXT\\_SSCG\\_N\\_P](#))
- #define [PLL\\_SSCG1\\_MD\\_FRACT\\_P](#) 0
- #define [PLL\\_SSCG1\\_MD\\_INT\\_P](#) 11
- #define [PLL\\_SSCG1\\_MF\\_P](#) 20
- #define [PLL\\_SSCG1\\_MC\\_P](#) 26
- #define [PLL\\_SSCG1\\_MR\\_P](#) 23
- #define [PLL\\_SSCG1\\_MD\\_FRACT\\_M](#) (0x7FFUL << [PLL\\_SSCG1\\_MD\\_FRACT\\_P](#))
- #define [PLL\\_SSCG1\\_MD\\_INT\\_M](#) (0xFFUL << [PLL\\_SSCG1\\_MD\\_INT\\_P](#))
- #define [PLL\\_SSCG1\\_MF\\_M](#) (0x7UL << [PLL\\_SSCG1\\_MF\\_P](#))
- #define [PLL\\_SSCG1\\_MC\\_M](#) (0x3UL << [PLL\\_SSCG1\\_MC\\_P](#))
- #define [PLL\\_SSCG1\\_MR\\_M](#) (0x7UL << [PLL\\_SSCG1\\_MR\\_P](#))
- #define [PLL\\_SSCG1\\_MD\\_FRACT\\_SET](#)(value)
- #define [PLL\\_SSCG1\\_MD\\_INT\\_SET](#)(value)
- #define [PLL0\\_SSCG\\_MF\\_FREQ\\_VALUE](#) 4
- #define [PLL0\\_SSCG\\_MC\\_COMP\\_VALUE](#) 2
- #define [PLL0\\_SSCG\\_MR\\_DEPTH\\_VALUE](#) 4
- #define [PLL0\\_SSCG\\_DITHER\\_VALUE](#) 0
- #define [PLL\\_MAX\\_N\\_DIV](#) 0x100

## Functions

- static uint32\_t [pllEncodeN](#) (uint32\_t N)
- static uint32\_t [pllDecodeN](#) (uint32\_t NDEC)
- static uint32\_t [pllEncodeP](#) (uint32\_t P)
- static uint32\_t [pllDecodeP](#) (uint32\_t PDEC)
- static uint32\_t [pllEncodeM](#) (uint32\_t M)
- static uint32\_t [pllDecodeM](#) (uint32\_t MDEC)
- static void [pllFindSel](#) (uint32\_t M, bool bypassFBDIV2, uint32\_t \*pSelP, uint32\_t \*pSelM, uint32\_t \*pSelR)
- uint32\_t [findPIIPreDiv](#) (uint32\_t ctrlReg, uint32\_t nDecReg)
- uint32\_t [findPIIPostDiv](#) (uint32\_t ctrlReg, uint32\_t pDecReg)
- uint32\_t [findPIIMult](#) (uint32\_t ctrlReg, uint32\_t mDecReg)
- static uint32\_t [FindGreatestCommonDivisor](#) (uint32\_t m, uint32\_t n)
- static [PLL\\_ERROR\\_T](#) [Chip\\_Clock\\_GetPLLConfig](#) (uint32\_t finHz, uint32\_t foutHz, [PLL\\_SETUP\\_T](#) \*pSetup, bool useFeedbackDiv2, bool useSS)
- static void [Chip\\_Clock\\_GetSystemPLLOutFromSetupUpdate](#) ([PLL\\_SETUP\\_T](#) \*pSetup)

- uint32\_t [Chip\\_Clock\\_GetSystemPLLInClockRate](#) (void)  
*Return System PLL input clock rate.*
- uint32\_t [Chip\\_Clock\\_GetSystemPLLOutFromSetup](#) (PLL\_SETUP\_T \*pSetup)  
*Return System PLL output clock rate from setup structure.*
- uint32\_t [Chip\\_Clock\\_GetStoredPLLClockRate](#) (void)  
*Get the rate of pll from the stored value.*
- void [Chip\\_Clock\\_SetStoredPLLClockRate](#) (uint32\_t rate)  
*Store the current PLL rate.*
- uint32\_t [Chip\\_Clock\\_GetSystemPLLOutClockRate](#) (bool recompute)  
*Return System PLL output clock rate.*
- void [Chip\\_Clock\\_SetBypassPLL](#) (bool bypass)  
*Enables and disables PLL bypass mode.*
- PLL\_ERROR\_T [Chip\\_Clock\\_SetupPLLData](#) (PLL\_CONFIG\_T \*pControl, PLL\_SETUP\_T \*pSetup)  
*Set PLL output based on the passed PLL setup data.*
- PLL\_ERROR\_T [Chip\\_Clock\\_SetupSystemPLLPrec](#) (PLL\_SETUP\_T \*pSetup)  
*Set PLL output from PLL setup structure (precise frequency)*
- PLL\_ERROR\_T [Chip\\_Clock\\_SetPLLFreq](#) (const PLL\_SETUP\_T \*pSetup)  
*Set PLL output from PLL setup structure (precise frequency)*
- void [Chip\\_Clock\\_SetupSystemPLL](#) (uint32\_t multiply\_by, uint32\_t input\_freq)  
*Set PLL output based on the multiplier and input frequency.*

## Variables

- static uint32\_t [curPIIRate](#)

## 8.67.1 Macro Definition Documentation

### 8.67.1.1 #define MVALMAX (0x8000)

Definition at line 40 of file pll\_5411x.c.

### 8.67.1.2 #define NVALMAX (0x100)

Definition at line 38 of file pll\_5411x.c.

### 8.67.1.3 #define PLL0\_SSCG\_DITHER\_VALUE 0

Definition at line 169 of file pll\_5411x.c.

### 8.67.1.4 #define PLL0\_SSCG\_MC\_COMP\_VALUE 2

Definition at line 167 of file pll\_5411x.c.

### 8.67.1.5 #define PLL0\_SSCG\_MF\_FREQ\_VALUE 4

Definition at line 166 of file pll\_5411x.c.

### 8.67.1.6 #define PLL0\_SSCG\_MR\_DEPTH\_VALUE 4

Definition at line 168 of file pll\_5411x.c.



**8.67.1.7   #define PLL\_CTRL\_BANDSEL\_SSCGREG\_N (1 << PLL\_CTRL\_BANDSEL\_SSCGREG\_N\_P)**

Definition at line 107 of file pll\_5411x.c.

**8.67.1.8   #define PLL\_CTRL\_BANDSEL\_SSCGREG\_N\_P 18**

Definition at line 99 of file pll\_5411x.c.

**8.67.1.9   #define PLL\_CTRL\_BYPASS (1 << PLL\_CTRL\_BYPASS\_P)**

Definition at line 103 of file pll\_5411x.c.

**8.67.1.10   #define PLL\_CTRL\_BYPASS\_FBDIV2 (1 << PLL\_CTRL\_BYPASS\_FBDIV2\_P)**

Definition at line 108 of file pll\_5411x.c.

**8.67.1.11   #define PLL\_CTRL\_BYPASS\_FBDIV2\_P 16**

Definition at line 97 of file pll\_5411x.c.

**8.67.1.12   #define PLL\_CTRL\_BYPASS\_P 15**

Definition at line 96 of file pll\_5411x.c.

**8.67.1.13   #define PLL\_CTRL\_DIRECTI (1 << PLL\_CTRL\_DIRECTI\_P)**

Definition at line 104 of file pll\_5411x.c.

**8.67.1.14   #define PLL\_CTRL\_DIRECTI\_P 19**

Definition at line 100 of file pll\_5411x.c.

**8.67.1.15   #define PLL\_CTRL\_DIRECTO (1 << PLL\_CTRL\_DIRECTO\_P)**

Definition at line 105 of file pll\_5411x.c.

**8.67.1.16   #define PLL\_CTRL\_DIRECTO\_P 20**

Definition at line 101 of file pll\_5411x.c.

**8.67.1.17   #define PLL\_CTRL\_UPLIMOFF (1 << PLL\_CTRL\_UPLIMOFF\_P)**

Definition at line 106 of file pll\_5411x.c.

**8.67.1.18   #define PLL\_CTRL\_UPLIMOFF\_P 17**

Definition at line 98 of file pll\_5411x.c.

8.67.1.19 **#define PLL\_LOWER\_IN\_LIMIT (4000)**

Minimum PLL input rate

Definition at line 63 of file pll\_5411x.c.

8.67.1.20 **#define PLL\_MAX\_CCO\_FREQ\_MHZ (150000000)**

Definition at line 62 of file pll\_5411x.c.

8.67.1.21 **#define PLL\_MAX\_IN\_SSMODE (4000000)**

Definition at line 65 of file pll\_5411x.c.

8.67.1.22 **#define PLL\_MAX\_N\_DIV 0x100**

Definition at line 171 of file pll\_5411x.c.

8.67.1.23 **#define PLL\_MIN\_CCO\_FREQ\_MHZ (75000000)**

Definition at line 61 of file pll\_5411x.c.

8.67.1.24 **#define PLL\_MIN\_IN\_SSMODE (2000000)**

Definition at line 64 of file pll\_5411x.c.

8.67.1.25 **#define PLL\_NDEC\_NREQ (1 << PLL\_NDEC\_NREQ\_P)**

Definition at line 128 of file pll\_5411x.c.

8.67.1.26 **#define PLL\_NDEC\_NREQ\_P 10**

Definition at line 127 of file pll\_5411x.c.

8.67.1.27 **#define PLL\_NDEC\_VAL\_M (0x3FFUL << PLL\_NDEC\_VAL\_P)**

Definition at line 57 of file pll\_5411x.c.

8.67.1.28 **#define PLL\_NDEC\_VAL\_P (0)**

Definition at line 56 of file pll\_5411x.c.

8.67.1.29 **#define PLL\_NDEC\_VAL\_SET( value ) (((unsigned long) (value) << PLL\_NDEC\_VAL\_P) & PLL\_NDEC\_VAL\_M)**

Definition at line 126 of file pll\_5411x.c.

8.67.1.30 **#define PLL\_PDEC\_PREQ (1 << PLL\_PDEC\_PREQ\_P)**

Definition at line 133 of file pll\_5411x.c.

8.67.1.31 `#define PLL_PDEC_PREQ_P 7`

Definition at line 132 of file pll\_5411x.c.

8.67.1.32 `#define PLL_PDEC_VAL_M (0x3FFUL << PLL_PDEC_VAL_P)`

Definition at line 59 of file pll\_5411x.c.

8.67.1.33 `#define PLL_PDEC_VAL_P (0)`

Definition at line 58 of file pll\_5411x.c.

8.67.1.34 `#define PLL_PDEC_VAL_SET( value ) (((unsigned long) (value) << PLL_PDEC_VAL_P) & PLL_PDEC_VAL_M)`

Definition at line 131 of file pll\_5411x.c.

8.67.1.35 `#define PLL_SSCG0_MDEC_VAL_M (0x1FFFUL << PLL_SSCG0_MDEC_VAL_P)`

Definition at line 55 of file pll\_5411x.c.

8.67.1.36 `#define PLL_SSCG0_MDEC_VAL_P (0)`

Definition at line 54 of file pll\_5411x.c.

8.67.1.37 `#define PLL_SSCG0_MDEC_VAL_SET( value ) (((unsigned long) (value) << PLL_SSCG0_MDEC_VAL_P) & PLL_SSCG0_MDEC_VAL_M)`

Definition at line 136 of file pll\_5411x.c.

8.67.1.38 `#define PLL_SSCG0_MREQ (1 << PLL_SSCG0_MREQ_P)`

Definition at line 138 of file pll\_5411x.c.

8.67.1.39 `#define PLL_SSCG0_MREQ (1 << PLL_SSCG0_MREQ_P)`

Definition at line 138 of file pll\_5411x.c.

8.67.1.40 `#define PLL_SSCG0_MREQ_P 17`

Definition at line 137 of file pll\_5411x.c.

8.67.1.41 `#define PLL_SSCG0_MREQ_P 17`

Definition at line 137 of file pll\_5411x.c.

8.67.1.42 `#define PLL_SSCG0_SEL_EXT_SSCG_N (1 << PLL_SSCG0_SEL_EXT_SSCG_N_P)`

Definition at line 140 of file pll\_5411x.c.

8.67.1.43 **#define PLL\_SSCG0\_SEL\_EXT\_SSCG\_N**(1 << PLL\_SSCG0\_SEL\_EXT\_SSCG\_N\_P)

Definition at line 140 of file pll\_5411x.c.

8.67.1.44 **#define PLL\_SSCG0\_SEL\_EXT\_SSCG\_N\_P** 18

Definition at line 139 of file pll\_5411x.c.

8.67.1.45 **#define PLL\_SSCG0\_SEL\_EXT\_SSCG\_N\_P** 18

Definition at line 139 of file pll\_5411x.c.

8.67.1.46 **#define PLL\_SSCG1\_DITHER**(1 << PLL\_SSCG1\_DITHER\_P)

Definition at line 122 of file pll\_5411x.c.

8.67.1.47 **#define PLL\_SSCG1\_DITHER\_P** 29

Definition at line 120 of file pll\_5411x.c.

8.67.1.48 **#define PLL\_SSCG1\_MC\_M**(0x3UL << PLL\_SSCG1\_MC\_P)

Definition at line 152 of file pll\_5411x.c.

8.67.1.49 **#define PLL\_SSCG1\_MC\_P** 26

Definition at line 146 of file pll\_5411x.c.

8.67.1.50 **#define PLL\_SSCG1\_MD\_FRACT\_M**(0x7FFUL << PLL\_SSCG1\_MD\_FRACT\_P)

Definition at line 149 of file pll\_5411x.c.

8.67.1.51 **#define PLL\_SSCG1\_MD\_FRACT\_P** 0

Definition at line 143 of file pll\_5411x.c.

8.67.1.52 **#define PLL\_SSCG1\_MD\_FRACT\_SET**( value )

**Value:**

```
((unsigned long) (value) << \
    PLL_SSCG1_MD_FRACT_P) &
    PLL_SSCG1_MD_FRACT_M)
```

Definition at line 155 of file pll\_5411x.c.

8.67.1.53 **#define PLL\_SSCG1\_MD\_INT\_M**(0xFFUL << PLL\_SSCG1\_MD\_INT\_P)

Definition at line 150 of file pll\_5411x.c.

8.67.1.54 **#define** PLL\_SSCG1\_MD\_INT\_P 11

Definition at line 144 of file pll\_5411x.c.

8.67.1.55 **#define** PLL\_SSCG1\_MD\_INT\_SET( *value* )

**Value:**

```
((unsigned long) (value) << \
                    PLL_SSCG1_MD_INT_P) &
    PLL_SSCG1_MD_INT_M)
```

Definition at line 157 of file pll\_5411x.c.

8.67.1.56 **#define** PLL\_SSCG1\_MD\_REQ (1 << PLL\_SSCG1\_MD\_REQ\_P)

Definition at line 123 of file pll\_5411x.c.

8.67.1.57 **#define** PLL\_SSCG1\_MD\_REQ\_P 19

Definition at line 118 of file pll\_5411x.c.

8.67.1.58 **#define** PLL\_SSCG1\_MF\_M (0x7UL << PLL\_SSCG1\_MF\_P)

Definition at line 151 of file pll\_5411x.c.

8.67.1.59 **#define** PLL\_SSCG1\_MF\_P 20

Definition at line 145 of file pll\_5411x.c.

8.67.1.60 **#define** PLL\_SSCG1\_MOD\_PD\_SSCGCLK\_N (1 << PLL\_SSCG1\_MOD\_PD\_SSCGCLK\_N\_P)

Definition at line 121 of file pll\_5411x.c.

8.67.1.61 **#define** PLL\_SSCG1\_MOD\_PD\_SSCGCLK\_N\_P 28

Definition at line 119 of file pll\_5411x.c.

8.67.1.62 **#define** PLL\_SSCG1\_MR\_M (0x7UL << PLL\_SSCG1\_MR\_P)

Definition at line 153 of file pll\_5411x.c.

8.67.1.63 **#define** PLL\_SSCG1\_MR\_P 23

Definition at line 147 of file pll\_5411x.c.

8.67.1.64 **#define** PLL\_SSCG\_DITHER\_VALUE 0

Definition at line 71 of file pll\_5411x.c.

8.67.1.65 `#define PLL_SSCG_MC_COMP_VALUE 2`

Definition at line 69 of file pll\_5411x.c.

8.67.1.66 `#define PLL_SSCG_MF_FREQ_VALUE 4`

Definition at line 68 of file pll\_5411x.c.

8.67.1.67 `#define PLL_SSCG_MR_DEPTH_VALUE 4`

Definition at line 70 of file pll\_5411x.c.

8.67.1.68 `#define PVALMAX (0x20)`

Definition at line 39 of file pll\_5411x.c.

8.67.1.69 `#define SYS_PLL_BANDSEL (1 << 18)`

Enable MDEC control

Definition at line 49 of file pll\_5411x.c.

8.67.1.70 `#define SYS_PLL_BYPASS (1 << 15)`

Enable PLL bypass

Definition at line 46 of file pll\_5411x.c.

8.67.1.71 `#define SYS_PLL_BYPASSCCODIV2 (1 << 16)`

Enable bypass of extra divider by 2

Definition at line 47 of file pll\_5411x.c.

8.67.1.72 `#define SYS_PLL_DIRECTI (1 << 19)`

PLL0 direct input enable

Definition at line 50 of file pll\_5411x.c.

8.67.1.73 `#define SYS_PLL_DIRECTO (1 << 20)`

PLL0 direct output enable

Definition at line 51 of file pll\_5411x.c.

8.67.1.74 `#define SYS_PLL_SEL( d ) (((d) & 0x3f) << 4)`

Bandwidth select I value

Definition at line 44 of file pll\_5411x.c.

**8.67.1.75** `#define SYS_PLL_SELP( d ) (((d) & 0x1f) << 10)`

Bandwidth select P value

Definition at line 45 of file pll\_5411x.c.

**8.67.1.76** `#define SYS_PLL_SELRL( d ) (((d) & 0xf) << 0)`

Bandwidth select R value

Definition at line 43 of file pll\_5411x.c.

**8.67.1.77** `#define SYS_PLL_UPLIMOFF (1 << 17)`

Enable spread spectrum/fractional mode

Definition at line 48 of file pll\_5411x.c.

**8.67.1.78** `#define SYSCON_SYSPLLCTRL_BANDSEL_SSCGREG_N (1UL << SYSCON_SYSPLLCTRL_BANDSEL_SSCGREG↵  
_N_P)`

Definition at line 87 of file pll\_5411x.c.

**8.67.1.79** `#define SYSCON_SYSPLLCTRL_BANDSEL_SSCGREG_N_P 18`

Definition at line 86 of file pll\_5411x.c.

**8.67.1.80** `#define SYSCON_SYSPLLCTRL_BYPASS (1UL << SYSCON_SYSPLLCTRL_BYPASS_P)`

Definition at line 81 of file pll\_5411x.c.

**8.67.1.81** `#define SYSCON_SYSPLLCTRL_BYPASS_FBDIV2 (1UL << SYSCON_SYSPLLCTRL_BYPASS_FBDIV2_P)`

Definition at line 83 of file pll\_5411x.c.

**8.67.1.82** `#define SYSCON_SYSPLLCTRL_BYPASS_FBDIV2_P 16`

Definition at line 82 of file pll\_5411x.c.

**8.67.1.83** `#define SYSCON_SYSPLLCTRL_BYPASS_P 15`

Definition at line 80 of file pll\_5411x.c.

**8.67.1.84** `#define SYSCON_SYSPLLCTRL_DIRECTI (1UL << SYSCON_SYSPLLCTRL_DIRECTI_P)`

Definition at line 89 of file pll\_5411x.c.

**8.67.1.85** `#define SYSCON_SYSPLLCTRL_DIRECTI_P 19`

Definition at line 88 of file pll\_5411x.c.

**8.67.1.86 #define SYSCON\_SYSPLLCTRL\_DIRECTO (1UL << SYSCON\_SYSPLLCTRL\_DIRECTO\_P)**

Definition at line 91 of file pll\_5411x.c.

**8.67.1.87 #define SYSCON\_SYSPLLCTRL\_DIRECTO\_P 20**

Definition at line 90 of file pll\_5411x.c.

**8.67.1.88 #define SYSCON\_SYSPLLCTRL\_SEL\_I\_M (0x3FUL << SYSCON\_SYSPLLCTRL\_SEL\_I\_P)**

Definition at line 77 of file pll\_5411x.c.

**8.67.1.89 #define SYSCON\_SYSPLLCTRL\_SEL\_I\_P 4**

Definition at line 76 of file pll\_5411x.c.

**8.67.1.90 #define SYSCON\_SYSPLLCTRL\_SEL\_P\_M (0x1FUL << SYSCON\_SYSPLLCTRL\_SEL\_P\_P)**

Definition at line 79 of file pll\_5411x.c.

**8.67.1.91 #define SYSCON\_SYSPLLCTRL\_SEL\_P\_P 10**

Definition at line 78 of file pll\_5411x.c.

**8.67.1.92 #define SYSCON\_SYSPLLCTRL\_SEL\_R\_M (0xFUL << SYSCON\_SYSPLLCTRL\_SEL\_R\_P)**

Definition at line 75 of file pll\_5411x.c.

**8.67.1.93 #define SYSCON\_SYSPLLCTRL\_SEL\_R\_P 0**

Definition at line 74 of file pll\_5411x.c.

**8.67.1.94 #define SYSCON\_SYSPLLCTRL\_UPLIMOFF (1UL << SYSCON\_SYSPLLCTRL\_UPLIMOFF\_P)**

Definition at line 85 of file pll\_5411x.c.

**8.67.1.95 #define SYSCON\_SYSPLLCTRL\_UPLIMOFF\_P 17**

Definition at line 84 of file pll\_5411x.c.

**8.67.1.96 #define SYSCON\_SYSPLLSTAT\_LOCK (1UL << SYSCON\_SYSPLLSTAT\_LOCK\_P)**

Definition at line 94 of file pll\_5411x.c.

**8.67.1.97 #define SYSCON\_SYSPLLSTAT\_LOCK\_P 0**

Definition at line 93 of file pll\_5411x.c.



## 8.67.2 Function Documentation

**8.67.2.1** `static PLL_ERROR_T Chip_Clock_GetPllConfig ( uint32_t finHz, uint32_t fouthHz, PLL_SETUP_T * pSetup, bool useFeedbackDiv2, bool useSS ) [static]`

Definition at line 488 of file pll\_5411x.c.

**8.67.2.2** `static void Chip_Clock_GetSystemPLLOutFromSetupUpdate ( PLL_SETUP_T * pSetup ) [static]`

Definition at line 643 of file pll\_5411x.c.

**8.67.2.3** `static uint32_t FindGreatestCommonDivisor ( uint32_t m, uint32_t n ) [static]`

Definition at line 474 of file pll\_5411x.c.

**8.67.2.4** `uint32_t findPiIMMult ( uint32_t ctrlReg, uint32_t mDecReg )`

Definition at line 455 of file pll\_5411x.c.

**8.67.2.5** `uint32_t findPiIPostDiv ( uint32_t ctrlReg, uint32_t pDecReg )`

Definition at line 437 of file pll\_5411x.c.

**8.67.2.6** `uint32_t findPiIPreDiv ( uint32_t ctrlReg, uint32_t nDecReg )`

Definition at line 419 of file pll\_5411x.c.

**8.67.2.7** `static uint32_t pllDecodeM ( uint32_t MDEC ) [static]`

Definition at line 346 of file pll\_5411x.c.

**8.67.2.8** `static uint32_t pllDecodeN ( uint32_t NDEC ) [static]`

Definition at line 216 of file pll\_5411x.c.

**8.67.2.9** `static uint32_t pllDecodeP ( uint32_t PDEC ) [static]`

Definition at line 281 of file pll\_5411x.c.

**8.67.2.10** `static uint32_t pllEncodeM ( uint32_t M ) [static]`

Definition at line 316 of file pll\_5411x.c.

**8.67.2.11** `static uint32_t pllEncodeN ( uint32_t N ) [static]`

Definition at line 186 of file pll\_5411x.c.

**8.67.2.12** `static uint32_t pllEncodeP ( uint32_t P ) [static]`

Definition at line 251 of file pll\_5411x.c.

**8.67.2.13** `static void pllFindSel ( uint32_t M, bool bypassFBDIV2, uint32_t * pSelP, uint32_t * pSelI, uint32_t * pSelR )`  
`[static]`

Definition at line 381 of file `pll_5411x.c`.

### 8.67.3 Variable Documentation

**8.67.3.1** `uint32_t curPIIRate` `[static]`

Definition at line 175 of file `pll_5411x.c`.

## 8.68 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/ring\_buffer.c File Reference

```
#include <string.h>
#include "ring_buffer.h"
```

### Macros

- `#define RB_INDH(rb) ((rb)->head & ((rb)->count - 1))`
- `#define RB_INDT(rb) ((rb)->tail & ((rb)->count - 1))`

### Functions

- `int RingBuffer_Init (RINGBUFF_T *RingBuff, void *buffer, int itemSize, int count, void *(*copyFunc)(void *dst, const void *src, uint32_t len))`  
*Initialize ring buffer.*
- `int RingBuffer_Insert (RINGBUFF_T *RingBuff, const void *data)`  
*Insert a single item into ring buffer.*
- `int RingBuffer_InsertMult (RINGBUFF_T *RingBuff, const void *data, int num)`  
*Insert an array of items into ring buffer.*
- `int RingBuffer_Pop (RINGBUFF_T *RingBuff, void *data)`  
*Pop an item from the ring buffer.*
- `int RingBuffer_PopMult (RINGBUFF_T *RingBuff, void *data, int num)`  
*Pop an array of items from the ring buffer.*

### 8.68.1 Macro Definition Documentation

**8.68.1.1** `#define RB_INDH( rb ) ((rb)->head & ((rb)->count - 1))`

Definition at line 39 of file `ring_buffer.c`.

**8.68.1.2** `#define RB_INDT( rb ) ((rb)->tail & ((rb)->count - 1))`

Definition at line 40 of file `ring_buffer.c`.

## 8.69 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/rtc\_ut.c File Reference

```
#include "rtc_ut.h"
```

## Macros

- #define [SECSPERMIN](#) (60)
- #define [MINSPERHOUR](#) (60)
- #define [SECSPERHOUR](#) ([SECSPERMIN](#) \* [MINSPERHOUR](#))
- #define [HOURSPERDAY](#) (24)
- #define [SECSPERDAY](#) ([SECSPERMIN](#) \* [MINSPERHOUR](#) \* [HOURSPERDAY](#))
- #define [DAYSPERWEEK](#) (7)
- #define [MONETHSPERYEAR](#) (12)
- #define [DAYSPERYEAR](#) (365)
- #define [DAYSPERLEAPYEAR](#) (366)

## Functions

- static void [GetDMLY](#) (int dayOff, struct tm \*pTime)
- void [ConvertRtcTime](#) (uint32\_t rtcTick, struct tm \*pTime)  
*Converts a RTC tick time to Universal time.*
- void [ConvertTimeRtc](#) (struct tm \*pTime, uint32\_t \*rtcTick)  
*Converts a Universal time to RTC tick time.*

## Variables

- static uint8\_t [daysPerMonth](#) [2][[MONETHSPERYEAR](#)]

### 8.69.1 Macro Definition Documentation

#### 8.69.1.1 #define DAYSPERLEAPYEAR (366)

Definition at line 48 of file rtc\_ut.c.

#### 8.69.1.2 #define DAYSPERWEEK (7)

Definition at line 45 of file rtc\_ut.c.

#### 8.69.1.3 #define DAYSPERYEAR (365)

Definition at line 47 of file rtc\_ut.c.

#### 8.69.1.4 #define HOURSPERDAY (24)

Definition at line 43 of file rtc\_ut.c.

#### 8.69.1.5 #define MINSPERHOUR (60)

Definition at line 41 of file rtc\_ut.c.

#### 8.69.1.6 #define MONETHSPERYEAR (12)

Definition at line 46 of file rtc\_ut.c.

#### 8.69.1.7 `#define SECSPERDAY (SECSPERMIN * MINSPERHOUR * HOURSPERDAY)`

Definition at line 44 of file `rtc_ut.c`.

#### 8.69.1.8 `#define SECSPERHOUR (SECSPERMIN * MINSPERHOUR)`

Definition at line 42 of file `rtc_ut.c`.

#### 8.69.1.9 `#define SECSPERMIN (60)`

Definition at line 40 of file `rtc_ut.c`.

### 8.69.2 Function Documentation

#### 8.69.2.1 `static void GetDMLY ( int dayOff, struct tm * pTime ) [static]`

Definition at line 66 of file `rtc_ut.c`.

### 8.69.3 Variable Documentation

#### 8.69.3.1 `uint8_t daysPerMonth[2][MONETHSPERYEAR] [static]`

**Initial value:**

```
= {
  {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
  {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
}
```

Definition at line 51 of file `rtc_ut.c`.

## 8.70 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/sct\_5411x.c File Reference

```
#include "chip.h"
```

### Functions

- void [Chip\\_SCT\\_Init](#) ([LPC\\_SCT\\_T](#) \*pSCT)  
*Initializes the State Configurable Timer.*
- void [Chip\\_SCT\\_DeInit](#) ([LPC\\_SCT\\_T](#) \*pSCT)  
*Deinitializes the State Configurable Timer.*
- void [Chip\\_SCT\\_SetClrControl](#) ([LPC\\_SCT\\_T](#) \*pSCT, [uint32\\_t](#) value, [FunctionalState](#) ena)  
*Set or Clear the Control register.*
- void [Chip\\_SCT\\_SetConflictResolution](#) ([LPC\\_SCT\\_T](#) \*pSCT, [uint8\\_t](#) outnum, [uint8\\_t](#) value)  
*Set the conflict resolution.*

## 8.71 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/sct\_pwm\_5411x.c File Reference

```
#include "chip.h"
```

## Functions

- void [Chip\\_SCTPWM\\_SetOutPin](#) (LPC\_SCT\_T \*pSCT, uint8\_t index, uint8\_t pin)  
*Setup the OUTPUT pin and associate it with an index.*
- void [Chip\\_SCTPWM\\_SetRate](#) (LPC\_SCT\_T \*pSCT, uint32\_t freq)  
*Sets the frequency of the generated PWM wave.*

## 8.72 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/spi\_common\_5411x.c File Reference

```
#include "chip.h"
```

## Functions

- int [Chip\\_SPI\\_Init](#) (LPC\_SPI\_T \*pSPI)  
*Initialize the SPI.*
- void [Chip\\_SPI\\_ConfigureSPI](#) (LPC\_SPI\_T \*pSPI, SPI\_CFGSETUP\_T \*pCFG)  
*Setup SPI configuration.*

## 8.73 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/spim\_5411x.c File Reference

```
#include "chip.h"
```

## Functions

- static uint16\_t [get\\_tx\\_data](#) (SPIM\_XFER\_T \*xfer)
- static void [put\\_rx\\_data](#) (SPIM\_XFER\_T \*xfer, uint16\_t data)
- static uint16\_t [spim\\_get\\_xfercfg](#) (const SPIM\_XFER\_T \*xfer)
- static void [spim\\_do\\_txrx](#) (LPC\_SPI\_T \*pSPI, uint32\_t stat, SPIM\_XFER\_T \*xfer)
- uint32\_t [Chip\\_SPIM\\_SetClockRate](#) (LPC\_SPI\_T \*pSPI, uint32\_t rate)  
*Set SPI master bit rate.*
- void [Chip\\_SPIM\\_DelayConfig](#) (LPC\_SPI\_T \*pSPI, SPIM\_DELAY\_CONFIG\_T \*pConfig)  
*Config SPI Delay parameters.*
- void [Chip\\_SPIM\\_XferHandler](#) (LPC\_SPI\_T \*pSPI, SPIM\_XFER\_T \*xfer)  
*SPI master transfer state change handler.*
- void [Chip\\_SPIM\\_Xfer](#) (LPC\_SPI\_T \*pSPI, SPIM\_XFER\_T \*xfer)  
*Start non-blocking SPI master transfer.*
- void [Chip\\_SPIM\\_XferFIFO](#) (LPC\_SPI\_T \*pSPI, SPIM\_XFER\_T \*xfer)  
*Start polled SPI master transfer.*
- void [Chip\\_SPIM\\_XferBlocking](#) (LPC\_SPI\_T \*pSPI, SPIM\_XFER\_T \*xfer)  
*Perform blocking SPI master transfer.*

### 8.73.1 Function Documentation

#### 8.73.1.1 static uint16\_t get\_tx\_data ( SPIM\_XFER\_T \* xfer ) [static]

Definition at line 46 of file spim\_5411x.c.

**8.73.1.2** `static void put_rx_data ( SPIM_XFER_T * xfer, uint16_t data ) [static]`

Definition at line 66 of file `spim_5411x.c`.

**8.73.1.3** `static void spim_do_txx ( LPC_SPI_T * pSPI, uint32_t stat, SPIM_XFER_T * xfer ) [static]`

Definition at line 101 of file `spim_5411x.c`.

**8.73.1.4** `static uint16_t spim_get_xfercfg ( const SPIM_XFER_T * xfer ) [static]`

Definition at line 83 of file `spim_5411x.c`.

## 8.74 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/spis\_5411x.c File Reference

```
#include "chip.h"
```

### Functions

- static uint16\_t [get\\_tx\\_data](#) (SPIS\_XFER\_T \*xfer)
- static void [put\\_rx\\_data](#) (SPIS\_XFER\_T \*xfer, uint16\_t data)
- static void [spis\\_do\\_txx](#) (LPC\_SPI\_T \*pSPI, uint32\_t stat, SPIS\_XFER\_T \*xfer)
- void [Chip\\_SPIS\\_Init](#) (LPC\_SPI\_T \*pSPI)  
*SPI slave initialization.*
- void [Chip\\_SPIS\\_EnableInts](#) (LPC\_SPI\_T \*pSPI)  
*SPI slave interrupt enable.*
- void [Chip\\_SPIS\\_DisableInts](#) (LPC\_SPI\_T \*pSPI)  
*SPI slave interrupt disable.*
- void [Chip\\_SPIS\\_LoadFIFO](#) (LPC\_SPI\_T \*pSPI, SPIS\_XFER\_T \*xfer)  
*Load slave transmit FIFO.*
- void [Chip\\_SPIS\\_ReadFIFO](#) (LPC\_SPI\_T \*pSPI, SPIS\_XFER\_T \*xfer)  
*SPI slave FIFO read.*
- void [Chip\\_SPIS\\_XferHandler](#) (LPC\_SPI\_T \*pSPI, SPIS\_XFER\_T \*xfer)  
*SPI slave transfer state change handler.*

### 8.74.1 Function Documentation

**8.74.1.1** `static uint16_t get_tx_data ( SPIS_XFER_T * xfer ) [static]`

Definition at line 47 of file `spis_5411x.c`.

**8.74.1.2** `static void put_rx_data ( SPIS_XFER_T * xfer, uint16_t data ) [static]`

Definition at line 68 of file `spis_5411x.c`.

**8.74.1.3** `static void spis_do_txx ( LPC_SPI_T * pSPI, uint32_t stat, SPIS_XFER_T * xfer ) [static]`

Definition at line 85 of file `spis_5411x.c`.

## 8.75 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/stopwatch\_5411x.c File Reference

```
#include "chip.h"
#include "stopwatch.h"
```

### Macros

- `#define LPC_TIMER32_1 LPC_TIMER0`

### Functions

- void `StopWatch_Init` (void)  
*Initialize stopwatch.*
- uint32\_t `StopWatch_Start` (void)  
*Start a stopwatch.*
- uint32\_t `StopWatch_TicksPerSecond` (void)  
*Returns number of ticks per second of the stopwatch timer.*
- uint32\_t `StopWatch_TicksToMs` (uint32\_t ticks)  
*Converts from stopwatch ticks to mS.*
- uint32\_t `StopWatch_TicksToUs` (uint32\_t ticks)  
*Converts from stopwatch ticks to uS.*
- uint32\_t `StopWatch_MsToTicks` (uint32\_t mS)  
*Converts from mS to stopwatch ticks.*
- uint32\_t `StopWatch_UsToTicks` (uint32\_t uS)  
*Converts from uS to stopwatch ticks.*

### Variables

- static uint32\_t `ticksPerSecond`
- static uint32\_t `ticksPerMs`
- static uint32\_t `ticksPerUs`

#### 8.75.1 Macro Definition Documentation

##### 8.75.1.1 `#define LPC_TIMER32_1 LPC_TIMER0`

Definition at line 45 of file stopwatch\_5411x.c.

#### 8.75.2 Variable Documentation

##### 8.75.2.1 `uint32_t ticksPerMs` `[static]`

Definition at line 41 of file stopwatch\_5411x.c.

##### 8.75.2.2 `uint32_t ticksPerSecond` `[static]`

Definition at line 40 of file stopwatch\_5411x.c.

### 8.75.2.3 uint32\_t ticksPerUs [static]

Definition at line 42 of file stopwatch\_5411x.c.

## 8.76 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/syscon\_5411x.c File Reference

```
#include "chip.h"
```

### Functions

- void [Chip\\_SYSCON\\_SetNMISource](#) (uint32\_t intsrc)  
*Set source for non-maskable interrupt (NMI)*
- void [Chip\\_SYSCON\\_EnableNMISource](#) (void)  
*Enable interrupt used for NMI source.*
- void [Chip\\_SYSCON\\_DisableNMISource](#) (void)  
*Disable interrupt used for NMI source.*
- void [Chip\\_SYSCON\\_Enable\\_ASYNC\\_Syscon](#) (bool enable)  
*Enable or disable asynchronous APB bridge and subsystem.*
- uint32\_t [Chip\\_SYSCON\\_GetCompFreqMeas](#) (uint32\_t refClockRate)  
*Returns the computed value for a frequency measurement cycle.*
- void [Chip\\_SYSCON\\_PowerUp](#) (uint32\_t powerupmask)  
*Power up one or more blocks or peripherals.*

## 8.77 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/sysinit\_5411x.c File Reference

```
#include "chip.h"
```

### Functions

- static void [setupFlashClocks](#) (uint32\_t freq)
- void [Chip\\_SetupFROClocking](#) (uint32\_t iFreq)  
*Initialize the Core clock to given frequency (12, 48 or 96 MHz)*
- void [Chip\\_SetupIrcClocking](#) (uint32\_t iFreq)  
*Clock and PLL initialization based on the internal oscillator.*
- void [Chip\\_SetupExtInClocking](#) (uint32\_t iFreq)  
*Clock and PLL initialization based on the external clock input.*
- void [Chip\\_SystemInit](#) (void)  
*Set up and initialize hardware prior to call to main()*

### 8.77.1 Function Documentation

#### 8.77.1.1 static void setupFlashClocks ( uint32\_t freq ) [static]

Definition at line 47 of file sysinit\_5411x.c.



## 8.78 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/timer\_5411x.c File Reference

```
#include "chip.h"
```

### Macros

- `#define LAST_TIMER` (4)

### Functions

- void `Chip_TIMER_Reset` (`LPC_TIMER_T *pTMR`)  
*Resets the timer terminal and prescale counts to 0.*
- void `Chip_TIMER_ExtMatchControlSet` (`LPC_TIMER_T *pTMR`, `int8_t initial_state`, `TIMER_PIN_MATCH_←_STATE_T matchState`, `int8_t matchnum`)  
*Sets external match control (MATn.matchnum) pin control. For the pin selected with matchnum, sets the function of the pin that occurs on a terminal count match for the match count.*

### 8.78.1 Macro Definition Documentation

#### 8.78.1.1 `#define LAST_TIMER` (4)

Definition at line 37 of file timer\_5411x.c.

## 8.79 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/uart\_5411x.c File Reference

```
#include "chip.h"
```

### Functions

- static `uint32_t _UART_DivClk` (`uint32_t pclk`, `uint32_t m`)
- static `uint32_t _UART_GetHighDiv` (`uint32_t val`, `uint8_t strict`)
- static `int32_t _CalcErr` (`uint32_t n`, `uint32_t d`, `uint32_t *prev`)
- static `ErrorCode_t _UART_CalcDiv` (`UART_BAUD_T *ub`)
- static void `_UART_CalcMul` (`UART_BAUD_T *ub`)
- int `Chip_UART_Init` (`LPC_USART_T *pUART`)  
*Initialize the UART peripheral.*
- void `Chip_UART_DeInit` (`LPC_USART_T *pUART`)  
*Deinitialize the UART peripheral.*
- void `Chip_UART_ConfigDMA` (`LPC_USART_T *pUART`)  
*Configure UART for DMA.*
- int `Chip_UART_Send` (`LPC_USART_T *pUART`, `const void *data`, `int numBytes`)  
*Transmit a byte array through the UART peripheral (non-blocking)*
- int `Chip_UART_SendBlocking` (`LPC_USART_T *pUART`, `const void *data`, `int numBytes`)  
*Transmit a byte array through the UART peripheral (blocking)*
- int `Chip_UART_Read` (`LPC_USART_T *pUART`, `void *data`, `int numBytes`)  
*Read data through the UART peripheral (non-blocking)*
- int `Chip_UART_ReadBlocking` (`LPC_USART_T *pUART`, `void *data`, `int numBytes`)  
*Read data through the UART peripheral (blocking)*

- int [Chip\\_UART\\_CalculateBaud](#) ([UART\\_BAUD\\_T](#) \*baud)
- uint32\_t [Chip\\_UART\\_SetBaud](#) ([LPC\\_USART\\_T](#) \*pUART, uint32\_t baudrate)  
*Set baud rate for UART.*
- void [Chip\\_UART\\_RXIntHandlerRB](#) ([LPC\\_USART\\_T](#) \*pUART, [RINGBUFF\\_T](#) \*pRB)  
*UART receive-only interrupt handler for ring buffers.*
- void [Chip\\_UART\\_TXIntHandlerRB](#) ([LPC\\_USART\\_T](#) \*pUART, [RINGBUFF\\_T](#) \*pRB)  
*UART transmit-only interrupt handler for ring buffers.*
- uint32\_t [Chip\\_UART\\_SendRB](#) ([LPC\\_USART\\_T](#) \*pUART, [RINGBUFF\\_T](#) \*pRB, const void \*data, int count)  
*Populate a transmit ring buffer and start UART transmit.*
- int [Chip\\_UART\\_ReadRB](#) ([LPC\\_USART\\_T](#) \*pUART, [RINGBUFF\\_T](#) \*pRB, void \*data, int bytes)  
*Copy data from a receive ring buffer.*
- void [Chip\\_UART\\_IRQHandlerRB](#) ([LPC\\_USART\\_T](#) \*pUART, [UART\\_STATISTICS\\_T](#) \*stat, [RINGBUFF\\_T](#) \*pRXRB, [RINGBUFF\\_T](#) \*pTXRB)  
*UART receive/transmit interrupt handler for ring buffers.*
- void [Chip\\_UART\\_IRQHandlerDMA](#) ([LPC\\_USART\\_T](#) \*pUART, [UART\\_STATISTICS\\_T](#) \*stat)  
*UART receive/transmit interrupt handler for DMA.*

## Variables

- static const uint32\_t [fifo\\_config\\_int](#) = ([UART\\_FIFOCFG\\_ENABLETX](#) | [UART\\_FIFOCFG\\_ENABLERX](#))
- static const uint32\_t [fifo\\_config\\_dma](#) = ([UART\\_FIFOCFG\\_ENABLETX](#) | [UART\\_FIFOCFG\\_ENABLERX](#) | [UART\\_FIFOCFG\\_DMATX](#) | [UART\\_FIFOCFG\\_DMARX](#))

## 8.79.1 Function Documentation

8.79.1.1 `static int32_t _CalcErr ( uint32_t n, uint32_t d, uint32_t * prev )` [static]

Definition at line 70 of file `uart_5411x.c`.

8.79.1.2 `static ErrorCode_t _UART_CalcDiv ( UART_BAUD_T * ub )` [static]

Definition at line 86 of file `uart_5411x.c`.

8.79.1.3 `static void _UART_CalcMul ( UART_BAUD_T * ub )` [static]

Definition at line 124 of file `uart_5411x.c`.

8.79.1.4 `static uint32_t _UART_DivClk ( uint32_t pclk, uint32_t m )` [static]

Definition at line 48 of file `uart_5411x.c`.

8.79.1.5 `static uint32_t _UART_GetHighDiv ( uint32_t val, uint8_t strict )` [static]

Definition at line 58 of file `uart_5411x.c`.

8.79.1.6 `int Chip_UART_CalculateBaud ( UART_BAUD_T * baud )`

Definition at line 242 of file `uart_5411x.c`.

## 8.79.2 Variable Documentation

8.79.2.1 `const uint32_t fifo_config_dma = (UART_FIFOCFG_ENABLETX | UART_FIFOCFG_ENABLERX | UART_FIFOCFG_DMATX | UART_FIFOCFG_DMARX) [static]`

Definition at line 38 of file `uart_5411x.c`.

8.79.2.2 `const uint32_t fifo_config_int = (UART_FIFOCFG_ENABLETX | UART_FIFOCFG_ENABLERX) [static]`

Definition at line 37 of file `uart_5411x.c`.

## 8.80 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/cr\_startup\_lpc5411x-m0.c File Reference

### Macros

- `#define WEAK __attribute__ ((weak))`
- `#define ALIAS(f) __attribute__ ((weak, alias (#f)))`

### Functions

- void `ResetISR` (void)
- `WEAK` void `NMI_Handler` (void)
- `WEAK` void `HardFault_Handler` (void)
- `WEAK` void `SVC_Handler` (void)
- `WEAK` void `PendSV_Handler` (void)
- `WEAK` void `SysTick_Handler` (void)
- `WEAK` void `IntDefaultHandler` (void)
- void `WDT_BOD_IRQHandler` (void DMA\_IRQHandler() `ALIAS`(`IntDefaultHandler`) void)
- `__attribute__ ((section(".after_vectors")))`
- `__attribute__ ((section(".after_vectors.reset")))`

### Variables

- unsigned int `__data_section_table`
- unsigned int `__data_section_table_end`
- unsigned int `__bss_section_table`
- unsigned int `__bss_section_table_end`

## 8.80.1 Macro Definition Documentation

8.80.1.1 `#define ALIAS( f ) __attribute__ ((weak, alias (#f)))`

Definition at line 48 of file `cr_startup_lpc5411x-m0.c`.

8.80.1.2 `#define WEAK __attribute__ ((weak))`

Definition at line 47 of file `cr_startup_lpc5411x-m0.c`.

## 8.80.2 Function Documentation

### 8.80.2.1 `__attribute__ ( (section("after_vectors")) )`

Definition at line 234 of file `cr_startup_lpc5411x-m0.c`.

### 8.80.2.2 `__attribute__ ( (section("after_vectors.reset")) )`

Definition at line 328 of file `cr_startup_lpc5411x-m0.c`.

### 8.80.2.3 `WEAK void HardFault_Handler ( void )`

### 8.80.2.4 `WEAK void IntDefaultHandler ( void )`

### 8.80.2.5 `WEAK void NMI_Handler ( void )`

### 8.80.2.6 `WEAK void PendSV_Handler ( void )`

### 8.80.2.7 `void ResetISR ( void )`

### 8.80.2.8 `WEAK void SVC_Handler ( void )`

### 8.80.2.9 `WEAK void SysTick_Handler ( void )`

### 8.80.2.10 `void WDT_BOD_IRQHandler ( void DMA_IRQHandler() ALIAS(IntDefaultHandler) void )`

Definition at line 92 of file `cr_startup_lpc5411x-m0.c`.

## 8.80.3 Variable Documentation

### 8.80.3.1 `unsigned int __bss_section_table`

### 8.80.3.2 `unsigned int __bss_section_table_end`

### 8.80.3.3 `unsigned int __data_section_table`

### 8.80.3.4 `unsigned int __data_section_table_end`

## 8.81 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/cr\_startup\_lpc5411x.c File Reference

### Macros

- `#define WEAK __attribute__ ((weak))`
- `#define ALIAS(f) __attribute__ ((weak, alias (#f)))`

### Functions

- `void ResetISR (void)`
- `WEAK void NMI_Handler (void)`
- `WEAK void HardFault_Handler (void)`
- `WEAK void MemManage_Handler (void)`

- **WEAK** void [BusFault\\_Handler](#) (void)
- **WEAK** void [UsageFault\\_Handler](#) (void)
- **WEAK** void [SVC\\_Handler](#) (void)
- **WEAK** void [DebugMon\\_Handler](#) (void)
- **WEAK** void [PendSV\\_Handler](#) (void)
- **WEAK** void [SysTick\\_Handler](#) (void)
- **WEAK** void [IntDefaultHandler](#) (void)
- void [WDT\\_BOD\\_IRQHandler](#) (void DMA\_IRQHandler() **ALIAS**([IntDefaultHandler](#)) void)
- [\\_\\_attribute\\_\\_](#)((section(".after\_vectors")))
- [\\_\\_attribute\\_\\_](#)((section(".after\_vectors.reset")))

## Variables

- unsigned int [\\_\\_SWVtrace\\_Enabled](#)
- unsigned int [\\_\\_data\\_section\\_table](#)
- unsigned int [\\_\\_data\\_section\\_table\\_end](#)
- unsigned int [\\_\\_bss\\_section\\_table](#)
- unsigned int [\\_\\_bss\\_section\\_table\\_end](#)

## 8.81.1 Macro Definition Documentation

### 8.81.1.1 `#define ALIAS( f ) __attribute__((weak, alias (#f)))`

Definition at line 48 of file `cr_startup_lpc5411x.c`.

### 8.81.1.2 `#define WEAK __attribute__((weak))`

Definition at line 47 of file `cr_startup_lpc5411x.c`.

## 8.81.2 Function Documentation

### 8.81.2.1 `__attribute__((section(".after_vectors")))`

Definition at line 248 of file `cr_startup_lpc5411x.c`.

### 8.81.2.2 `__attribute__((section(".after_vectors.reset")))`

Definition at line 340 of file `cr_startup_lpc5411x.c`.

### 8.81.2.3 `WEAK void BusFault_Handler ( void )`

### 8.81.2.4 `WEAK void DebugMon_Handler ( void )`

### 8.81.2.5 `WEAK void HardFault_Handler ( void )`

### 8.81.2.6 `WEAK void IntDefaultHandler ( void )`

### 8.81.2.7 `WEAK void MemManage_Handler ( void )`

### 8.81.2.8 `WEAK void NMI_Handler ( void )`

8.81.2.9 WEAK void PendSV\_Handler ( void )

8.81.2.10 void ResetISR ( void )

8.81.2.11 WEAK void SVC\_Handler ( void )

8.81.2.12 WEAK void SysTick\_Handler ( void )

8.81.2.13 WEAK void UsageFault\_Handler ( void )

8.81.2.14 void WDT\_BOD\_IRQHandler ( void DMA\_IRQHandler() ALIAS(IntDefaultHandler) void )

Definition at line 98 of file cr\_startup\_lpc5411x.c.

### 8.81.3 Variable Documentation

8.81.3.1 unsigned int \_\_bss\_section\_table

8.81.3.2 unsigned int \_\_bss\_section\_table\_end

8.81.3.3 unsigned int \_\_data\_section\_table

8.81.3.4 unsigned int \_\_data\_section\_table\_end

8.81.3.5 unsigned int \_\_SWVtrace\_Enabled

Definition at line 64 of file cr\_startup\_lpc5411x.c.

## 8.82 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/crp.c File Reference

## 8.83 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/mtb.c File Reference

## 8.84 C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/sysinit.c File Reference

```
#include "board.h"
```

### Functions

- void [SystemInit](#) (void)

### 8.84.1 Function Documentation

8.84.1.1 void SystemInit ( void )

Definition at line 59 of file sysinit.c.

## 8.85 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/community\_support.dox File Reference

- 8.86 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/copyright.dox File Reference
- 8.87 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/installation\_insts.dox File Reference
- 8.88 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/toolchains.dox File Reference
- 8.89 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/version\_history.dox File Reference
- 8.90 boards\_5411x.dox File Reference
- 8.91 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/chip\_5411x.dox File Reference
- 8.92 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/irq\_vector\_names.dox File Reference
- 8.93 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/lpcopen\_layout.dox File Reference
- 8.94 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/main\_layout.dox File Reference
- 8.95 C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/multicore.dox File Reference





# Index

- `_BIT`
    - LPC Public Macros, [475](#)
  - `_BITMASK`
    - LPC Public Macros, [475](#)
  - `_BIT_SHIFT`
    - NVIC Functions, [486](#)
  - `_BM_T`
    - Dir, [551](#)
    - Recipient, [551](#)
    - Type, [551](#)
  - `_CalcErr`
    - uart\_5411x.c, [804](#)
  - `_IP_IDX`
    - NVIC Functions, [486](#)
  - `_REQUEST_TYPE`
    - B, [632](#)
    - BM, [632](#)
  - `_SBF`
    - LPC Public Macros, [475](#)
  - `_SHP_IDX`
    - NVIC Functions, [487](#)
  - `_UART_CalcDiv`
    - uart\_5411x.c, [804](#)
  - `_UART_CalcMul`
    - uart\_5411x.c, [804](#)
  - `_UART_DivClk`
    - uart\_5411x.c, [804](#)
  - `_UART_GetHighDiv`
    - uart\_5411x.c, [804](#)
  - `_USB_COMMON_DESCRIPTOR`
    - bDescriptorType, [650](#)
    - bLength, [650](#)
  - `_USB_CONFIGURATION_DESCRIPTOR`
    - bConfigurationValue, [651](#)
    - bDescriptorType, [651](#)
    - bLength, [651](#)
    - bMaxPower, [652](#)
    - bNumInterfaces, [652](#)
    - bmAttributes, [651](#)
    - iConfiguration, [652](#)
    - wTotalLength, [652](#)
  - `_USB_DEVICE_DESCRIPTOR`
    - bDescriptorType, [653](#)
    - bDeviceClass, [653](#)
    - bDeviceProtocol, [653](#)
    - bDeviceSubClass, [654](#)
    - bLength, [654](#)
    - bMaxPacketSize0, [654](#)
    - bNumConfigurations, [654](#)
    - bcdDevice, [653](#)
    - bcdUSB, [653](#)
    - iManufacturer, [654](#)
    - iProduct, [654](#)
    - iSerialNumber, [654](#)
    - idProduct, [654](#)
    - idVendor, [654](#)
  - `_USB_DEVICE_QUALIFIER_DESCRIPTOR`
    - bDescriptorType, [655](#)
    - bDeviceClass, [655](#)
    - bDeviceProtocol, [655](#)
    - bDeviceSubClass, [655](#)
    - bLength, [656](#)
    - bMaxPacketSize0, [656](#)
    - bNumConfigurations, [656](#)
    - bReserved, [656](#)
    - bcdUSB, [655](#)
  - `_USB_ENDPOINT_DESCRIPTOR`
    - bDescriptorType, [657](#)
    - bEndpointAddress, [657](#)
    - bInterval, [657](#)
    - bLength, [657](#)
    - bmAttributes, [657](#)
    - wMaxPacketSize, [658](#)
  - `_USB_IAD_DESCRIPTOR`
    - bDescriptorType, [659](#)
    - bFirstInterface, [659](#)
    - bFunctionClass, [659](#)
    - bFunctionProtocol, [659](#)
    - bFunctionSubClass, [659](#)
    - bInterfaceCount, [659](#)
    - bLength, [660](#)
    - iFunction, [660](#)
  - `_USB_INTERFACE_DESCRIPTOR`
    - bAlternateSetting, [660](#)
    - bDescriptorType, [660](#)
    - bInterfaceClass, [660](#)
    - bInterfaceNumber, [661](#)
    - bInterfaceProtocol, [661](#)
    - bInterfaceSubClass, [661](#)
    - bLength, [661](#)
    - bNumEndpoints, [661](#)
    - iInterface, [661](#)
  - `_USB_OTHER_SPEED_CONFIGURATION`
    - bConfigurationValue, [662](#)
    - bDescriptorType, [662](#)
    - bLength, [662](#)
    - bMaxPower, [662](#)
    - bNumInterfaces, [662](#)

- bmAttributes, [662](#)
- lConfiguration, [663](#)
- wTotalLength, [663](#)
- \_\_USB\_SETUP\_PACKET
  - bRequest, [663](#)
  - bmRequestType, [663](#)
  - wIndex, [664](#)
  - wLength, [664](#)
  - wValue, [664](#)
- \_\_USB\_STRING\_DESCRIPTOR
  - bDescriptorType, [664](#)
  - bLength, [664](#)
  - bString, [665](#)
- \_\_WB\_T
  - H, [665](#)
  - L, [665](#)
- \_\_CM0PLUS\_CMSIS\_VERSION
  - core\_cm0plus.h, [684](#)
- \_\_CM0PLUS\_CMSIS\_VERSION\_MAIN
  - core\_cm0plus.h, [684](#)
- \_\_CM0PLUS\_CMSIS\_VERSION\_SUB
  - core\_cm0plus.h, [684](#)
- \_\_CM0PLUS\_REV
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [198](#)
- \_\_CM4\_CMSIS\_VERSION
  - core\_cm4.h, [694](#)
- \_\_CM4\_CMSIS\_VERSION\_MAIN
  - core\_cm4.h, [694](#)
- \_\_CM4\_CMSIS\_VERSION\_SUB
  - core\_cm4.h, [694](#)
- \_\_CM4\_REV
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [202](#)
- \_\_CORE\_CM0PLUS\_H\_DEPENDANT
  - core\_cm0plus.h, [684](#)
- \_\_CORE\_CM0PLUS\_H\_GENERIC
  - core\_cm0plus.h, [684](#)
- \_\_CORE\_CM4\_H\_DEPENDANT
  - core\_cm4.h, [694](#)
- \_\_CORE\_CM4\_H\_GENERIC
  - core\_cm4.h, [694](#)
- \_\_CORE\_CM4\_SIMD\_H
  - core\_cm4\_simd.h, [695](#)
- \_\_CORTEX\_M
  - core\_cm0plus.h, [684](#)
  - core\_cm4.h, [694](#)
- \_\_FPU\_PRESENT
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [202](#)
- \_\_FPU\_USED
  - core\_cm0plus.h, [684](#)
- \_\_I
  - core\_cm0plus.h, [684](#)
  - core\_cm4.h, [695](#)
- \_\_IO
  - core\_cm0plus.h, [684](#)
  - core\_cm4.h, [695](#)
- \_\_MPU\_PRESENT
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [198](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [202](#)
- \_\_NVIC\_PRIO\_BITS
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [198](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [202](#)
- \_\_O
  - core\_cm0plus.h, [685](#)
  - core\_cm4.h, [695](#)
- \_\_SWVtrace\_Enabled
  - cr\_startup\_lpc5411x.c, [808](#)
- \_\_VTOR\_PRESENT
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
- \_\_Vendor\_SysTickConfig
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [198](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [202](#)
- \_\_WORD\_BYTE
  - W, [666](#)
  - WB, [666](#)
- \_\_attribute\_\_
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [807](#)
- \_\_bss\_section\_table
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [808](#)
- \_\_bss\_section\_table\_end
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [808](#)
- \_\_data\_section\_table
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [808](#)
- \_\_data\_section\_table\_end
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [808](#)
- \_\_reserved0
  - APSR\_Type, [549](#)
  - CONTROL\_Type, [552](#)
  - IPSR\_Type, [565](#)
  - xPSR\_Type, [667](#)
- \_\_reserved1
  - APSR\_Type, [550](#)
  - xPSR\_Type, [667](#)
- ABORT
  - LPC\_DMA\_COMMON\_T, [576](#)
- ACPR
  - TPI\_Type, [645](#)
- ACTIVE
  - LPC\_DMA\_COMMON\_T, [576](#)
- ACTLR
  - SCnSCB\_Type, [637](#)
- ADC0\_SEQA\_IRQHandler

- CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
- CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- ADC0\_SEQA\_IRQn
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- ADC0\_SEQB\_IRQHandler
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- ADC0\_SEQB\_IRQn
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- ADC0\_THCMP\_IRQHandler
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- ADC0\_THCMP\_IRQn
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- ADC\_CALIB
  - CHIP: LPC5411X A/D conversion driver, [21](#)
- ADC\_CALREQD
  - CHIP: LPC5411X A/D conversion driver, [21](#)
- ADC\_CR\_ASYNC\_MODE
  - CHIP: LPC5411X A/D conversion driver, [21](#)
- ADC\_CR\_BITACC
  - CHIP: LPC5411X A/D conversion driver, [21](#)
- ADC\_CR\_BYPASS
  - CHIP: LPC5411X A/D conversion driver, [21](#)
- ADC\_CR\_CALMODEBIT
  - CHIP: LPC5411X A/D conversion driver, [21](#)
- ADC\_CR\_CLKDIV
  - CHIP: LPC5411X A/D conversion driver, [21](#)
- ADC\_CR\_CLKDIV\_BITPOS
  - CHIP: LPC5411X A/D conversion driver, [22](#)
- ADC\_CR\_CLKDIV\_MASK
  - CHIP: LPC5411X A/D conversion driver, [22](#)
- ADC\_CR\_LPWRMODEBIT
  - CHIP: LPC5411X A/D conversion driver, [22](#)
- ADC\_CR\_RESOL
  - CHIP: LPC5411X A/D conversion driver, [22](#)
- ADC\_CR\_TSAMP
  - CHIP: LPC5411X A/D conversion driver, [22](#)
- ADC\_DR\_CHAN\_BITPOS
  - CHIP: LPC5411X A/D conversion driver, [22](#)
- ADC\_DR\_CHAN\_MASK
  - CHIP: LPC5411X A/D conversion driver, [22](#)
- ADC\_DR\_CHANNEL
  - CHIP: LPC5411X A/D conversion driver, [22](#)
- CHIP: LPC5411X A/D conversion driver, [22](#)
- ADC\_DR\_DATAVALID
  - CHIP: LPC5411X A/D conversion driver, [22](#)
- ADC\_DR\_DONE
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_OVERRUN
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_RESULT
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_RESULT\_BITPOS
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_THCMPCROSS
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_THCMPCROSS\_BITPOS
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_THCMPCROSS\_DOWNWARD
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_DR\_THCMPCROSS\_MASK
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_THCMPCROSS\_NOCROSS
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_DR\_THCMPCROSS\_RESERVED
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_DR\_THCMPCROSS\_T
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_DR\_THCMPCROSS\_UPWARD
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_DR\_THCMPRANGE
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_THCMPRANGE\_ABOVE
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_DR\_THCMPRANGE\_BELOW
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_DR\_THCMPRANGE\_BITPOS
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_THCMPRANGE\_INRANGE
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_DR\_THCMPRANGE\_MASK
  - CHIP: LPC5411X A/D conversion driver, [23](#)
- ADC\_DR\_THCMPRANGE\_RESERVED
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_DR\_THCMPRANGE\_T
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_FLAGS\_OVERRUN\_INT\_MASK
  - CHIP: LPC5411X A/D conversion driver, [24](#)
- ADC\_FLAGS\_OVERRUN\_MASK
  - CHIP: LPC5411X A/D conversion driver, [24](#)
- ADC\_FLAGS\_SEQA\_INT\_MASK
  - CHIP: LPC5411X A/D conversion driver, [24](#)
- ADC\_FLAGS\_SEQA\_OVERRUN\_MASK
  - CHIP: LPC5411X A/D conversion driver, [24](#)
- ADC\_FLAGS\_SEQB\_INT\_MASK
  - CHIP: LPC5411X A/D conversion driver, [24](#)
- ADC\_FLAGS\_SEQB\_OVERRUN\_MASK
  - CHIP: LPC5411X A/D conversion driver, [24](#)
- ADC\_FLAGS\_SEQN\_INT\_MASK
  - CHIP: LPC5411X A/D conversion driver, [24](#)
- ADC\_FLAGS\_SEQN\_OVERRUN\_MASK

- CHIP: LPC5411X A/D conversion driver, [24](#)
- ADC\_FLAGS\_THCMP\_INT\_MASK
  - CHIP: LPC5411X A/D conversion driver, [24](#)
- ADC\_FLAGS\_THCMP\_MASK
  - CHIP: LPC5411X A/D conversion driver, [25](#)
- ADC\_INTEN\_CMP\_CROSSTH
  - CHIP: LPC5411X A/D conversion driver, [25](#)
- ADC\_INTEN\_CMP\_DISBALE
  - CHIP: LPC5411X A/D conversion driver, [25](#)
- ADC\_INTEN\_CMP\_ENABLE
  - CHIP: LPC5411X A/D conversion driver, [25](#)
- ADC\_INTEN\_CMP\_MASK
  - CHIP: LPC5411X A/D conversion driver, [25](#)
- ADC\_INTEN\_CMP\_OUTSIDETH
  - CHIP: LPC5411X A/D conversion driver, [25](#)
- ADC\_INTEN\_OVERRUN\_ENABLE
  - CHIP: LPC5411X A/D conversion driver, [25](#)
- ADC\_INTEN\_SEQA\_ENABLE
  - CHIP: LPC5411X A/D conversion driver, [25](#)
- ADC\_INTEN\_SEQB\_ENABLE
  - CHIP: LPC5411X A/D conversion driver, [25](#)
- ADC\_INTEN\_SEQN\_ENABLE
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_INTEN\_THCMP\_CROSSING
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_INTEN\_THCMP\_DISABLE
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_INTEN\_THCMP\_OUTSIDE
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_INTEN\_THCMP\_T
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_MAX\_CHANNEL\_NUM
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_MAX\_SAMPLE\_RATE
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_SAMPLE\_RATE\_CONFIG\_MASK
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_SEQ\_CTRL\_BURST
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_SEQ\_CTRL\_CHANNEL\_EN
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_SEQ\_CTRL\_HWTRIG\_POLPOS
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_SEQ\_CTRL\_HWTRIG\_SYNCBYPASS
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_SEQ\_CTRL\_LOWPRIO
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_SEQ\_CTRL\_MODE\_EOS
  - CHIP: LPC5411X A/D conversion driver, [26](#)
- ADC\_SEQ\_CTRL\_SEQ\_ENA
  - CHIP: LPC5411X A/D conversion driver, [27](#)
- ADC\_SEQ\_CTRL\_SINGLESTEP
  - CHIP: LPC5411X A/D conversion driver, [27](#)
- ADC\_SEQ\_CTRL\_START
  - CHIP: LPC5411X A/D conversion driver, [27](#)
- ADC\_SEQ\_CTRL\_TRIGGER
  - CHIP: LPC5411X A/D conversion driver, [27](#)
- ADC\_SEQ\_GDAT\_CHAN\_BITPOS
  - CHIP: LPC5411X A/D conversion driver, [27](#)
- ADC\_SEQ\_GDAT\_CHAN\_MASK
  - CHIP: LPC5411X A/D conversion driver, [27](#)
- ADC\_SEQ\_GDAT\_DATAVALID
  - CHIP: LPC5411X A/D conversion driver, [27](#)
- ADC\_SEQ\_GDAT\_OVERRUN
  - CHIP: LPC5411X A/D conversion driver, [27](#)
- ADC\_SEQ\_GDAT\_RESULT\_BITPOS
  - CHIP: LPC5411X A/D conversion driver, [27](#)
- ADC\_SEQ\_GDAT\_RESULT\_MASK
  - CHIP: LPC5411X A/D conversion driver, [28](#)
- ADC\_SEQ\_GDAT\_THCMPCROSS\_BITPOS
  - CHIP: LPC5411X A/D conversion driver, [28](#)
- ADC\_SEQ\_GDAT\_THCMPCROSS\_MASK
  - CHIP: LPC5411X A/D conversion driver, [28](#)
- ADC\_SEQ\_GDAT\_THCMPRANGE\_BITPOS
  - CHIP: LPC5411X A/D conversion driver, [28](#)
- ADC\_SEQ\_GDAT\_THCMPRANGE\_MASK
  - CHIP: LPC5411X A/D conversion driver, [28](#)
- ADC\_SEQ\_IDX\_T
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_SEQA\_IDX
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADC\_SEQA\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- ADC\_SEQB\_IDX
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADC\_SEQB\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- ADC\_STARTUP\_ENABLE
  - CHIP: LPC5411X A/D conversion driver, [28](#)
- ADC\_STARTUP\_INIT
  - CHIP: LPC5411X A/D conversion driver, [28](#)
- ADC\_THCMP\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- ADC\_THR\_VAL\_MASK
  - CHIP: LPC5411X A/D conversion driver, [28](#)
- ADC\_THR\_VAL\_POS
  - CHIP: LPC5411X A/D conversion driver, [28](#)
- ADC\_THRSEL\_CHAN\_SEL\_THR1
  - CHIP: LPC5411X A/D conversion driver, [29](#)
- ADC\_TSAMP\_2CLK5
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADC\_TSAMP\_3CLK5
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADC\_TSAMP\_4CLK5
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADC\_TSAMP\_5CLK5
  - CHIP: LPC5411X A/D conversion driver, [30](#)

- ADC\_TSAMP\_6CLK5
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADC\_TSAMP\_7CLK5
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADC\_TSAMP\_8CLK5
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADC\_TSAMP\_9CLK5
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADC\_TSAMP\_T
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- ADCCLKDIV
  - LPC\_SYSCON\_T, [609](#)
- ADCCLKSEL
  - LPC\_SYSCON\_T, [609](#)
- ADDR
  - LPC\_USART\_T, [621](#)
- ADR
  - SCB\_Type, [634](#)
- AFSR
  - SCB\_Type, [634](#)
- AHBCLKCTRL
  - LPC\_SYSCON\_T, [609](#)
- AHBCLKCTRLCLR
  - LPC\_SYSCON\_T, [609](#)
- AHBCLKCTRLSET
  - LPC\_SYSCON\_T, [609](#)
- AHBCLKDIV
  - LPC\_SYSCON\_T, [609](#)
- AHBMATPRIO
  - LPC\_SYSCON\_T, [609](#)
- AIRCR
  - SCB\_Type, [634](#)
- ALIAS
  - cr\_startup\_lpc5411x-m0.c, [805](#)
  - cr\_startup\_lpc5411x.c, [807](#)
- ALIGN
  - LPC Public Types, [477](#)
- ALIGNED
  - packing.h, [745](#)
- APSR\_Type, [549](#)
  - \_reserved0, [549](#)
  - \_reserved1, [550](#)
  - b, [550](#)
  - C, [550](#)
  - GE, [550](#)
  - N, [550](#)
  - Q, [550](#)
  - V, [550](#)
  - w, [550](#)
  - Z, [550](#)
- ASYNCAPBCLKCTRL
  - LPC\_ASYNC\_SYSCON\_T, [573](#)
- ASYNCAPBCLKCTRLCLR
  - LPC\_ASYNC\_SYSCON\_T, [573](#)
- ASYNCAPBCLKCTRLSET
  - LPC\_ASYNC\_SYSCON\_T, [573](#)
- ASYNCAPBCLKSELA
  - LPC\_ASYNC\_SYSCON\_T, [573](#)
- ASYNCAPBCTRL
  - LPC\_SYSCON\_T, [609](#)
- ASYNCPRESETCTRLCLR
  - LPC\_ASYNC\_SYSCON\_T, [573](#)
- ASYNCPRESETCTRLSET
  - LPC\_ASYNC\_SYSCON\_T, [573](#)
- AUTOCGOR
  - LPC\_SYSCON\_T, [610](#)
- AYSNCPRESETCTRL
  - LPC\_ASYNC\_SYSCON\_T, [573](#)
- B
  - \_REQUEST\_TYPE, [632](#)
  - LPC\_GPIO\_T, [581](#)
- b
  - APSR\_Type, [550](#)
  - CONTROL\_Type, [552](#)
  - IPSR\_Type, [565](#), [566](#)
  - xPSR\_Type, [667](#)
- B3VAL
  - USBD\_Core, [537](#)
- bAlternateSetting
  - \_USB\_INTERFACE\_DESCRIPTOR, [660](#)
- bConfigurationValue
  - \_USB\_CONFIGURATION\_DESCRIPTOR, [651](#)
  - \_USB\_OTHER\_SPEED\_CONFIGURATION, [662](#)
- bDescriptorType
  - \_USB\_COMMON\_DESCRIPTOR, [650](#)
  - \_USB\_CONFIGURATION\_DESCRIPTOR, [651](#)
  - \_USB\_DEVICE\_DESCRIPTOR, [653](#)
  - \_USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, [655](#)
  - \_USB\_ENDPOINT\_DESCRIPTOR, [657](#)
  - \_USB\_IAD\_DESCRIPTOR, [659](#)
  - \_USB\_INTERFACE\_DESCRIPTOR, [660](#)
  - \_USB\_OTHER\_SPEED\_CONFIGURATION, [662](#)
  - \_USB\_STRING\_DESCRIPTOR, [664](#)
- bDeviceClass
  - \_USB\_DEVICE\_DESCRIPTOR, [653](#)
  - \_USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, [655](#)
- bDeviceProtocol
  - \_USB\_DEVICE\_DESCRIPTOR, [653](#)
  - \_USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, [655](#)
- bDeviceSubClass
  - \_USB\_DEVICE\_DESCRIPTOR, [654](#)
  - \_USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, [655](#)
- bEndpointAddress
  - \_USB\_ENDPOINT\_DESCRIPTOR, [657](#)
- BFAR
  - SCB\_Type, [634](#)
- bFirstInterface
  - \_USB\_IAD\_DESCRIPTOR, [659](#)
- bFunctionClass
  - \_USB\_IAD\_DESCRIPTOR, [659](#)
- bFunctionProtocol
  - \_USB\_IAD\_DESCRIPTOR, [659](#)
- bFunctionSubClass
  - \_USB\_IAD\_DESCRIPTOR, [659](#)
- bInterfaceClass
  - \_USB\_INTERFACE\_DESCRIPTOR, [660](#)

- bInterfaceCount
  - \_USB\_IAD\_DESCRIPTOR, 659
- bInterfaceNumber
  - \_USB\_INTERFACE\_DESCRIPTOR, 661
- bInterfaceProtocol
  - \_USB\_INTERFACE\_DESCRIPTOR, 661
- bInterfaceSubClass
  - \_USB\_INTERFACE\_DESCRIPTOR, 661
- bInterval
  - \_USB\_ENDPOINT\_DESCRIPTOR, 657
- BLOCKING
  - LPC Public Types, 480
- bLength
  - \_USB\_COMMON\_DESCRIPTOR, 650
  - \_USB\_CONFIGURATION\_DESCRIPTOR, 651
  - \_USB\_DEVICE\_DESCRIPTOR, 654
  - \_USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, 656
  - \_USB\_ENDPOINT\_DESCRIPTOR, 657
  - \_USB\_IAD\_DESCRIPTOR, 660
  - \_USB\_INTERFACE\_DESCRIPTOR, 661
  - \_USB\_OTHER\_SPEED\_CONFIGURATION, 662
  - \_USB\_STRING\_DESCRIPTOR, 664
- BM
  - \_REQUEST\_TYPE, 632
- BM\_T, 551
- bMaxPacketSize0
  - \_USB\_DEVICE\_DESCRIPTOR, 654
  - \_USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, 656
- bMaxPower
  - \_USB\_CONFIGURATION\_DESCRIPTOR, 652
  - \_USB\_OTHER\_SPEED\_CONFIGURATION, 662
- bNumConfigurations
  - \_USB\_DEVICE\_DESCRIPTOR, 654
  - \_USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, 656
- bNumEndpoints
  - \_USB\_INTERFACE\_DESCRIPTOR, 661
- bNumInterfaces
  - \_USB\_CONFIGURATION\_DESCRIPTOR, 652
  - \_USB\_OTHER\_SPEED\_CONFIGURATION, 662
- BOARD: Common board components used with board drivers, 15
- BOARD: LPC5411X boards, 16
- BODCTRL
  - LPC\_PMU\_T, 594
- BOOL\_16
  - LPC Public Types, 478
- BOOL\_32
  - LPC Public Types, 478
- BOOL\_8
  - LPC Public Types, 478
- BOX
  - LPC\_MBOX\_T, 588
- BRG
  - LPC\_USART\_T, 621
- bRequest
  - \_USB\_SETUP\_PACKET, 663
- bReserved
  - \_USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, 656
- bString
  - \_USB\_STRING\_DESCRIPTOR, 665
- BUSY
  - LPC\_DMA\_COMMON\_T, 577
- baud
  - UART\_BAUD\_T, 648
- bcdDevice
  - \_USB\_DEVICE\_DESCRIPTOR, 653
- bcdUSB
  - \_USB\_DEVICE\_DESCRIPTOR, 653
  - \_USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, 655
- bmAttributes
  - \_USB\_CONFIGURATION\_DESCRIPTOR, 651
  - \_USB\_ENDPOINT\_DESCRIPTOR, 657
  - \_USB\_OTHER\_SPEED\_CONFIGURATION, 662
- bmRequestType
  - \_USB\_SETUP\_PACKET, 663
- Board specific drivers and support functions, 17
- boards\_5411x.dox, 809
- Bool
  - LPC Public Types, 479
- BusFault\_Handler
  - cr\_startup\_lpc5411x.c, 807
- BusFault\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, 430
- C
  - APSR\_Type, 550
  - xPSR\_Type, 667
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/adc\_5411x.h, 669
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/chip.h, 672
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/clock\_5411x.h, 675
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/cmsis.h, 679
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/cmsis\_5411x.h, 679
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/cmsis\_5411x\_m0.h, 680
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm0plus.h, 681
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm4.h, 685
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cm4\_simd.h, 695
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cmFunc.h, 696
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/core\_cmInstr.h, 696
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/cpuctrl\_5411x.h, 696
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/crc\_5411x.h, 697
  - C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/dma\_5411x.h, 698



- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/dma\_↔  
service\_5411x.h, [701](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/dmic\_↔  
\_5411x.h, [702](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/error.h, [703](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/flexcomm\_↔  
\_5411x.h, [710](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/fpu\_↔  
init.h, [711](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/gpio\_↔  
\_5411x.h, [711](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/gpiogroup\_↔  
\_5411x.h, [712](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/i2c\_↔  
common\_5411x.h, [713](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/i2cm\_↔  
\_5411x.h, [717](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/i2cs\_↔  
\_5411x.h, [718](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/i2s\_↔  
\_5411x.h, [720](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/iap.h, [738](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/inmux\_↔  
\_5411x.h, [739](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/iocon\_↔  
\_5411x.h, [740](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/lpc\_↔  
assert.h, [741](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/lpc\_↔  
types.h, [741](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/mailbox\_↔  
\_5411x.h, [742](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/mrt\_↔  
\_5411x.h, [743](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/packing\_↔  
h, [745](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/pinint\_↔  
\_5411x.h, [745](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/pintable\_↔  
\_5411x.h, [747](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/pll\_↔  
\_5411x.h, [747](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/pmu\_↔  
\_5411x.h, [749](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/power\_↔  
\_lib\_5411x.h, [749](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/ring\_↔  
buffer.h, [750](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/romapi\_↔  
\_5411x.h, [751](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/rtc\_↔  
\_5411x.h, [751](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/rtc\_↔  
ut.h, [752](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/sct\_↔  
\_5411x.h, [753](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/sct\_↔  
pwm\_5411x.h, [756](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/spi\_↔  
common\_5411x.h, [756](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/spim\_↔  
\_5411x.h, [760](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/spis\_↔  
\_5411x.h, [761](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/stopwatch\_↔  
h, [762](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/syscon\_↔  
\_5411x.h, [762](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/timer\_↔  
\_5411x.h, [765](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/uart\_↔  
\_5411x.h, [767](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/usbd.h, [771](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/utick\_↔  
\_5411x.h, [774](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/inc/wwdt\_↔  
\_5411x.h, [774](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/adc\_↔  
\_5411x.c, [775](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/chip\_↔  
\_5411x.c, [776](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/clock\_↔  
\_5411x.c, [776](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/dma\_↔  
\_5411x.c, [778](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/dma\_↔  
service\_5411x.c, [778](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/dmic\_↔  
\_5411x.c, [779](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/flexcomm\_↔  
\_5411x.c, [780](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/fpu\_↔  
init.c, [780](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/i2cm\_↔  
\_5411x.c, [780](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/i2cs\_↔  
\_5411x.c, [780](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/i2s\_↔  
\_5411x.c, [781](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/iap.c, [783](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/pll\_↔  
\_5411x.c, [783](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/ring\_↔  
buffer.c, [796](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/rtc\_↔  
ut.c, [796](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/sct\_↔  
\_5411x.c, [798](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/sct\_↔  
pwm\_5411x.c, [798](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/spi\_↔  
common\_5411x.c, [799](#)

- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/spim\_5411x.c, [799](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/spis\_5411x.c, [800](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/stopwatch\_5411x.c, [801](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/syscon\_5411x.c, [802](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/sysinit\_5411x.c, [802](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/timer\_5411x.c, [803](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/src/uart\_5411x.c, [803](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/cr\_startup\_lpc5411x-m0.c, [805](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/cr\_startup\_lpc5411x.c, [806](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/crp.c, [808](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/mtb.c, [808](#)
- C:/Jenkins/LPC5411x/lpc5411x/chip\_5411x/startup/sysinit.c, [808](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/community\_support.dox, [808](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/copyright.dox, [809](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/installation\_insts.dox, [809](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/toolchain.dox, [809](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/common/version\_history.dox, [809](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/chip\_5411x.dox, [809](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/irq\_vector\_names.dox, [809](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/lpcopen\_layout.dox, [809](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/main\_layout.dox, [809](#)
- C:/Jenkins/LPC5411x/lpcdocs/docs/doxygen/dox\_lpc5411x/chip\_5411x/multicore.dox, [809](#)
- CALIB
  - SysTick\_Type, [643](#)
- CALIBR
  - LPC\_ADC\_T, [571](#)
- CAP
  - LPC\_SCT\_T, [599](#)
- CAPCTRL
  - LPC\_SCT\_T, [599](#)
- CCR
  - LPC\_TIMER\_T, [619](#)
  - SCB\_Type, [635](#)
- CFG
  - LPC\_DMA\_CHANNEL\_T, [575](#)
  - LPC\_SPI\_T, [604](#)
  - LPC\_USART\_T, [621](#)
  - CFG1
    - LPC\_I2S\_T, [584](#)
  - CFG2
    - LPC\_I2S\_T, [584](#)
  - CFSR
    - SCB\_Type, [635](#)
  - CHAN\_THRSEL
    - LPC\_ADC\_T, [571](#)
  - CHANEN
    - CHIP: LPC5411X DMIC driver, [143](#)
  - CHANNEL
    - CHIP: LPC5411X DMIC driver, [143](#)
    - LPC\_MRT\_T, [591](#)
  - CHAR
    - LPC Public Types, [478](#)
  - CHIP: LPC5411X A/D conversion driver, [18](#)
    - ADC\_CALIB, [21](#)
    - ADC\_CALREQD, [21](#)
    - ADC\_CR\_ASYNC\_MODE, [21](#)
    - ADC\_CR\_BITACC, [21](#)
    - ADC\_CR\_BYPASS, [21](#)
    - ADC\_CR\_CALMODEBIT, [21](#)
    - ADC\_CR\_CLKDIV, [21](#)
    - ADC\_CR\_CLKDIV\_BITPOS, [22](#)
    - ADC\_CR\_CLKDIV\_MASK, [22](#)
    - ADC\_CR\_LPWRMODEBIT, [22](#)
    - ADC\_CR\_RESOL, [22](#)
    - ADC\_CR\_TSAMP, [22](#)
    - ADC\_DR\_CHAN\_BITPOS, [22](#)
    - ADC\_DR\_CHAN\_MASK, [22](#)
    - ADC\_DR\_CHANNEL, [22](#)
    - ADC\_DR\_DATAVALID, [22](#)
    - ADC\_DR\_DONE, [23](#)
    - ADC\_DR\_OVERRUN, [23](#)
    - ADC\_DR\_RESULT, [23](#)
    - ADC\_DR\_RESULT\_BITPOS, [23](#)
    - ADC\_DR\_THCMPCROSS, [23](#)
    - ADC\_DR\_THCMPCROSS\_BITPOS, [23](#)
    - ADC\_DR\_THCMPCROSS\_DOWNWARD, [29](#)
    - ADC\_DR\_THCMPCROSS\_MASK, [23](#)
    - ADC\_DR\_THCMPCROSS\_NOCROSS, [29](#)
    - ADC\_DR\_THCMPCROSS\_RESERVED, [29](#)
    - ADC\_DR\_THCMPCROSS\_T, [29](#)
    - ADC\_DR\_THCMPCROSS\_UPWARD, [29](#)
    - ADC\_DR\_THCMPRANGE, [23](#)
    - ADC\_DR\_THCMPRANGE\_ABOVE, [29](#)
    - ADC\_DR\_THCMPRANGE\_BELOW, [29](#)
    - ADC\_DR\_THCMPRANGE\_BITPOS, [23](#)
    - ADC\_DR\_THCMPRANGE\_INRANGE, [29](#)
    - ADC\_DR\_THCMPRANGE\_MASK, [23](#)
    - ADC\_DR\_THCMPRANGE\_RESERVED, [29](#)
    - ADC\_DR\_THCMPRANGE\_T, [29](#)
    - ADC\_FLAGS\_OVERRUN\_INT\_MASK, [24](#)
    - ADC\_FLAGS\_OVERRUN\_MASK, [24](#)
    - ADC\_FLAGS\_SEQA\_INT\_MASK, [24](#)



- ADC\_FLAGS\_SEQA\_OVRRUN\_MASK, 24
- ADC\_FLAGS\_SEQB\_INT\_MASK, 24
- ADC\_FLAGS\_SEQB\_OVRRUN\_MASK, 24
- ADC\_FLAGS\_SEQN\_INT\_MASK, 24
- ADC\_FLAGS\_SEQN\_OVRRUN\_MASK, 24
- ADC\_FLAGS\_THCMP\_INT\_MASK, 24
- ADC\_FLAGS\_THCMP\_MASK, 25
- ADC\_INTEN\_CMP\_CROSSTH, 25
- ADC\_INTEN\_CMP\_DISBALE, 25
- ADC\_INTEN\_CMP\_ENABLE, 25
- ADC\_INTEN\_CMP\_MASK, 25
- ADC\_INTEN\_CMP\_OUTSIDETH, 25
- ADC\_INTEN\_OVRRUN\_ENABLE, 25
- ADC\_INTEN\_SEQA\_ENABLE, 25
- ADC\_INTEN\_SEQB\_ENABLE, 25
- ADC\_INTEN\_SEQN\_ENABLE, 26
- ADC\_INTEN\_THCMP\_CROSSING, 29
- ADC\_INTEN\_THCMP\_DISABLE, 29
- ADC\_INTEN\_THCMP\_OUTSIDE, 29
- ADC\_INTEN\_THCMP\_T, 29
- ADC\_MAX\_CHANNEL\_NUM, 26
- ADC\_MAX\_SAMPLE\_RATE, 26
- ADC\_SAMPLE\_RATE\_CONFIG\_MASK, 26
- ADC\_SEQ\_CTRL\_BURST, 26
- ADC\_SEQ\_CTRL\_CHANNEL\_EN, 26
- ADC\_SEQ\_CTRL\_HWTRIG\_POLPOS, 26
- ADC\_SEQ\_CTRL\_HWTRIG\_SYNCBYPASS, 26
- ADC\_SEQ\_CTRL\_LOWPRIO, 26
- ADC\_SEQ\_CTRL\_MODE\_EOS, 26
- ADC\_SEQ\_CTRL\_SEQ\_ENA, 27
- ADC\_SEQ\_CTRL\_SINGLESTEP, 27
- ADC\_SEQ\_CTRL\_START, 27
- ADC\_SEQ\_CTRL\_TRIGGER, 27
- ADC\_SEQ\_GDAT\_CHAN\_BITPOS, 27
- ADC\_SEQ\_GDAT\_CHAN\_MASK, 27
- ADC\_SEQ\_GDAT\_DATAVALID, 27
- ADC\_SEQ\_GDAT\_OVERRUN, 27
- ADC\_SEQ\_GDAT\_RESULT\_BITPOS, 27
- ADC\_SEQ\_GDAT\_RESULT\_MASK, 28
- ADC\_SEQ\_GDAT\_THCMPCROSS\_BITPOS, 28
- ADC\_SEQ\_GDAT\_THCMPCROSS\_MASK, 28
- ADC\_SEQ\_GDAT\_THCMPRANGE\_BITPOS, 28
- ADC\_SEQ\_GDAT\_THCMPRANGE\_MASK, 28
- ADC\_SEQ\_IDX\_T, 29
- ADC\_SEQA\_IDX, 30
- ADC\_SEQB\_IDX, 30
- ADC\_STARTUP\_ENABLE, 28
- ADC\_STARTUP\_INIT, 28
- ADC\_THR\_VAL\_MASK, 28
- ADC\_THR\_VAL\_POS, 28
- ADC\_THRSEL\_CHAN\_SEL\_THR1, 29
- ADC\_TSAMP\_2CLK5, 30
- ADC\_TSAMP\_3CLK5, 30
- ADC\_TSAMP\_4CLK5, 30
- ADC\_TSAMP\_5CLK5, 30
- ADC\_TSAMP\_6CLK5, 30
- ADC\_TSAMP\_7CLK5, 30
- ADC\_TSAMP\_8CLK5, 30
- ADC\_TSAMP\_9CLK5, 30
- ADC\_TSAMP\_T, 30
- Chip\_ADC\_Calibration, 30
- Chip\_ADC\_ClearFlags, 30
- Chip\_ADC\_ClearSequencerBits, 31
- Chip\_ADC\_ClearTHRSELBits, 31
- Chip\_ADC\_DeInit, 31
- Chip\_ADC\_DisableInt, 32
- Chip\_ADC\_DisableSequencer, 32
- Chip\_ADC\_EnableInt, 32
- Chip\_ADC\_EnableSequencer, 33
- Chip\_ADC\_GetDataReg, 33
- Chip\_ADC\_GetDivider, 33
- Chip\_ADC\_GetFlags, 35
- Chip\_ADC\_GetGlobalDataReg, 35
- Chip\_ADC\_GetSequencerCtrl, 36
- Chip\_ADC\_Init, 36
- Chip\_ADC\_SelectTH0Channels, 37
- Chip\_ADC\_SelectTH1Channels, 37
- Chip\_ADC\_SelectTempSensorInput, 36
- Chip\_ADC\_SetClockRate, 37
- Chip\_ADC\_SetDivider, 38
- Chip\_ADC\_SetSequencerBits, 38
- Chip\_ADC\_SetTHRSELBits, 40
- Chip\_ADC\_SetThrHighValue, 40
- Chip\_ADC\_SetThrLowValue, 40
- Chip\_ADC\_SetThresholdInt, 38
- Chip\_ADC\_SetupSequencer, 40
- Chip\_ADC\_StartBurstSequencer, 42
- Chip\_ADC\_StartSequencer, 42
- Chip\_ADC\_StopBurstSequencer, 43
- CHIP: CHIP\_LPC5411X family IRQ vector names and mapped NVIC IRQ numbers, 44
- CHIP: Common Chip ISP/IAP commands and return codes, 46
- Chip\_IAP\_BlankCheckSector, 50
- Chip\_IAP\_Compare, 51
- Chip\_IAP\_CopyRamToFlash, 51
- Chip\_IAP\_ErasePage, 51
- Chip\_IAP\_EraseSector, 52
- Chip\_IAP\_PreSectorForReadWrite, 52
- Chip\_IAP\_ReadBootCode, 52
- Chip\_IAP\_ReadPID, 53
- Chip\_IAP\_ReadUID, 53
- Chip\_IAP\_ReinvokeISP, 53
- IAP\_ADDR\_ERROR, 47
- IAP\_ADDR\_NOT\_MAPPED, 47
- IAP\_BLANK\_CHECK\_SECTOR\_CMD, 47
- IAP\_BUSY, 47
- IAP\_CMD\_LOCKED, 47
- IAP\_CMD\_SUCCESS, 47
- IAP\_COMPARE\_CMD, 47
- IAP\_COMPARE\_ERROR, 47
- IAP\_COUNT\_ERROR, 48
- IAP\_CRP\_ENABLED, 48
- IAP\_DST\_ADDR\_ERROR, 48
- IAP\_DST\_ADDR\_NOT\_MAPPED, 48
- IAP\_EEPROM\_READ, 48

- IAP\_EEPROM\_WRITE, [48](#)
- IAP\_ENTRY\_T, [50](#)
- IAP\_ERASE\_PAGE\_CMD, [48](#)
- IAP\_ERSECTOR\_CMD, [48](#)
- IAP\_INVALID\_BAUD\_RATE, [48](#)
- IAP\_INVALID\_CODE, [49](#)
- IAP\_INVALID\_COMMAND, [49](#)
- IAP\_INVALID\_SECTOR, [49](#)
- IAP\_INVALID\_STOP\_BIT, [49](#)
- IAP\_PARAM\_ERROR, [49](#)
- IAP\_PREWRITE\_CMD, [49](#)
- IAP\_READ\_BOOT\_CODE\_CMD, [49](#)
- IAP\_READ\_UID\_CMD, [49](#)
- IAP\_REINVOKE\_ISP\_CMD, [49](#)
- IAP\_REPID\_CMD, [50](#)
- IAP\_SECTOR\_NOT\_BLANK, [50](#)
- IAP\_SECTOR\_NOT\_PREPARED, [50](#)
- IAP\_SRC\_ADDR\_ERROR, [50](#)
- IAP\_SRC\_ADDR\_NOT\_MAPPED, [50](#)
- IAP\_WRISECTOR\_CMD, [50](#)
- CHIP: FPU initialization, [54](#)
- fpulnit, [54](#)
- CHIP: LPC Common Types, [55](#)
- CHIP: LPC5410x family CMSIS include files, [56](#)
- CHIP: LPC5411X 32-bit Timer driver, [57](#)
- Chip\_TIMER\_CaptureDisableInt, [61](#)
- Chip\_TIMER\_CaptureEnableInt, [61](#)
- Chip\_TIMER\_CaptureFallingEdgeDisable, [61](#)
- Chip\_TIMER\_CaptureFallingEdgeEnable, [62](#)
- Chip\_TIMER\_CapturePending, [62](#)
- Chip\_TIMER\_CaptureRisingEdgeDisable, [62](#)
- Chip\_TIMER\_CaptureRisingEdgeEnable, [62](#)
- Chip\_TIMER\_ClearCapture, [64](#)
- Chip\_TIMER\_ClearMatch, [64](#)
- Chip\_TIMER\_DeInit, [64](#)
- Chip\_TIMER\_Disable, [65](#)
- Chip\_TIMER\_Enable, [65](#)
- Chip\_TIMER\_ExtMatchControlSet, [65](#)
- Chip\_TIMER\_Init, [66](#)
- Chip\_TIMER\_MatchDisableInt, [66](#)
- Chip\_TIMER\_MatchEnableInt, [66](#)
- Chip\_TIMER\_MatchPending, [67](#)
- Chip\_TIMER\_PrescaleSet, [67](#)
- Chip\_TIMER\_ReadCapture, [67](#)
- Chip\_TIMER\_ReadCount, [68](#)
- Chip\_TIMER\_ReadPrescale, [68](#)
- Chip\_TIMER\_Reset, [68](#)
- Chip\_TIMER\_ResetOnMatchDisable, [69](#)
- Chip\_TIMER\_ResetOnMatchEnable, [69](#)
- Chip\_TIMER\_SetMatch, [69](#)
- Chip\_TIMER\_StopOnMatchDisable, [69](#)
- Chip\_TIMER\_StopOnMatchEnable, [71](#)
- Chip\_TIMER\_TIMER\_SetCountClockSrc, [71](#)
- TIMER\_CAP\_FALLING, [59](#)
- TIMER\_CAP\_INT, [59](#)
- TIMER\_CAP\_RISING, [59](#)
- TIMER\_CAP\_SRC\_STATE\_T, [60](#)
- TIMER\_CAPSRC\_BOTH\_CAPN, [60](#)
- TIMER\_CAPSRC\_FALLING\_CAPN, [60](#)
- TIMER\_CAPSRC\_RISING\_CAPN, [60](#)
- TIMER\_CAPSRC\_RISING\_PCLK, [60](#)
- TIMER\_CCR\_MASK, [59](#)
- TIMER\_CTCR\_MASK, [59](#)
- TIMER\_CTRL\_MASK, [59](#)
- TIMER\_EMR\_MASK, [59](#)
- TIMER\_ENABLE, [59](#)
- TIMER\_EXTMATCH\_CLEAR, [61](#)
- TIMER\_EXTMATCH\_DO\_NOTHING, [61](#)
- TIMER\_EXTMATCH\_SET, [61](#)
- TIMER\_EXTMATCH\_TOGGLE, [61](#)
- TIMER\_INT\_ON\_CAP, [59](#)
- TIMER\_INT\_ON\_MATCH, [59](#)
- TIMER\_IR\_CLR, [60](#)
- TIMER\_MATCH\_INT, [60](#)
- TIMER\_MCR\_MASK, [60](#)
- TIMER\_PIN\_MATCH\_STATE\_T, [60](#)
- TIMER\_RESET, [60](#)
- TIMER\_RESET\_ON\_MATCH, [60](#)
- TIMER\_STOP\_ON\_MATCH, [60](#)
- CHIP: LPC5411X CPU multi-core support driver, [72](#)
- CORESELECT\_M0PLUS, [72](#)
- CORESELECT\_M4, [72](#)
- CORESELECT\_T, [72](#)
- Chip\_CPU\_CM0Boot, [72](#)
- Chip\_CPU\_CM4Boot, [73](#)
- Chip\_CPU\_IsM4Core, [73](#)
- Chip\_CPU\_IsMasterCore, [73](#)
- Chip\_CPU\_SelectMasterCore, [73](#)
- CHIP: LPC5411X Clock Driver, [74](#)
- CHIP\_ASYNC\_SYSCON\_SRC\_T, [78](#)
- CHIP\_SYSCON\_ADCCCLKSELSRC\_T, [78](#)
- CHIP\_SYSCON\_CLKOUTSRC\_T, [78](#)
- CHIP\_SYSCON\_CLOCK\_T, [78](#)
- CHIP\_SYSCON\_FLEXCOMMCLKSELSRC\_T, [79](#)
- CHIP\_SYSCON\_FRGCLKSRC\_T, [80](#)
- CHIP\_SYSCON\_MAIN\_A\_CLKSRC\_T, [80](#)
- CHIP\_SYSCON\_MAIN\_B\_CLKSRC\_T, [80](#)
- CHIP\_SYSCON\_MAINCLKSRC\_T, [80](#)
- CHIP\_SYSCON\_MCLKSRC\_T, [81](#)
- CHIP\_SYSCON\_USBCLKSRC\_T, [81](#)
- Chip\_Clock\_DisablePeriphClock, [82](#)
- Chip\_Clock\_DisableRTCOsc, [82](#)
- Chip\_Clock\_EnablePeriphClock, [82](#)
- Chip\_Clock\_EnableRTCOsc, [84](#)
- Chip\_Clock\_GetADCClockDiv, [84](#)
- Chip\_Clock\_GetADCClockRate, [84](#)
- Chip\_Clock\_GetADCClockSource, [84](#)
- Chip\_Clock\_GetAsyncSyscon\_ClockRate, [84](#)
- Chip\_Clock\_GetAsyncSysconClockSource, [85](#)
- Chip\_Clock\_GetCLKOUTDiv, [85](#)
- Chip\_Clock\_GetCLKOUTSource, [85](#)
- Chip\_Clock\_GetExtClockInRate, [85](#)
- Chip\_Clock\_GetFLEXCOMMClockRate, [85](#)
- Chip\_Clock\_GetFLEXCOMMClockSource, [86](#)
- Chip\_Clock\_GetFRGClockRate, [86](#)
- Chip\_Clock\_GetFRGClockSource, [86](#)

- Chip\_Clock\_GetFRGInClockRate, 86
- Chip\_Clock\_GetFROHFRate, 86
- Chip\_Clock\_GetIntOscRate, 77
- Chip\_Clock\_GetMCLKDir, 88
- Chip\_Clock\_GetMCLKDiv, 88
- Chip\_Clock\_GetMCLKSource, 88
- Chip\_Clock\_GetMain\_A\_ClockRate, 87
- Chip\_Clock\_GetMain\_A\_ClockSource, 87
- Chip\_Clock\_GetMain\_B\_ClockRate, 87
- Chip\_Clock\_GetMain\_B\_ClockSource, 87
- Chip\_Clock\_GetMainClockRate, 87
- Chip\_Clock\_GetMainClockSource, 87
- Chip\_Clock\_GetRTCOsc, 88
- Chip\_Clock\_GetRTCOscRate, 88
- Chip\_Clock\_GetSysClockDiv, 88
- Chip\_Clock\_GetSysTickClockDiv, 89
- Chip\_Clock\_GetSysTickClockRate, 89
- Chip\_Clock\_GetSystemClockRate, 89
- Chip\_Clock\_GetUSBClockDiv, 89
- Chip\_Clock\_GetUSBClockSource, 89
- Chip\_Clock\_GetWDTOSCRate, 90
- Chip\_Clock\_SetADCClockDiv, 90
- Chip\_Clock\_SetADCClockSource, 90
- Chip\_Clock\_SetAsyncSysconClockSource, 90
- Chip\_Clock\_SetCLKOUTSource, 91
- Chip\_Clock\_SetFLEXCOMMClockSource, 91
- Chip\_Clock\_SetFRGClockRate, 91
- Chip\_Clock\_SetFRGClockSource, 91
- Chip\_Clock\_SetMCLKClockSource, 93
- Chip\_Clock\_SetMCLKDir, 93
- Chip\_Clock\_SetMCLKDirInput, 93
- Chip\_Clock\_SetMCLKDirOutput, 93
- Chip\_Clock\_SetMain\_A\_ClockSource, 92
- Chip\_Clock\_SetMain\_B\_ClockSource, 92
- Chip\_Clock\_SetMainClockSource, 92
- Chip\_Clock\_SetSysClockDiv, 93
- Chip\_Clock\_SetSysTickClockDiv, 95
- Chip\_Clock\_SetUSBClockSource, 95
- Chip\_Clock\_SetWDTOSCRate, 95
- SYSCON\_ADCCLKSELSRC\_FROHF, 78
- SYSCON\_ADCCLKSELSRC\_MAINCLK, 78
- SYSCON\_ADCCLKSELSRC\_SYSPLLOUT, 78
- SYSCON\_ASYNC\_FRO12MHZ, 78
- SYSCON\_ASYNC\_MAINCLK, 78
- SYSCON\_CLKOUTSRC\_CLKIN, 78
- SYSCON\_CLKOUTSRC\_DISABLED, 78
- SYSCON\_CLKOUTSRC\_FRO12MHZ, 78
- SYSCON\_CLKOUTSRC\_FROHF, 78
- SYSCON\_CLKOUTSRC\_MAINCLK, 78
- SYSCON\_CLKOUTSRC\_PLL, 78
- SYSCON\_CLKOUTSRC\_RTC, 78
- SYSCON\_CLKOUTSRC\_WDTOSC, 78
- SYSCON\_CLOCK\_ADC0, 79
- SYSCON\_CLOCK\_CRC, 79
- SYSCON\_CLOCK\_DMA, 79
- SYSCON\_CLOCK\_DMIC, 79
- SYSCON\_CLOCK\_FLASH, 79
- SYSCON\_CLOCK\_FLEXCOMM0, 79
- SYSCON\_CLOCK\_FLEXCOMM1, 79
- SYSCON\_CLOCK\_FLEXCOMM2, 79
- SYSCON\_CLOCK\_FLEXCOMM3, 79
- SYSCON\_CLOCK\_FLEXCOMM4, 79
- SYSCON\_CLOCK\_FLEXCOMM5, 79
- SYSCON\_CLOCK\_FLEXCOMM6, 79
- SYSCON\_CLOCK\_FLEXCOMM7, 79
- SYSCON\_CLOCK\_FMC, 79
- SYSCON\_CLOCK\_GINT, 79
- SYSCON\_CLOCK\_GPIO0, 79
- SYSCON\_CLOCK\_GPIO1, 79
- SYSCON\_CLOCK\_INPUTMUX, 79
- SYSCON\_CLOCK\_IOCON, 79
- SYSCON\_CLOCK\_MAILBOX, 79
- SYSCON\_CLOCK\_MRT, 79
- SYSCON\_CLOCK\_PINT, 79
- SYSCON\_CLOCK\_ROM, 79
- SYSCON\_CLOCK\_RTC, 79
- SYSCON\_CLOCK\_SCT0, 79
- SYSCON\_CLOCK\_SPIFI, 79
- SYSCON\_CLOCK\_SRAM1, 79
- SYSCON\_CLOCK\_SRAM2, 79
- SYSCON\_CLOCK\_SRAMX, 79
- SYSCON\_CLOCK\_TIMER0, 79
- SYSCON\_CLOCK\_TIMER1, 79
- SYSCON\_CLOCK\_TIMER2, 79
- SYSCON\_CLOCK\_TIMER3, 79
- SYSCON\_CLOCK\_TIMER4, 79
- SYSCON\_CLOCK\_USB, 79
- SYSCON\_CLOCK\_UTICK, 79
- SYSCON\_CLOCK\_WWDT, 79
- SYSCON\_FLEXCOMMCLKSELSRC\_FRG, 80
- SYSCON\_FLEXCOMMCLKSELSRC\_FRO12MHz, 80
- SYSCON\_FLEXCOMMCLKSELSRC\_FROHF, 80
- SYSCON\_FLEXCOMMCLKSELSRC\_MCLK, 80
- SYSCON\_FLEXCOMMCLKSELSRC\_NONE, 80
- SYSCON\_FLEXCOMMCLKSELSRC\_PLL, 80
- SYSCON\_FRGCLKSRC\_FRO12MHZ, 80
- SYSCON\_FRGCLKSRC\_FROHF, 80
- SYSCON\_FRGCLKSRC\_MAINCLK, 80
- SYSCON\_FRGCLKSRC\_NONE, 80
- SYSCON\_FRGCLKSRC\_PLL, 80
- SYSCON\_FRO12MHZ\_FREQ, 77
- SYSCON\_FRO48MHZ\_FREQ, 77
- SYSCON\_FRO96MHZ\_FREQ, 77
- SYSCON\_MAIN\_A\_CLKSRC\_FRO12MHZ, 80
- SYSCON\_MAIN\_A\_CLKSRCA\_CLKIN, 80
- SYSCON\_MAIN\_A\_CLKSRCA\_FROHF, 80
- SYSCON\_MAIN\_A\_CLKSRCA\_WDTOSC, 80
- SYSCON\_MAIN\_B\_CLKSRC\_MAINCLKSELA, 80
- SYSCON\_MAIN\_B\_CLKSRC\_PLL, 80
- SYSCON\_MAIN\_B\_CLKSRC\_RTC, 80
- SYSCON\_MAINCLKSRC\_CLKIN, 81
- SYSCON\_MAINCLKSRC\_FRO12MHZ, 81
- SYSCON\_MAINCLKSRC\_FROHF, 81
- SYSCON\_MAINCLKSRC\_PLLOUT, 81
- SYSCON\_MAINCLKSRC\_RTC, 81

- SYSCON\_MAINCLKSRC\_WDTOSC, [81](#)
- SYSCON\_MCLKSRC\_DISABLED, [81](#)
- SYSCON\_MCLKSRC\_FROHF, [81](#)
- SYSCON\_MCLKSRC\_MCLKIN, [81](#)
- SYSCON\_MCLKSRC\_PLL, [81](#)
- SYSCON\_RTC\_FREQ, [77](#)
- SYSCON\_USBCCLKSRC\_DISABLED, [81](#)
- SYSCON\_USBCCLKSRC\_FROHF, [81](#)
- SYSCON\_USBCCLKSRC\_PLL, [81](#)
- SYSCON\_WDTOSC\_FREQ, [78](#)
- WDT\_FREQ\_1000000, [81](#)
- WDT\_FREQ\_1200000, [81](#)
- WDT\_FREQ\_1300000, [81](#)
- WDT\_FREQ\_1400000, [82](#)
- WDT\_FREQ\_1500000, [82](#)
- WDT\_FREQ\_1600000, [82](#)
- WDT\_FREQ\_1700000, [82](#)
- WDT\_FREQ\_1800000, [82](#)
- WDT\_FREQ\_1900000, [82](#)
- WDT\_FREQ\_2000000, [82](#)
- WDT\_FREQ\_2050000, [82](#)
- WDT\_FREQ\_2100000, [82](#)
- WDT\_FREQ\_2200000, [82](#)
- WDT\_FREQ\_2250000, [82](#)
- WDT\_FREQ\_2300000, [82](#)
- WDT\_FREQ\_2400000, [82](#)
- WDT\_FREQ\_2450000, [82](#)
- WDT\_FREQ\_2500000, [82](#)
- WDT\_FREQ\_2600000, [82](#)
- WDT\_FREQ\_2650000, [82](#)
- WDT\_FREQ\_2700000, [82](#)
- WDT\_FREQ\_2800000, [82](#)
- WDT\_FREQ\_2850000, [82](#)
- WDT\_FREQ\_2900000, [82](#)
- WDT\_FREQ\_2950000, [82](#)
- WDT\_FREQ\_3000000, [82](#)
- WDT\_FREQ\_3050000, [82](#)
- WDT\_FREQ\_400000, [81](#)
- WDT\_FREQ\_600000, [81](#)
- WDT\_FREQ\_750000, [81](#)
- WDT\_FREQ\_900000, [81](#)
- WDT\_FREQ\_RESERVED, [81](#)
- WDT\_OSC\_FREQ\_T, [81](#)
- CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [97](#)
  - CRC\_MODE\_POLY\_BITMASK, [98](#)
  - CRC\_MODE\_POLY\_CCITT, [98](#)
  - CRC\_MODE\_POLY\_CRC16, [98](#)
  - CRC\_MODE\_POLY\_CRC32, [98](#)
  - CRC\_MODE\_SUM\_BIT\_RVS, [98](#)
  - CRC\_MODE\_SUM\_BITMASK, [98](#)
  - CRC\_MODE\_SUM\_CMPL, [98](#)
  - CRC\_MODE\_WRDATA\_BIT\_RVS, [99](#)
  - CRC\_MODE\_WRDATA\_BITMASK, [99](#)
  - CRC\_MODE\_WRDATA\_CMPL, [99](#)
  - CRC\_POLY\_CCITT, [99](#)
  - CRC\_POLY\_CRC16, [99](#)
  - CRC\_POLY\_CRC32, [99](#)
  - CRC\_POLY\_LAST, [100](#)
  - CRC\_POLY\_T, [99](#)
  - CRC\_SEED\_CCITT, [99](#)
  - CRC\_SEED\_CRC16, [99](#)
  - CRC\_SEED\_CRC32, [99](#)
  - Chip\_CRC\_CRC16, [100](#)
  - Chip\_CRC\_CRC32, [100](#)
  - Chip\_CRC\_CRC8, [100](#)
  - Chip\_CRC\_Deinit, [100](#)
  - Chip\_CRC\_GetMode, [101](#)
  - Chip\_CRC\_GetSeed, [101](#)
  - Chip\_CRC\_Init, [101](#)
  - Chip\_CRC\_SetMode, [101](#)
  - Chip\_CRC\_SetPoly, [102](#)
  - Chip\_CRC\_SetSeed, [102](#)
  - Chip\_CRC\_Sum, [102](#)
  - Chip\_CRC\_UseCCITT, [103](#)
  - Chip\_CRC\_UseCRC16, [103](#)
  - Chip\_CRC\_UseCRC32, [103](#)
  - Chip\_CRC\_UseDefaultConfig, [103](#)
  - Chip\_CRC\_Write16, [104](#)
  - Chip\_CRC\_Write32, [104](#)
  - Chip\_CRC\_Write8, [104](#)
  - MODE\_CFG\_CCITT, [99](#)
  - MODE\_CFG\_CRC16, [99](#)
  - MODE\_CFG\_CRC32, [99](#)
- CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [105](#)
  - Chip\_DMA\_ClearTranBits, [105](#)
  - Chip\_DMA\_GetChannelStatus, [106](#)
  - Chip\_DMA\_SWTriggerChannel, [110](#)
  - Chip\_DMA\_SetChannelInvalid, [106](#)
  - Chip\_DMA\_SetChannelValid, [106](#)
  - Chip\_DMA\_SetTranBits, [106](#)
  - Chip\_DMA\_SetupChannelConfig, [108](#)
  - Chip\_DMA\_SetupChannelTransfer, [108](#)
  - Chip\_DMA\_SetupChannelTransferSize, [109](#)
  - Chip\_DMA\_SetupTranChannel, [109](#)
  - Chip\_DMA\_Table, [110](#)
- CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [111](#)
  - Chip\_DMA\_AbortChannel, [111](#)
  - Chip\_DMA\_ClearActiveIntAChannel, [112](#)
  - Chip\_DMA\_ClearActiveIntBChannel, [112](#)
  - Chip\_DMA\_ClearErrorIntChannel, [112](#)
  - Chip\_DMA\_DisableChannel, [113](#)
  - Chip\_DMA\_DisableIntChannel, [113](#)
  - Chip\_DMA\_EnableChannel, [113](#)
  - Chip\_DMA\_EnableIntChannel, [113](#)
  - Chip\_DMA\_GetActiveChannels, [114](#)
  - Chip\_DMA\_GetActiveIntAChannels, [114](#)
  - Chip\_DMA\_GetActiveIntBChannels, [114](#)
  - Chip\_DMA\_GetBusyChannels, [115](#)
  - Chip\_DMA\_GetEnableIntChannels, [115](#)
  - Chip\_DMA\_GetEnabledChannels, [115](#)
  - Chip\_DMA\_GetErrorIntChannels, [116](#)
  - Chip\_DMA\_SetTrigChannel, [116](#)
  - Chip\_DMA\_SetValidChannel, [116](#)

- CHIP: LPC5411X DMA Controller driver common functions (legacy), [118](#)
  - Chip\_DMA\_DeInit, [118](#)
  - Chip\_DMA\_Disable, [118](#)
  - Chip\_DMA\_Enable, [118](#)
  - Chip\_DMA\_GetIntStatus, [119](#)
  - Chip\_DMA\_GetSRAMBase, [119](#)
  - Chip\_DMA\_Init, [119](#)
  - Chip\_DMA\_SetSRAMBase, [119](#)
- CHIP: LPC5411X DMA Engine driver (legacy), [121](#)
  - DMA\_ADDR, [122](#)
  - DMA\_CFG\_BURSTPOWER, [122](#)
  - DMA\_CFG\_BURSTPOWER\_1, [122](#)
  - DMA\_CFG\_BURSTPOWER\_1024, [122](#)
  - DMA\_CFG\_BURSTPOWER\_128, [123](#)
  - DMA\_CFG\_BURSTPOWER\_16, [123](#)
  - DMA\_CFG\_BURSTPOWER\_2, [123](#)
  - DMA\_CFG\_BURSTPOWER\_256, [123](#)
  - DMA\_CFG\_BURSTPOWER\_32, [123](#)
  - DMA\_CFG\_BURSTPOWER\_4, [123](#)
  - DMA\_CFG\_BURSTPOWER\_512, [123](#)
  - DMA\_CFG\_BURSTPOWER\_64, [123](#)
  - DMA\_CFG\_BURSTPOWER\_8, [123](#)
  - DMA\_CFG\_CHPRIORITY, [124](#)
  - DMA\_CFG\_DSTBURSTWRAP, [124](#)
  - DMA\_CFG\_HWTRIGEN, [124](#)
  - DMA\_CFG\_PERIPHREQEN, [124](#)
  - DMA\_CFG\_SRCBURSTWRAP, [124](#)
  - DMA\_CFG\_TRIGBURST\_BURST, [124](#)
  - DMA\_CFG\_TRIGBURST\_SNGL, [124](#)
  - DMA\_CFG\_TRIGPOL\_HIGH, [124](#)
  - DMA\_CFG\_TRIGPOL\_LOW, [124](#)
  - DMA\_CFG\_TRIGTYPE\_EDGE, [125](#)
  - DMA\_CFG\_TRIGTYPE\_LEVEL, [125](#)
  - DMA\_CH0, [127](#)
  - DMA\_CH1, [127](#)
  - DMA\_CH10, [128](#)
  - DMA\_CH11, [128](#)
  - DMA\_CH12, [128](#)
  - DMA\_CH13, [128](#)
  - DMA\_CH14, [128](#)
  - DMA\_CH15, [128](#)
  - DMA\_CH16, [128](#)
  - DMA\_CH17, [128](#)
  - DMA\_CH18, [128](#)
  - DMA\_CH19, [128](#)
  - DMA\_CH2, [127](#)
  - DMA\_CH3, [127](#)
  - DMA\_CH4, [128](#)
  - DMA\_CH5, [128](#)
  - DMA\_CH6, [128](#)
  - DMA\_CH7, [128](#)
  - DMA\_CH8, [128](#)
  - DMA\_CH9, [128](#)
  - DMA\_CHID\_T, [127](#)
  - DMA\_CTLSTAT\_TRIG, [125](#)
  - DMA\_CTLSTAT\_VALIDPENDING, [125](#)
  - DMA\_INTSTAT\_ACTIVEERRINT, [125](#)
  - DMA\_INTSTAT\_ACTIVEINT, [125](#)
  - DMA\_XFERCFG\_CFGVALID, [125](#)
  - DMA\_XFERCFG\_CLRTRIG, [125](#)
  - DMA\_XFERCFG\_DSTINC\_0, [125](#)
  - DMA\_XFERCFG\_DSTINC\_1, [126](#)
  - DMA\_XFERCFG\_DSTINC\_2, [126](#)
  - DMA\_XFERCFG\_DSTINC\_4, [126](#)
  - DMA\_XFERCFG\_RELOAD, [126](#)
  - DMA\_XFERCFG\_SETINTA, [126](#)
  - DMA\_XFERCFG\_SETINTB, [126](#)
  - DMA\_XFERCFG\_SRCINC\_0, [126](#)
  - DMA\_XFERCFG\_SRCINC\_1, [126](#)
  - DMA\_XFERCFG\_SRCINC\_2, [126](#)
  - DMA\_XFERCFG\_SRCINC\_4, [127](#)
  - DMA\_XFERCFG\_SWTRIG, [127](#)
  - DMA\_XFERCFG\_WIDTH\_16, [127](#)
  - DMA\_XFERCFG\_WIDTH\_32, [127](#)
  - DMA\_XFERCFG\_WIDTH\_8, [127](#)
  - DMA\_XFERCFG\_XFERCOUNT, [127](#)
  - DMAREQ\_DMIC0, [128](#)
  - DMAREQ\_DMIC1, [128](#)
  - DMAREQ\_FLEXCOMM0\_RX, [128](#)
  - DMAREQ\_FLEXCOMM0\_TX, [128](#)
  - DMAREQ\_FLEXCOMM1\_RX, [128](#)
  - DMAREQ\_FLEXCOMM1\_TX, [128](#)
  - DMAREQ\_FLEXCOMM2\_RX, [128](#)
  - DMAREQ\_FLEXCOMM2\_TX, [128](#)
  - DMAREQ\_FLEXCOMM3\_RX, [128](#)
  - DMAREQ\_FLEXCOMM3\_TX, [128](#)
  - DMAREQ\_FLEXCOMM4\_RX, [128](#)
  - DMAREQ\_FLEXCOMM4\_TX, [128](#)
  - DMAREQ\_FLEXCOMM5\_RX, [128](#)
  - DMAREQ\_FLEXCOMM5\_TX, [128](#)
  - DMAREQ\_FLEXCOMM6\_RX, [128](#)
  - DMAREQ\_FLEXCOMM6\_TX, [128](#)
  - DMAREQ\_FLEXCOMM7\_RX, [128](#)
  - DMAREQ\_FLEXCOMM7\_TX, [128](#)
  - DMAREQ\_SPIFI, [128](#)
  - MAX\_DMA\_CHANNEL, [127](#)
  - CHIP: LPC5411X DMA Service driver, [129](#)
    - channel, [132](#)
    - Chip\_DMASERVICE\_DoubleBuffer, [129](#)
    - Chip\_DMASERVICE\_Init, [130](#)
    - Chip\_DMASERVICE\_Isr, [130](#)
    - Chip\_DMASERVICE\_RegisterCb, [130](#)
    - Chip\_DMASERVICE\_SingleBuffer, [130](#)
    - DMA\_CALLBACK\_T, [129](#)
    - descr, [132](#)
    - dst\_increment, [132](#)
    - register\_location, [132](#)
    - src\_increment, [132](#)
    - width, [132](#)
    - write, [132](#)
  - CHIP: LPC5411X DMIC driver, [133](#)
    - CHANEN, [143](#)
    - CHANNEL, [143](#)
    - COMPENSATION\_T, [137](#)
    - Chip\_DMIC\_CfgChannel, [138](#)



- Chip\_DMIC\_CfgChannelDc, [139](#)
- Chip\_DMIC\_CfgIO, [139](#)
- Chip\_DMIC\_EnableChannel, [139](#)
- Chip\_DMIC\_FifoChannel, [139](#)
- Chip\_DMIC\_FifoClearStatus, [141](#)
- Chip\_DMIC\_FifoGetData, [141](#)
- Chip\_DMIC\_FifoGetStatus, [141](#)
- Chip\_DMIC\_Init, [141](#)
- Chip\_DMIC\_SetOpMode, [143](#)
- Chip\_DMIC\_Use2fs, [143](#)
- DC\_CTRL, [143](#)
- DC\_REMOVAL\_T, [137](#)
- DIVHFCLK, [143](#)
- DMIC\_COMP0\_0, [137](#)
- DMIC\_COMP0\_13, [137](#)
- DMIC\_COMP0\_15, [137](#)
- DMIC\_COMP0\_16, [137](#)
- DMIC\_DC\_CUT155, [137](#)
- DMIC\_DC\_CUT39, [137](#)
- DMIC\_DC\_CUT78, [137](#)
- DMIC\_DC\_NOREMOVE, [137](#)
- DMIC\_DCGAIN\_REDUCE\_P, [135](#)
- DMIC\_DCPOLE\_P, [135](#)
- DMIC\_FIFO\_DMAEN, [135](#)
- DMIC\_FIFO\_DMAEN\_P, [135](#)
- DMIC\_FIFO\_ENABLE, [135](#)
- DMIC\_FIFO\_ENABLE\_P, [135](#)
- DMIC\_FIFO\_INT, [135](#)
- DMIC\_FIFO\_INT\_P, [135](#)
- DMIC\_FIFO\_INTREN, [136](#)
- DMIC\_FIFO\_INTREN\_P, [136](#)
- DMIC\_FIFO\_OVERRUN, [136](#)
- DMIC\_FIFO\_OVERRUN\_P, [136](#)
- DMIC\_FIFO\_RESETN, [136](#)
- DMIC\_FIFO\_RESETN\_P, [136](#)
- DMIC\_FIFO\_TLVL\_P, [136](#)
- DMIC\_FIFO\_UNDERRUN, [136](#)
- DMIC\_FIFO\_UNDERRUN\_P, [136](#)
- DMIC\_IO\_T, [137](#)
- DMIC\_LEFT, [138](#)
- DMIC\_OP\_DMA, [138](#)
- DMIC\_OP\_INTR, [138](#)
- DMIC\_OP\_POLL, [138](#)
- DMIC\_PDM\_DIV1, [138](#)
- DMIC\_PDM\_DIV12, [138](#)
- DMIC\_PDM\_DIV128, [138](#)
- DMIC\_PDM\_DIV16, [138](#)
- DMIC\_PDM\_DIV2, [138](#)
- DMIC\_PDM\_DIV24, [138](#)
- DMIC\_PDM\_DIV3, [138](#)
- DMIC\_PDM\_DIV32, [138](#)
- DMIC\_PDM\_DIV4, [138](#)
- DMIC\_PDM\_DIV48, [138](#)
- DMIC\_PDM\_DIV6, [138](#)
- DMIC\_PDM\_DIV64, [138](#)
- DMIC\_PDM\_DIV8, [138](#)
- DMIC\_PDM\_DIV96, [138](#)
- DMIC\_PHY\_FALL, [136](#)
- DMIC\_PHY\_FALL\_P, [136](#)
- DMIC\_PHY\_HALF, [136](#)
- DMIC\_PHY\_HALF\_P, [137](#)
- DMIC\_RIGHT, [138](#)
- DMIC\_SATURATE\_AT16BIT\_P, [137](#)
- divhclk, [144](#)
- dmic\_ch0\_dma\_context, [144](#)
- dmic\_ch0\_dma\_interleaved\_context, [144](#)
- dmic\_ch1\_dma\_context, [144](#)
- dmic\_ch1\_dma\_interleaved\_context, [144](#)
- FIFO\_CTRL, [144](#)
- FIFO\_DATA, [144](#)
- FIFO\_STATUS, [144](#)
- fifo\_ints, [144](#)
- fifo\_overrun, [144](#)
- fifo\_underrun, [144](#)
- GAINSHFT, [145](#)
- gainshft, [145](#)
- HWVADGAIN, [145](#)
- HWVADHPFS, [145](#)
- HWVADLOWZ, [145](#)
- HWVADRSTT, [145](#)
- HWVADST10, [145](#)
- HWVADTHGN, [145](#)
- HWVADTHGS, [145](#)
- ID, [145](#)
- IOCFG, [145](#)
- OP\_MODE\_T, [137](#)
- OSR, [145](#)
- osr, [146](#)
- PDM\_DIV\_T, [138](#)
- PHY\_CTRL, [146](#)
- PREAC2FSCOEF, [146](#)
- PREAC4FSCOEF, [146](#)
- pdm\_bypass, [137](#)
- pdm\_bypass\_clk0, [137](#)
- pdm\_bypass\_clk1, [137](#)
- pdm\_dual, [137](#)
- pdm\_stereo, [137](#)
- preac2coef, [146](#)
- preac4coef, [146](#)
- reserved, [146](#)
- reserved0, [146](#)
- reserved1, [146](#)
- STEREO\_SIDE\_T, [138](#)
- side, [146](#)
- TDM19EN, [147](#)
- TDM96EN, [147](#)
- USE2FS, [147](#)
- CHIP: LPC5411X Enhanced boot block support, [148](#)
- IFSEL\_T, [149](#)
- IMAGE\_BOOT\_BLOCK\_OFF, [148](#)
- IMAGE\_DUAL\_ENH\_SIG, [148](#)
- IMAGE\_ENH\_BLOCK\_MARKER, [148](#)
- IMAGE\_ENH\_MARKER\_OFF, [148](#)
- IMAGE\_SINGLE\_ENH\_SIG, [148](#)
- IMAGE\_T, [149](#)
- IMG\_ISP\_WAIT, [149](#)

- IMG\_JUST\_BOOT, [149](#)
- IMG\_NO\_CRC, [149](#)
- IMG\_NO\_WAIT, [149](#)
- IMG\_NORMAL, [149](#)
- SETPORTPIN, [148](#)
- SL\_AUTO, [149](#)
- SL\_I2C0, [149](#)
- SL\_I2C1, [149](#)
- SL\_I2C2, [149](#)
- SL\_SPI0, [149](#)
- SL\_SPI1, [149](#)
- CHIP: LPC5411X flexcomm API, [378](#)
  - Chip\_FLEXCOMM\_DelInit, [380](#)
  - Chip\_FLEXCOMM\_GetFunc, [380](#)
  - Chip\_FLEXCOMM\_GetIndex, [380](#)
  - Chip\_FLEXCOMM\_Init, [382](#)
  - Chip\_FLEXCOMM\_IsLocked, [382](#)
  - Chip\_FLEXCOMM\_Lock, [382](#)
  - Chip\_FLEXCOMM\_SetPeriph, [382](#)
  - ERR\_FLEXCOMM\_FUNCNOTSUPPORTED, [379](#)
  - ERR\_FLEXCOMM\_INVALIDBASE, [379](#)
  - ERR\_FLEXCOMM\_NOTFREE, [379](#)
  - FLEXCOMM\_ID\_I2C, [379](#)
  - FLEXCOMM\_ID\_I2S, [379](#)
  - FLEXCOMM\_ID\_SPI, [379](#)
  - FLEXCOMM\_ID\_USART, [379](#)
  - FLEXCOMM\_LOCK, [379](#)
  - FLEXCOMM\_PERIPH\_I2C, [380](#)
  - FLEXCOMM\_PERIPH\_I2S\_RX, [380](#)
  - FLEXCOMM\_PERIPH\_I2S\_TX, [380](#)
  - FLEXCOMM\_PERIPH\_NONE, [380](#)
  - FLEXCOMM\_PERIPH\_SPI, [380](#)
  - FLEXCOMM\_PERIPH\_T, [380](#)
  - FLEXCOMM\_PERIPH\_USART, [380](#)
  - FLEXCOMM\_PSEL\_OFFSET, [379](#)
  - LPC\_FLEXCOMM\_T, [380](#)
- CHIP: LPC5411X GPIO driver, [150](#)
  - Chip\_GPIO\_ClearValue, [151](#)
  - Chip\_GPIO\_DelInit, [151](#)
  - Chip\_GPIO\_GetMaskedPortValue, [153](#)
  - Chip\_GPIO\_GetPinDIR, [153](#)
  - Chip\_GPIO\_GetPinState, [153](#)
  - Chip\_GPIO\_GetPortDIR, [153](#)
  - Chip\_GPIO\_GetPortMask, [155](#)
  - Chip\_GPIO\_GetPortValue, [155](#)
  - Chip\_GPIO\_Init, [155](#)
  - Chip\_GPIO\_ReadDirBit, [156](#)
  - Chip\_GPIO\_ReadPortBit, [156](#)
  - Chip\_GPIO\_ReadValue, [156](#)
  - Chip\_GPIO\_SetDir, [157](#)
  - Chip\_GPIO\_SetMaskedPortValue, [157](#)
  - Chip\_GPIO\_SetPinDIR, [157](#)
  - Chip\_GPIO\_SetPinDIRInput, [157](#)
  - Chip\_GPIO\_SetPinDIROutput, [158](#)
  - Chip\_GPIO\_SetPinOutHigh, [158](#)
  - Chip\_GPIO\_SetPinOutLow, [158](#)
  - Chip\_GPIO\_SetPinState, [159](#)
  - Chip\_GPIO\_SetPinToggle, [159](#)
  - Chip\_GPIO\_SetPortDIR, [159](#)
  - Chip\_GPIO\_SetPortDIRInput, [160](#)
  - Chip\_GPIO\_SetPortDIROutput, [160](#)
  - Chip\_GPIO\_SetPortMask, [160](#)
  - Chip\_GPIO\_SetPortOutHigh, [162](#)
  - Chip\_GPIO\_SetPortOutLow, [162](#)
  - Chip\_GPIO\_SetPortToggle, [162](#)
  - Chip\_GPIO\_SetPortValue, [163](#)
  - Chip\_GPIO\_SetValue, [163](#)
  - Chip\_GPIO\_WriteDirBit, [163](#)
  - Chip\_GPIO\_WritePortBit, [164](#)
- CHIP: LPC5411X GPIO group driver, [165](#)
  - Chip\_GPIOPG\_ClearIntStatus, [166](#)
  - Chip\_GPIOPG\_DelInit, [166](#)
  - Chip\_GPIOPG\_DisableGroupPins, [166](#)
  - Chip\_GPIOPG\_EnableGroupPins, [167](#)
  - Chip\_GPIOPG\_GetIntStatus, [167](#)
  - Chip\_GPIOPG\_Init, [167](#)
  - Chip\_GPIOPG\_SelectAndMode, [168](#)
  - Chip\_GPIOPG\_SelectEdgeMode, [168](#)
  - Chip\_GPIOPG\_SelectHighLevel, [168](#)
  - Chip\_GPIOPG\_SelectLevelMode, [168](#)
  - Chip\_GPIOPG\_SelectLowLevel, [170](#)
  - Chip\_GPIOPG\_SelectOrMode, [170](#)
  - GPIOPGR\_COMB, [166](#)
  - GPIOPGR\_INT, [166](#)
  - GPIOPGR\_TRIG, [166](#)
- CHIP: LPC5411X I2C master-only driver, [171](#)
  - Chip\_I2CM\_ClearStatus, [173](#)
  - Chip\_I2CM\_Disable, [173](#)
  - Chip\_I2CM\_Enable, [173](#)
  - Chip\_I2CM\_GetMasterState, [173](#)
  - Chip\_I2CM\_GetStatus, [174](#)
  - Chip\_I2CM\_IsMasterPending, [174](#)
  - Chip\_I2CM\_MasterContinue, [174](#)
  - Chip\_I2CM\_ReadByte, [175](#)
  - Chip\_I2CM\_SendStart, [175](#)
  - Chip\_I2CM\_SendStop, [175](#)
  - Chip\_I2CM\_SetBusSpeed, [176](#)
  - Chip\_I2CM\_SetDutyCycle, [176](#)
  - Chip\_I2CM\_WriteByte, [177](#)
  - Chip\_I2CM\_Xfer, [177](#)
  - Chip\_I2CM\_XferBlocking, [178](#)
  - Chip\_I2CM\_XferHandler, [178](#)
  - I2CM\_STATUS\_ARBLOST, [172](#)
  - I2CM\_STATUS\_BUS\_ERROR, [172](#)
  - I2CM\_STATUS\_BUSY, [172](#)
  - I2CM\_STATUS\_ERROR, [172](#)
  - I2CM\_STATUS\_NAK\_ADR, [172](#)
  - I2CM\_STATUS\_NAK\_DAT, [172](#)
  - I2CM\_STATUS\_OK, [172](#)
- CHIP: LPC5411X I2C slave-only driver, [179](#)
  - Chip\_I2CS\_ClearStatus, [181](#)
  - Chip\_I2CS\_Disable, [181](#)
  - Chip\_I2CS\_DisableSlaveAddr, [181](#)
  - Chip\_I2CS\_Enable, [182](#)
  - Chip\_I2CS\_EnableSlaveAddr, [182](#)
  - Chip\_I2CS\_GetSlaveAddr, [182](#)

- Chip\_I2CS\_GetSlaveMatchIndex, [182](#)
- Chip\_I2CS\_GetSlaveState, [183](#)
- Chip\_I2CS\_GetStatus, [183](#)
- Chip\_I2CS\_IsSlaveDeSelected, [183](#)
- Chip\_I2CS\_IsSlavePending, [183](#)
- Chip\_I2CS\_IsSlaveSelected, [184](#)
- Chip\_I2CS\_ReadByte, [184](#)
- Chip\_I2CS\_SetSlaveAddr, [184](#)
- Chip\_I2CS\_SetSlaveQual0, [185](#)
- Chip\_I2CS\_SlaveContinue, [185](#)
- Chip\_I2CS\_SlaveDisableDMA, [185](#)
- Chip\_I2CS\_SlaveEnableDMA, [186](#)
- Chip\_I2CS\_SlaveNACK, [186](#)
- Chip\_I2CS\_WriteByte, [186](#)
- Chip\_I2CS\_XferHandler, [187](#)
- I2CSlaveXferDone, [180](#)
- I2CSlaveXferRecv, [180](#)
- I2CSlaveXferSend, [180](#)
- I2CSlaveXferStart, [181](#)
- CHIP: LPC5411X IOCON register block and driver, [188](#)
  - Chip\_IOCON\_PinMux, [192](#)
  - Chip\_IOCON\_PinMuxSet, [192](#)
  - Chip\_IOCON\_SetPinMuxing, [192](#)
  - IOCON\_ANALOG\_EN, [189](#)
  - IOCON\_CLKDIV, [189](#)
  - IOCON\_DIGITAL\_EN, [189](#)
  - IOCON\_FASTI2C\_EN, [189](#)
  - IOCON\_FUNC0, [189](#)
  - IOCON\_FUNC1, [189](#)
  - IOCON\_FUNC2, [189](#)
  - IOCON\_FUNC3, [189](#)
  - IOCON\_FUNC4, [189](#)
  - IOCON\_FUNC5, [189](#)
  - IOCON\_FUNC6, [190](#)
  - IOCON\_FUNC7, [190](#)
  - IOCON\_GPIO\_MODE, [190](#)
  - IOCON\_HYS\_EN, [190](#)
  - IOCON\_I2C\_SLEW, [190](#)
  - IOCON\_INPFILT\_OFF, [190](#)
  - IOCON\_INPFILT\_ON, [190](#)
  - IOCON\_INV\_EN, [190](#)
  - IOCON\_MODE\_INACT, [190](#)
  - IOCON\_MODE\_PULLDOWN, [191](#)
  - IOCON\_MODE\_PULLUP, [191](#)
  - IOCON\_MODE\_REPEATER, [191](#)
  - IOCON\_OPENDRAIN\_EN, [191](#)
  - IOCON\_S\_MODE, [191](#)
  - IOCON\_S\_MODE\_0CLK, [191](#)
  - IOCON\_S\_MODE\_1CLK, [191](#)
  - IOCON\_S\_MODE\_2CLK, [191](#)
  - IOCON\_S\_MODE\_3CLK, [191](#)
  - IOCON\_STDI2C\_EN, [192](#)
- CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
  - Chip\_INMUX\_PinIntSel, [195](#)
  - Chip\_INMUX\_SetDMAOutMux, [195](#)
  - Chip\_INMUX\_SetDMATrigger, [196](#)
  - Chip\_INMUX\_SetFreqMeasRefClock, [196](#)
  - Chip\_INMUX\_SetFreqMeasTargClock, [196](#)
- DMA\_TRIGSRC\_T, [194](#)
- DMATRIG\_ADC0\_SEQA\_IRQ, [194](#)
- DMATRIG\_ADC0\_SEQB\_IRQ, [194](#)
- DMATRIG\_OUTMUX0, [195](#)
- DMATRIG\_OUTMUX1, [195](#)
- DMATRIG\_OUTMUX2, [195](#)
- DMATRIG\_OUTMUX3, [195](#)
- DMATRIG\_PININT0, [195](#)
- DMATRIG\_PININT1, [195](#)
- DMATRIG\_PININT2, [195](#)
- DMATRIG\_PININT3, [195](#)
- DMATRIG\_SCT0\_DMA0, [194](#)
- DMATRIG\_SCT0\_DMA1, [194](#)
- DMATRIG\_TIMER0\_MATCH0, [194](#)
- DMATRIG\_TIMER0\_MATCH1, [194](#)
- DMATRIG\_TIMER1\_MATCH0, [194](#)
- DMATRIG\_TIMER2\_MATCH0, [194](#)
- DMATRIG\_TIMER2\_MATCH1, [194](#)
- DMATRIG\_TIMER3\_MATCH0, [195](#)
- DMATRIG\_TIMER4\_MATCH0, [195](#)
- DMATRIG\_TIMER4\_MATCH1, [195](#)
- FREQ\_MEAS\_MAIN\_CLK, [195](#)
- FREQMSR\_32KHZOSC, [195](#)
- FREQMSR\_CLKIN, [195](#)
- FREQMSR\_FRO12MHZ, [195](#)
- FREQMSR\_PIO0\_20, [195](#)
- FREQMSR\_PIO0\_24, [195](#)
- FREQMSR\_PIO0\_4, [195](#)
- FREQMSR\_PIO1\_4, [195](#)
- FREQMSR\_SRC\_T, [195](#)
- FREQMSR\_WDOSC, [195](#)
- CHIP: LPC5411X M0 core CMSIS include file, [197](#)
- CHIP: LPC5411X M0 core Cortex CMSIS definitions, [198](#)
  - \_\_CM0PLUS\_REV, [198](#)
  - \_\_MPU\_PRESENT, [198](#)
  - \_\_NVIC\_PRIO\_BITS, [198](#)
  - \_\_VTOR\_PRESENT, [199](#)
  - \_\_Vendor\_SysTickConfig, [198](#)
  - ADC0\_SEQA\_IRQHandler, [199](#)
  - ADC0\_SEQA\_IRQn, [199](#)
  - ADC0\_SEQB\_IRQHandler, [199](#)
  - ADC0\_SEQB\_IRQn, [199](#)
  - ADC0\_THCMP\_IRQHandler, [199](#)
  - ADC0\_THCMP\_IRQn, [199](#)
  - SCT\_IRQHandler, [199](#)
  - SCT\_IRQn, [199](#)
  - TIMER0\_IRQHandler, [199](#)
  - TIMER0\_IRQn, [199](#)
  - TIMER1\_IRQHandler, [200](#)
  - TIMER1\_IRQn, [200](#)
  - TIMER2\_IRQHandler, [200](#)
  - TIMER2\_IRQn, [200](#)
  - TIMER3\_IRQHandler, [200](#)
  - TIMER3\_IRQn, [200](#)
  - TIMER4\_IRQHandler, [200](#)
  - TIMER4\_IRQn, [200](#)
- CHIP: LPC5411X M4 core CMSIS include file, [201](#)



- CHIP: LPC5411X M4 core Cortex CMSIS definitions, [202](#)
- [\\_\\_CM4\\_REV](#), [202](#)
  - [\\_\\_FPU\\_PRESENT](#), [202](#)
  - [\\_\\_MPU\\_PRESENT](#), [202](#)
  - [\\_\\_NVIC\\_PRIO\\_BITS](#), [202](#)
  - [\\_\\_Vendor\\_SysTickConfig](#), [202](#)
  - [ADC0\\_SEQA\\_IRQHandler](#), [203](#)
  - [ADC0\\_SEQA\\_IRQn](#), [203](#)
  - [ADC0\\_SEQB\\_IRQHandler](#), [203](#)
  - [ADC0\\_SEQB\\_IRQn](#), [203](#)
  - [ADC0\\_THCMP\\_IRQHandler](#), [203](#)
  - [ADC0\\_THCMP\\_IRQn](#), [203](#)
  - [SCT\\_IRQHandler](#), [203](#)
  - [SCT\\_IRQn](#), [203](#)
  - [TIMER0\\_IRQHandler](#), [203](#)
  - [TIMER0\\_IRQn](#), [203](#)
  - [TIMER1\\_IRQHandler](#), [203](#)
  - [TIMER1\\_IRQn](#), [204](#)
  - [TIMER2\\_IRQHandler](#), [204](#)
  - [TIMER2\\_IRQn](#), [204](#)
  - [TIMER3\\_IRQHandler](#), [204](#)
  - [TIMER3\\_IRQn](#), [204](#)
  - [TIMER4\\_IRQHandler](#), [204](#)
  - [TIMER4\\_IRQn](#), [204](#)
- CHIP: LPC5411X Mailbox M4/M0+ driver, [205](#)
- [Chip\\_MBOX\\_ClearValueBits](#), [206](#)
  - [Chip\\_MBOX\\_DeInit](#), [206](#)
  - [Chip\\_MBOX\\_GetMutex](#), [206](#)
  - [Chip\\_MBOX\\_GetValue](#), [206](#)
  - [Chip\\_MBOX\\_Init](#), [208](#)
  - [Chip\\_MBOX\\_SetMutex](#), [208](#)
  - [Chip\\_MBOX\\_SetValue](#), [208](#)
  - [Chip\\_MBOX\\_SetValueBits](#), [209](#)
  - [MAILBOX\\_AVAIL](#), [205](#)
  - [MAILBOX\\_CM0PLUS](#), [205](#)
  - [MAILBOX\\_CM4](#), [205](#)
  - [MBOX\\_IDX\\_T](#), [205](#)
- CHIP: LPC5411X Micro Tick driver, [210](#)
- [Chip\\_UTICK\\_ClearInterrupt](#), [211](#)
  - [Chip\\_UTICK\\_DeInit](#), [211](#)
  - [Chip\\_UTICK\\_GetStatus](#), [211](#)
  - [Chip\\_UTICK\\_GetTick](#), [212](#)
  - [Chip\\_UTICK\\_Halt](#), [212](#)
  - [Chip\\_UTICK\\_Init](#), [212](#)
  - [Chip\\_UTICK\\_SetDelayMs](#), [212](#)
  - [Chip\\_UTICK\\_SetTick](#), [213](#)
  - [UTICK\\_CTRL\\_DELAY\\_MASK](#), [210](#)
  - [UTICK\\_CTRL\\_REPEAT](#), [210](#)
  - [UTICK\\_STATUS\\_ACTIVE](#), [210](#)
  - [UTICK\\_STATUS\\_INTR](#), [211](#)
  - [UTICK\\_STATUS\\_MASK](#), [211](#)
- CHIP: LPC5411X Multi-Rate Timer driver, [214](#)
- [Chip\\_MRT\\_ClearIntPending](#), [217](#)
  - [Chip\\_MRT\\_DeInit](#), [217](#)
  - [Chip\\_MRT\\_GetEnabled](#), [217](#)
  - [Chip\\_MRT\\_GetIdleChannel](#), [218](#)
  - [Chip\\_MRT\\_GetIdleChannelShifted](#), [218](#)
  - [Chip\\_MRT\\_GetIntPending](#), [218](#)
  - [Chip\\_MRT\\_GetIntPendingByChannel](#), [218](#)
  - [Chip\\_MRT\\_GetInterval](#), [218](#)
  - [Chip\\_MRT\\_GetMode](#), [219](#)
  - [Chip\\_MRT\\_GetRegPtr](#), [219](#)
  - [Chip\\_MRT\\_GetTimer](#), [219](#)
  - [Chip\\_MRT\\_Init](#), [219](#)
  - [Chip\\_MRT\\_IntClear](#), [219](#)
  - [Chip\\_MRT\\_IntPending](#), [220](#)
  - [Chip\\_MRT\\_IsOneShotMode](#), [220](#)
  - [Chip\\_MRT\\_IsRepeatMode](#), [220](#)
  - [Chip\\_MRT\\_Running](#), [220](#)
  - [Chip\\_MRT\\_SetDisabled](#), [221](#)
  - [Chip\\_MRT\\_SetEnabled](#), [221](#)
  - [Chip\\_MRT\\_SetInterval](#), [221](#)
  - [Chip\\_MRT\\_SetMode](#), [221](#)
  - [LPC\\_MRT\\_CH](#), [215](#)
  - [LPC\\_MRT\\_CH0](#), [215](#)
  - [LPC\\_MRT\\_CH1](#), [215](#)
  - [LPC\\_MRT\\_CH2](#), [215](#)
  - [LPC\\_MRT\\_CH3](#), [215](#)
  - [MRT0\\_INTFLAG](#), [216](#)
  - [MRT1\\_INTFLAG](#), [216](#)
  - [MRT2\\_INTFLAG](#), [216](#)
  - [MRT3\\_INTFLAG](#), [216](#)
  - [MRT\\_CHANNELS\\_NUM](#), [216](#)
  - [MRT\\_CTRL\\_INTEN\\_MASK](#), [216](#)
  - [MRT\\_CTRL\\_MODE\\_MASK](#), [216](#)
  - [MRT\\_INTVAL\\_IVALUE](#), [216](#)
  - [MRT\\_INTVAL\\_LOAD](#), [216](#)
  - [MRT\\_MODE\\_ONESHOT](#), [217](#)
  - [MRT\\_MODE\\_REPEAT](#), [217](#)
  - [MRT\\_MODE\\_T](#), [217](#)
  - [MRT\\_NO\\_IDLE\\_CHANNEL](#), [216](#)
  - [MRT\\_STAT\\_INTFLAG](#), [216](#)
  - [MRT\\_STAT\\_RUNNING](#), [216](#)
  - [MRTn\\_INTFLAG](#), [217](#)
- CHIP: LPC5411X PLL Driver, [224](#)
- [CHIP\\_SYSCON\\_PLLCLKSRC\\_T](#), [227](#)
  - [Chip\\_Clock\\_GetStoredPLLClockRate](#), [228](#)
  - [Chip\\_Clock\\_GetSystemPLLInClockRate](#), [228](#)
  - [Chip\\_Clock\\_GetSystemPLLOutClockRate](#), [228](#)
  - [Chip\\_Clock\\_GetSystemPLLOutFromSetup](#), [229](#)
  - [Chip\\_Clock\\_IsSystemPLLLocked](#), [229](#)
  - [Chip\\_Clock\\_SetBypassPLL](#), [229](#)
  - [Chip\\_Clock\\_SetPLLFreq](#), [229](#)
  - [Chip\\_Clock\\_SetStoredPLLClockRate](#), [230](#)
  - [Chip\\_Clock\\_SetSystemPLLSource](#), [230](#)
  - [Chip\\_Clock\\_SetupPLLData](#), [230](#)
  - [Chip\\_Clock\\_SetupSystemPLL](#), [231](#)
  - [Chip\\_Clock\\_SetupSystemPLLPrec](#), [231](#)
  - [PLL\\_CONFIGFLAG\\_FORCENOFRACT](#), [226](#)
  - [PLL\\_CONFIGFLAG\\_USEINRATE](#), [226](#)
  - [PLL\\_ERROR\\_INPUT\\_TOO\\_HIGH](#), [227](#)
  - [PLL\\_ERROR\\_INPUT\\_TOO\\_LOW](#), [227](#)
  - [PLL\\_ERROR\\_OUTPUT\\_TOO\\_HIGH](#), [227](#)
  - [PLL\\_ERROR\\_OUTPUT\\_TOO\\_LOW](#), [227](#)
  - [PLL\\_ERROR\\_OUTSIDE\\_INTLIMIT](#), [227](#)

- PLL\_ERROR\_SUCCESS, [227](#)
- PLL\_ERROR\_T, [227](#)
- PLL\_SETUPFLAG\_ADGVOLT, [226](#)
- PLL\_SETUPFLAG\_POWERUP, [226](#)
- PLL\_SETUPFLAG\_WAITLOCK, [226](#)
- SS\_MC\_MAXC, [227](#)
- SS\_MC\_NOC, [227](#)
- SS\_MC\_RECC, [227](#)
- SS\_MF\_128, [228](#)
- SS\_MF\_16, [228](#)
- SS\_MF\_24, [228](#)
- SS\_MF\_256, [228](#)
- SS\_MF\_32, [228](#)
- SS\_MF\_384, [228](#)
- SS\_MF\_512, [228](#)
- SS\_MF\_64, [228](#)
- SS\_MODWVCTRL\_T, [227](#)
- SS\_MR\_K0, [228](#)
- SS\_MR\_K1, [228](#)
- SS\_MR\_K1\_5, [228](#)
- SS\_MR\_K2, [228](#)
- SS\_MR\_K3, [228](#)
- SS\_MR\_K4, [228](#)
- SS\_MR\_K6, [228](#)
- SS\_MR\_K8, [228](#)
- SS\_PROGMODDP\_T, [227](#)
- SS\_PROGMODFM\_T, [228](#)
- SYSCON\_PLLCLKSRC\_CLKIN, [227](#)
- SYSCON\_PLLCLKSRC\_DISABLED, [227](#)
- SYSCON\_PLLCLKSRC\_FRO12MHZ, [227](#)
- SYSCON\_PLLCLKSRC\_RTC, [227](#)
- SYSCON\_PLLCLKSRC\_WDT, [227](#)
- CHIP: LPC5411X Peripheral addresses and register set declarations, [232](#)
  - LPC\_ADC, [233](#)
  - LPC\_ADC\_BASE, [233](#)
  - LPC\_ASYNC\_SYSCON, [233](#)
  - LPC\_ASYNC\_SYSCON\_BASE, [233](#)
  - LPC\_CRC, [233](#)
  - LPC\_CRC\_BASE, [233](#)
  - LPC\_DMA, [233](#)
  - LPC\_DMA\_BASE, [233](#)
  - LPC\_DMIC, [234](#)
  - LPC\_DMIC\_BASE, [234](#)
  - LPC\_FLASHMEM\_BASE, [234](#)
  - LPC\_FLEXCOMM0\_BASE, [234](#)
  - LPC\_FLEXCOMM1\_BASE, [234](#)
  - LPC\_FLEXCOMM2\_BASE, [234](#)
  - LPC\_FLEXCOMM3\_BASE, [234](#)
  - LPC\_FLEXCOMM4\_BASE, [234](#)
  - LPC\_FLEXCOMM5\_BASE, [234](#)
  - LPC\_FLEXCOMM6\_BASE, [234](#)
  - LPC\_FLEXCOMM7\_BASE, [234](#)
  - LPC\_FMC\_BASE, [234](#)
  - LPC\_GINT, [235](#)
  - LPC\_GPIO, [235](#)
  - LPC\_GPIO\_GROUPINT0\_BASE, [235](#)
  - LPC\_GPIO\_GROUPINT1\_BASE, [235](#)
  - LPC\_GPIO\_PORT\_BASE, [235](#)
  - LPC\_INMUX, [235](#)
  - LPC\_INMUX\_BASE, [235](#)
  - LPC\_IOCON, [235](#)
  - LPC\_IOCON\_BASE, [235](#)
  - LPC\_ISPAP\_BASE, [235](#)
  - LPC\_MBOX, [235](#)
  - LPC\_MBOX\_BASE, [235](#)
  - LPC\_MRT, [236](#)
  - LPC\_MRT\_BASE, [236](#)
  - LPC\_PIN\_INT\_BASE, [236](#)
  - LPC\_PININT, [236](#)
  - LPC\_PMU, [236](#)
  - LPC\_PMU\_BASE, [236](#)
  - LPC\_ROM\_BASE, [236](#)
  - LPC\_RTC, [236](#)
  - LPC\_RTC\_BASE, [236](#)
  - LPC\_SCT, [236](#)
  - LPC\_SCT\_BASE, [236](#)
  - LPC\_SPIFI\_BASE, [236](#)
  - LPC\_SRAM0\_BASE, [237](#)
  - LPC\_SRAM1\_BASE, [237](#)
  - LPC\_SRAM2\_BASE, [237](#)
  - LPC\_SRAMX\_BASE, [237](#)
  - LPC\_SYSCON, [237](#)
  - LPC\_SYSCON\_BASE, [237](#)
  - LPC\_TIMER0, [237](#)
  - LPC\_TIMER0\_BASE, [237](#)
  - LPC\_TIMER1, [237](#)
  - LPC\_TIMER1\_BASE, [237](#)
  - LPC\_TIMER2, [237](#)
  - LPC\_TIMER2\_BASE, [237](#)
  - LPC\_TIMER3, [238](#)
  - LPC\_TIMER3\_BASE, [238](#)
  - LPC\_TIMER4, [238](#)
  - LPC\_TIMER4\_BASE, [238](#)
  - LPC\_USB, [238](#)
  - LPC\_USB\_BASE, [238](#)
  - LPC\_UTICK, [238](#)
  - LPC\_UTICK\_BASE, [238](#)
  - LPC\_WWDT, [238](#)
  - LPC\_WWDT\_BASE, [238](#)
- CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [239](#)
  - Chip\_PININT\_BITSLICE\_CFG\_T, [242](#)
  - Chip\_PININT\_BITSLICE\_T, [242](#)
  - Chip\_PININT\_ClearFallStates, [243](#)
  - Chip\_PININT\_ClearIntStatus, [243](#)
  - Chip\_PININT\_ClearRiseStates, [243](#)
  - Chip\_PININT\_DeInit, [244](#)
  - Chip\_PININT\_DisableIntHigh, [244](#)
  - Chip\_PININT\_DisableIntLow, [244](#)
  - Chip\_PININT\_DisablePatternMatch, [244](#)
  - Chip\_PININT\_DisablePatternMatchRxEv, [245](#)
  - Chip\_PININT\_EnableIntHigh, [245](#)
  - Chip\_PININT\_EnableIntLow, [245](#)
  - Chip\_PININT\_EnablePatternMatch, [245](#)
  - Chip\_PININT\_EnablePatternMatchRxEv, [246](#)

- Chip\_PININT\_GetFallStates, [246](#)
- Chip\_PININT\_GetHighEnabled, [246](#)
- Chip\_PININT\_GetIntStatus, [246](#)
- Chip\_PININT\_GetLowEnabled, [247](#)
- Chip\_PININT\_GetPatternMatchState, [247](#)
- Chip\_PININT\_GetPinMode, [247](#)
- Chip\_PININT\_GetRiseStates, [247](#)
- Chip\_PININT\_Init, [248](#)
- Chip\_PININT\_SELECT\_T, [242](#)
- Chip\_PININT\_SetPatternMatchConfig, [248](#)
- Chip\_PININT\_SetPatternMatchSrc, [248](#)
- Chip\_PININT\_SetPinModeEdge, [249](#)
- Chip\_PININT\_SetPinModeLevel, [249](#)
- PININT\_ISEL\_PMODE\_MASK, [240](#)
- PININT\_PATTERNCONST0, [242](#)
- PININT\_PATTERNCONST1, [242](#)
- PININT\_PATTERNEVENT, [242](#)
- PININT\_PATTERNFALLING, [242](#)
- PININT\_PATTERNHIGH, [242](#)
- PININT\_PATTERNLOW, [242](#)
- PININT\_PATTERNRISING, [242](#)
- PININT\_PATTERNRISINGORFALLING, [242](#)
- PININT\_PMCTRL\_MASK, [240](#)
- PININT\_PMCTRL\_PMATCH\_SEL, [241](#)
- PININT\_PMCTRL\_RXEV\_ENA, [241](#)
- PININT\_SRC\_BITCFG\_MASK, [241](#)
- PININT\_SRC\_BITCFG\_START, [241](#)
- PININT\_SRC\_BITSOURCE\_MASK, [241](#)
- PININT\_SRC\_BITSOURCE\_START, [241](#)
- PININTBITSLICE0, [242](#)
- PININTBITSLICE1, [242](#)
- PININTBITSLICE2, [242](#)
- PININTBITSLICE3, [242](#)
- PININTBITSLICE4, [242](#)
- PININTBITSLICE5, [242](#)
- PININTBITSLICE6, [242](#)
- PININTBITSLICE7, [242](#)
- PININTCH, [241](#)
- PININTCH0, [241](#)
- PININTCH1, [241](#)
- PININTCH2, [241](#)
- PININTCH3, [241](#)
- PININTCH4, [241](#)
- PININTCH5, [242](#)
- PININTCH6, [242](#)
- PININTCH7, [242](#)
- PININTSELECT0, [243](#)
- PININTSELECT1, [243](#)
- PININTSELECT2, [243](#)
- PININTSELECT3, [243](#)
- PININTSELECT4, [243](#)
- PININTSELECT5, [243](#)
- PININTSELECT6, [243](#)
- PININTSELECT7, [243](#)
- CHIP: LPC5411X Power LIBRARY functions, [250](#)
  - Chip\_POWER\_EnterPowerMode, [252](#)
  - Chip\_POWER\_GetROMVersion, [252](#)
  - Chip\_POWER\_SetFROHFRate, [252](#)
  - Chip\_POWER\_SetLowPowerVoltage, [252](#)
  - Chip\_POWER\_SetPLL, [253](#)
  - Chip\_POWER\_SetVoltage, [253](#)
  - LPC5411X\_ROMVER\_0, [251](#)
  - LPC5411X\_ROMVER\_1, [251](#)
  - LPC5411X\_ROMVER\_2, [251](#)
  - POWER\_DEEP\_POWER\_DOWN, [251](#)
  - POWER\_DEEP\_SLEEP, [251](#)
  - POWER\_MODE\_T, [251](#)
  - POWER\_SLEEP, [251](#)
- CHIP: LPC5411X Power Management declarations and functions, [254](#)
  - CHIP\_PMU\_BODRINTVAL\_T, [255](#)
  - CHIP\_PMU\_BODRSTLVL\_T, [255](#)
  - Chip\_PMU\_DisableBODInt, [255](#)
  - Chip\_PMU\_DisableBODReset, [255](#)
  - Chip\_PMU\_EnableBODInt, [255](#)
  - Chip\_PMU\_EnableBODReset, [256](#)
  - Chip\_PMU\_SetBODLevels, [256](#)
  - PMU\_BOD\_INT, [254](#)
  - PMU\_BOD\_RST, [254](#)
  - PMU\_BODINTVAL\_2\_05v, [255](#)
  - PMU\_BODINTVAL\_2\_45v, [255](#)
  - PMU\_BODINTVAL\_2\_75v, [255](#)
  - PMU\_BODINTVAL\_3\_05v, [255](#)
  - PMU\_BODINTVAL\_LVL0, [255](#)
  - PMU\_BODINTVAL\_LVL1, [255](#)
  - PMU\_BODINTVAL\_LVL2, [255](#)
  - PMU\_BODINTVAL\_LVL3, [255](#)
  - PMU\_BODRSTLVL\_0, [255](#)
  - PMU\_BODRSTLVL\_1, [255](#)
  - PMU\_BODRSTLVL\_1\_50V, [255](#)
  - PMU\_BODRSTLVL\_1\_85V, [255](#)
  - PMU\_BODRSTLVL\_2, [255](#)
  - PMU\_BODRSTLVL\_2\_00V, [255](#)
  - PMU\_BODRSTLVL\_2\_30V, [255](#)
  - PMU\_BODRSTLVL\_3, [255](#)
- CHIP: LPC5411X ROM API declarations and functions, [257](#)
  - IAP\_ENTRY\_LOCATION, [257](#)
  - iap\_entry, [257](#)
  - LPC\_ROM\_API, [257](#)
  - LPC\_ROM\_API\_BASE\_LOC, [257](#)
- CHIP: LPC5411X Real Time clock, [258](#)
  - Chip\_RTC\_ClearStatus, [260](#)
  - Chip\_RTC\_DeInit, [260](#)
  - Chip\_RTC\_Disable, [260](#)
  - Chip\_RTC\_Disable1KHZ, [261](#)
  - Chip\_RTC\_DisableOptions, [261](#)
  - Chip\_RTC\_DisableWakeup, [261](#)
  - Chip\_RTC\_Enable, [262](#)
  - Chip\_RTC\_Enable1KHZ, [262](#)
  - Chip\_RTC\_EnableOptions, [262](#)
  - Chip\_RTC\_EnableWakeup, [263](#)
  - Chip\_RTC\_GetAlarm, [263](#)
  - Chip\_RTC\_GetCount, [263](#)
  - Chip\_RTC\_GetStatus, [264](#)
  - Chip\_RTC\_GetWake, [264](#)

- Chip\_RTC\_Init, [264](#)
- Chip\_RTC\_PowerDown, [264](#)
- Chip\_RTC\_PowerUp, [265](#)
- Chip\_RTC\_Reset, [265](#)
- Chip\_RTC\_SetAlarm, [265](#)
- Chip\_RTC\_SetCount, [265](#)
- Chip\_RTC\_SetWake, [267](#)
- RTC\_CTRL\_ALARM1HZ, [259](#)
- RTC\_CTRL\_ALARMDPD\_EN, [259](#)
- RTC\_CTRL\_MASK, [259](#)
- RTC\_CTRL\_RTC1KHZ\_EN, [259](#)
- RTC\_CTRL\_RTC\_EN, [259](#)
- RTC\_CTRL\_RTC\_OSC\_BYPASS, [259](#)
- RTC\_CTRL\_RTC\_OSC\_PD, [259](#)
- RTC\_CTRL\_SWRESET, [260](#)
- RTC\_CTRL\_WAKE1KHZ, [260](#)
- RTC\_CTRL\_WAKEDPD\_EN, [260](#)
- CHIP: LPC5411X SPI driver, [268](#)
  - Chip\_SPI\_ClearCFGRegBits, [280](#)
  - Chip\_SPI\_ClearFIFOCfg, [281](#)
  - Chip\_SPI\_ClearFIFOStatus, [281](#)
  - Chip\_SPI\_ClearStatus, [281](#)
  - Chip\_SPI\_ConfigureSPI, [281](#)
  - Chip\_SPI\_DeInit, [283](#)
  - Chip\_SPI\_Disable, [283](#)
  - Chip\_SPI\_DisableFIFOInts, [283](#)
  - Chip\_SPI\_DisableInts, [283](#)
  - Chip\_SPI\_Enable, [284](#)
  - Chip\_SPI\_EnableFIFOInts, [284](#)
  - Chip\_SPI\_EnableInts, [284](#)
  - Chip\_SPI\_EnableLSBFirst, [284](#)
  - Chip\_SPI\_EnableMSBFirst, [285](#)
  - Chip\_SPI\_EnableSlaveMode, [285](#)
  - Chip\_SPI\_FlushFIFOs, [272](#)
  - Chip\_SPI\_FlushFifos, [285](#)
  - Chip\_SPI\_GetEnabledInts, [285](#)
  - Chip\_SPI\_GetFIFOEnabledInts, [286](#)
  - Chip\_SPI\_GetFIFOPendingInts, [286](#)
  - Chip\_SPI\_GetFIFOStatus, [286](#)
  - Chip\_SPI\_GetFIFOTrigLevel, [286](#)
  - Chip\_SPI\_GetPendingInts, [288](#)
  - Chip\_SPI\_GetStatus, [288](#)
  - Chip\_SPI\_Init, [288](#)
  - Chip\_SPI\_ReadFIFO, [272](#)
  - Chip\_SPI\_ReadFIFOdata, [272](#)
  - Chip\_SPI\_ReadRXData, [289](#)
  - Chip\_SPI\_ReadRawRXFifo, [288](#)
  - Chip\_SPI\_SetCFGRegBits, [289](#)
  - Chip\_SPI\_SetCSPolHigh, [289](#)
  - Chip\_SPI\_SetCSPolLow, [290](#)
  - Chip\_SPI\_SetFIFOCfg, [290](#)
  - Chip\_SPI\_SetFIFOTrigLevel, [290](#)
  - Chip\_SPI\_SetSPIMode, [291](#)
  - Chip\_SPI\_SetTXCTRLData, [291](#)
  - Chip\_SPI\_WriteFIFO, [291](#)
  - Chip\_SPI\_WriteFIFOcd, [272](#)
  - Chip\_SPI\_WriteFIFOdata, [272](#)
  - Chip\_SPI\_WriteTXData, [291](#)
- ROM\_SPI\_CLOCK\_CPHA0\_CPOLO, [280](#)
- ROM\_SPI\_CLOCK\_CPHA0\_CPOL1, [280](#)
- ROM\_SPI\_CLOCK\_CPHA1\_CPOLO, [280](#)
- ROM\_SPI\_CLOCK\_CPHA1\_CPOL1, [280](#)
- ROM\_SPI\_CLOCK\_MODE0, [280](#)
- ROM\_SPI\_CLOCK\_MODE1, [280](#)
- ROM\_SPI\_CLOCK\_MODE2, [280](#)
- ROM\_SPI\_CLOCK\_MODE3, [280](#)
- ROM\_SPI\_CLOCK\_MODE\_T, [280](#)
- SPI\_CFG\_BITMASK, [272](#)
- SPI\_CFG\_CPHA\_FIRST, [272](#)
- SPI\_CFG\_CPHA\_SECOND, [272](#)
- SPI\_CFG\_CPOL\_HI, [272](#)
- SPI\_CFG\_CPOL\_LO, [272](#)
- SPI\_CFG\_LBM\_EN, [273](#)
- SPI\_CFG\_LSB\_FIRST\_EN, [273](#)
- SPI\_CFG\_MASTER\_EN, [273](#)
- SPI\_CFG\_MSB\_FIRST\_EN, [273](#)
- SPI\_CFG\_SLAVE\_EN, [273](#)
- SPI\_CFG\_SPI\_EN, [273](#)
- SPI\_CFG\_SPOL\_HI, [273](#)
- SPI\_CFG\_SPOL\_LO, [273](#)
- SPI\_CFG\_SPOLNUM\_HI, [273](#)
- SPI\_CLOCK\_CPHA0\_CPOL0, [280](#)
- SPI\_CLOCK\_CPHA0\_CPOL1, [280](#)
- SPI\_CLOCK\_CPHA1\_CPOL0, [280](#)
- SPI\_CLOCK\_CPHA1\_CPOL1, [280](#)
- SPI\_CLOCK\_MODE0, [280](#)
- SPI\_CLOCK\_MODE1, [280](#)
- SPI\_CLOCK\_MODE2, [280](#)
- SPI\_CLOCK\_MODE3, [280](#)
- SPI\_CLOCK\_MODE\_T, [280](#)
- SPI\_DIV\_VAL, [273](#)
- SPI\_DLY\_BITMASK, [273](#)
- SPI\_DLY\_FRAME\_DELAY, [273](#)
- SPI\_DLY\_POST\_DELAY, [274](#)
- SPI\_DLY\_PRE\_DELAY, [274](#)
- SPI\_DLY\_TRANSFER\_DELAY, [274](#)
- SPI\_FIFO\_DEPTH, [274](#)
- SPI\_FIFOCFG\_DMARX, [274](#)
- SPI\_FIFOCFG\_DMATX, [274](#)
- SPI\_FIFOCFG\_EMPTYRX, [274](#)
- SPI\_FIFOCFG\_EMPTYTX, [274](#)
- SPI\_FIFOCFG\_ENABLERX, [274](#)
- SPI\_FIFOCFG\_ENABLETX, [274](#)
- SPI\_FIFOCFG\_WAKERX, [275](#)
- SPI\_FIFOCFG\_WAKETX, [275](#)
- SPI\_FIFPOINT\_BITMASK, [275](#)
- SPI\_FIFPOINT\_PERINT, [275](#)
- SPI\_FIFPOINT\_RXERR, [275](#)
- SPI\_FIFPOINT\_RXLVL, [275](#)
- SPI\_FIFPOINT\_TXERR, [275](#)
- SPI\_FIFPOINT\_TXLVL, [275](#)
- SPI\_FIFOSTAT\_BITMASK, [275](#)
- SPI\_FIFOSTAT\_PERINT, [275](#)
- SPI\_FIFOSTAT\_RXERR, [276](#)
- SPI\_FIFOSTAT\_RXFULL, [276](#)
- SPI\_FIFOSTAT\_RXLVL, [276](#)

- SPI\_FIFOSTAT\_RXNOTEMPTY, [276](#)
- SPI\_FIFOSTAT\_TXEMPTY, [276](#)
- SPI\_FIFOSTAT\_TXERR, [276](#)
- SPI\_FIFOSTAT\_TXLVL, [276](#)
- SPI\_FIFOSTAT\_TXNOTFULL, [276](#)
- SPI\_FIFOTRIG\_BITMASK, [276](#)
- SPI\_FIFOTRIG\_RXLVL, [276](#)
- SPI\_FIFOTRIG\_RXLVL\_DEFAULT, [276](#)
- SPI\_FIFOTRIG\_RXLVLENA, [277](#)
- SPI\_FIFOTRIG\_TXLVL, [277](#)
- SPI\_FIFOTRIG\_TXLVL\_DEFAULT, [277](#)
- SPI\_FIFOTRIG\_TXLVLENA, [277](#)
- SPI\_INT\_BITMASK, [277](#)
- SPI\_INT\_MSTIDLE, [277](#)
- SPI\_INT\_SSAEN, [277](#)
- SPI\_INT\_SSDEN, [277](#)
- SPI\_RXDAT\_BITMASK, [277](#)
- SPI\_RXDAT\_DATA, [277](#)
- SPI\_RXDAT\_RXSSELN, [278](#)
- SPI\_RXDAT\_RXSSELN\_ACTIVE, [278](#)
- SPI\_RXDAT\_SOT, [278](#)
- SPI\_STAT\_BITMASK, [278](#)
- SPI\_STAT\_EOT, [278](#)
- SPI\_STAT\_MSTIDLE, [278](#)
- SPI\_STAT\_SSA, [278](#)
- SPI\_STAT\_SSD, [278](#)
- SPI\_STAT\_STALLED, [278](#)
- SPI\_TXDAT\_ASSERT\_SSEL, [278](#)
- SPI\_TXDAT\_ASSERTNUM\_SSEL, [278](#)
- SPI\_TXDAT\_BITMASK, [278](#)
- SPI\_TXDAT\_CTRLMASK, [279](#)
- SPI\_TXDAT\_DATA, [279](#)
- SPI\_TXDAT\_DEASSERT\_ALL, [279](#)
- SPI\_TXDAT\_DEASSERT\_SSEL, [279](#)
- SPI\_TXDAT\_DEASSERTNUM\_SSEL, [279](#)
- SPI\_TXDAT\_EOF, [279](#)
- SPI\_TXDAT\_EOT, [279](#)
- SPI\_TXDAT\_FLEN, [279](#)
- SPI\_TXDAT\_FLENMASK, [279](#)
- SPI\_TXDAT\_RXIGNORE, [279](#)
- CHIP: LPC5411X SPI master driver, [294](#)
  - Chip\_SPIM\_DelayConfig, [295](#)
  - Chip\_SPIM\_DisableLoopBack, [296](#)
  - Chip\_SPIM\_EnableLoopBack, [296](#)
  - Chip\_SPIM\_ForceEndOfTransfer, [296](#)
  - Chip\_SPIM\_GetClockRate, [296](#)
  - Chip\_SPIM\_SetClockRate, [297](#)
  - Chip\_SPIM\_Xfer, [297](#)
  - Chip\_SPIM\_XferBlocking, [298](#)
  - Chip\_SPIM\_XferFIFO, [298](#)
  - Chip\_SPIM\_XferHandler, [299](#)
  - SPI\_XFER\_STATE\_BUSY, [295](#)
  - SPI\_XFER\_STATE\_DONE, [295](#)
  - SPI\_XFER\_STATE\_ERROR, [295](#)
  - SPI\_XFER\_STATE\_IDLE, [295](#)
  - SPI\_XFER\_STATE\_STALL, [295](#)
  - SPIM\_EVENT\_DONE, [295](#)
  - SPIM\_EVENT\_ERROR, [295](#)
  - SPIM\_EVENT\_ERRORRX, [295](#)
  - SPIM\_EVENT\_ERRORTX, [295](#)
  - SPIM\_EVENT\_T, [295](#)
  - SPIM\_EVENT\_WAIT, [295](#)
  - SPIM\_XFER\_OPT\_DMA, [295](#)
  - SPIM\_XFER\_OPT\_FRAME\_ASSERT, [295](#)
  - SPIM\_XFER\_OPT\_FRAME\_DLY, [295](#)
  - SPIM\_XFER\_STATE\_T, [295](#)
- CHIP: LPC5411X SPI slave driver, [300](#)
  - Chip\_SPIS\_DisableInts, [300](#)
  - Chip\_SPIS\_EnableInts, [301](#)
  - Chip\_SPIS\_Init, [301](#)
  - Chip\_SPIS\_LoadFIFO, [301](#)
  - Chip\_SPIS\_ReadFIFO, [302](#)
  - Chip\_SPIS\_XferHandler, [302](#)
  - SPIS\_EVENT\_DONE, [300](#)
  - SPIS\_EVENT\_ERRORRX, [300](#)
  - SPIS\_EVENT\_ERRORTX, [300](#)
  - SPIS\_EVENT\_SASSERT, [300](#)
  - SPIS\_EVENT\_SDEASSERT, [300](#)
  - SPIS\_EVENT\_T, [300](#)
  - SPIS\_EVENT\_THRESHOLD, [300](#)
- CHIP: LPC5411X State Configurable Timer driver, [307](#)
  - CHIP\_SCT\_EVENT\_T, [316](#)
  - CHIP\_SCT\_MATCH\_REG\_T, [317](#)
  - CONFIG\_SCT\_nEV, [309](#)
  - CONFIG\_SCT\_nIN, [309](#)
  - CONFIG\_SCT\_nOU, [310](#)
  - CONFIG\_SCT\_nRG, [310](#)
  - COUNTUP\_TO, [310](#)
  - COUNTUP\_TO\_LIMIT\_THEN\_CLEAR\_TO\_ZERO, [310](#)
  - Chip\_SCT\_ClearControl, [317](#)
  - Chip\_SCT\_ClearEventFlag, [318](#)
  - Chip\_SCT\_Config, [318](#)
  - Chip\_SCT\_DeInit, [318](#)
  - Chip\_SCT\_DisableEventInt, [318](#)
  - Chip\_SCT\_EnableEventInt, [319](#)
  - Chip\_SCT\_EventControl, [319](#)
  - Chip\_SCT\_EventStateMask, [319](#)
  - Chip\_SCT\_Init, [319](#)
  - Chip\_SCT\_Limit, [321](#)
  - Chip\_SCT\_SetClrControl, [321](#)
  - Chip\_SCT\_SetConflictResolution, [321](#)
  - Chip\_SCT\_SetControl, [322](#)
  - Chip\_SCT\_SetCount, [322](#)
  - Chip\_SCT\_SetCountH, [322](#)
  - Chip\_SCT\_SetCountL, [322](#)
  - Chip\_SCT\_SetMatchCount, [323](#)
  - Chip\_SCT\_SetMatchReload, [323](#)
  - SCT\_CONFIG\_16BIT\_COUNTER, [310](#)
  - SCT\_CONFIG\_32BIT\_COUNTER, [310](#)
  - SCT\_CONFIG\_AUTOLIMIT\_H, [310](#)
  - SCT\_CONFIG\_AUTOLIMIT\_L, [311](#)
  - SCT\_CONFIG\_AUTOLIMIT\_U, [311](#)
  - SCT\_CONFIG\_CKSEL\_FALLING\_IN\_0, [311](#)
  - SCT\_CONFIG\_CKSEL\_FALLING\_IN\_1, [311](#)
  - SCT\_CONFIG\_CKSEL\_FALLING\_IN\_2, [311](#)

- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_3, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_4, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_5, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_6, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_7, [311](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_0, [311](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_1, [311](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_2, [312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_3, [312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_4, [312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_5, [312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_6, [312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_7, [312](#)
- SCT\_CONFIG\_CLKMODE\_BUSCLK, [312](#)
- SCT\_CONFIG\_CLKMODE\_INCLK, [312](#)
- SCT\_CONFIG\_CLKMODE\_INEDGECLK, [312](#)
- SCT\_CONFIG\_CLKMODE\_PRESCALED\_SCT←  
\_INPUT, [312](#)
- SCT\_CONFIG\_CLKMODE\_PRESCALED\_SYS←  
CLK, [312](#)
- SCT\_CONFIG\_CLKMODE\_SCT\_INPUT, [313](#)
- SCT\_CONFIG\_CLKMODE\_SCTCLK, [313](#)
- SCT\_CONFIG\_CLKMODE\_SYSCLK, [313](#)
- SCT\_CONFIG\_NORELOADH, [313](#)
- SCT\_CONFIG\_NORELOADL\_U, [313](#)
- SCT\_CTRL\_BIDIR\_H, [313](#)
- SCT\_CTRL\_BIDIR\_L, [313](#)
- SCT\_CTRL\_CLRCTR\_H, [313](#)
- SCT\_CTRL\_CLRCTR\_L, [313](#)
- SCT\_CTRL\_HALT\_H, [314](#)
- SCT\_CTRL\_HALT\_L, [314](#)
- SCT\_CTRL\_PRE\_H, [314](#)
- SCT\_CTRL\_PRE\_L, [314](#)
- SCT\_CTRL\_STOP\_H, [314](#)
- SCT\_CTRL\_STOP\_L, [314](#)
- SCT\_EV\_CTRL\_COMBMODE\_AND, [314](#)
- SCT\_EV\_CTRL\_COMBMODE\_IO, [314](#)
- SCT\_EV\_CTRL\_COMBMODE\_MATCH, [314](#)
- SCT\_EV\_CTRL\_COMBMODE\_OR, [314](#)
- SCT\_EV\_CTRL\_DIRECTION\_DOWN, [315](#)
- SCT\_EV\_CTRL\_DIRECTION\_INDEPENDENT,  
[315](#)
- SCT\_EV\_CTRL\_DIRECTION\_UP, [315](#)
- SCT\_EV\_CTRL\_HEVENT\_H, [315](#)
- SCT\_EV\_CTRL\_HEVENT\_L, [315](#)
- SCT\_EV\_CTRL\_IOCOND\_FALL, [315](#)
- SCT\_EV\_CTRL\_IOCOND\_HIGH, [315](#)
- SCT\_EV\_CTRL\_IOCOND\_LOW, [315](#)
- SCT\_EV\_CTRL\_IOCOND\_RISE, [315](#)
- SCT\_EV\_CTRL\_IOSEL, [315](#)
- SCT\_EV\_CTRL\_MATCHMEM, [315](#)
- SCT\_EV\_CTRL\_MATCHSEL, [315](#)
- SCT\_EV\_CTRL\_OUTSEL\_INPUT, [316](#)
- SCT\_EV\_CTRL\_OUTSEL\_OUTPUT, [316](#)
- SCT\_EV\_CTRL\_STATELD, [316](#)
- SCT\_EV\_CTRL\_STATEV, [316](#)
- SCT\_EVT\_0, [316](#)
- SCT\_EVT\_1, [316](#)
- SCT\_EVT\_10, [317](#)
- SCT\_EVT\_11, [317](#)
- SCT\_EVT\_12, [317](#)
- SCT\_EVT\_13, [317](#)
- SCT\_EVT\_14, [317](#)
- SCT\_EVT\_15, [317](#)
- SCT\_EVT\_2, [316](#)
- SCT\_EVT\_3, [316](#)
- SCT\_EVT\_4, [316](#)
- SCT\_EVT\_5, [316](#)
- SCT\_EVT\_6, [317](#)
- SCT\_EVT\_7, [317](#)
- SCT\_EVT\_8, [317](#)
- SCT\_EVT\_9, [317](#)
- SCT\_MATCH\_0, [317](#)
- SCT\_MATCH\_1, [317](#)
- SCT\_MATCH\_10, [317](#)
- SCT\_MATCH\_11, [317](#)
- SCT\_MATCH\_12, [317](#)
- SCT\_MATCH\_13, [317](#)
- SCT\_MATCH\_14, [317](#)
- SCT\_MATCH\_15, [317](#)
- SCT\_MATCH\_2, [317](#)
- SCT\_MATCH\_3, [317](#)
- SCT\_MATCH\_4, [317](#)
- SCT\_MATCH\_5, [317](#)
- SCT\_MATCH\_6, [317](#)
- SCT\_MATCH\_7, [317](#)
- SCT\_MATCH\_8, [317](#)
- SCT\_MATCH\_9, [317](#)
- SCT\_RES\_CLEAR\_OUTPUT, [316](#)
- SCT\_RES\_NOCHANGE, [316](#)
- SCT\_RES\_SET\_OUTPUT, [316](#)
- SCT\_RES\_TOGGLE\_OUTPUT, [316](#)
- CHIP: LPC5411X State Configurable Timer PWM driver,  
[303](#)
- Chip\_SCTPWM\_GetDutyCycle, [303](#)
- Chip\_SCTPWM\_GetTicksPerCycle, [303](#)
- Chip\_SCTPWM\_Init, [304](#)
- Chip\_SCTPWM\_PercentageToTicks, [304](#)
- Chip\_SCTPWM\_SetDutyCycle, [304](#)
- Chip\_SCTPWM\_SetOutPin, [305](#)
- Chip\_SCTPWM\_SetRate, [305](#)
- Chip\_SCTPWM\_Start, [305](#)
- Chip\_SCTPWM\_Stop, [306](#)
- CHIP: LPC5411X support functions, [385](#)
- Chip\_SetupExtInClocking, [385](#)
- Chip\_SetupFROClocking, [385](#)
- Chip\_SetupIrcClocking, [385](#)
- Chip\_SystemInit, [386](#)
- Chip\_USB\_Init, [386](#)
- Chip\_USB\_TrimOff, [386](#)
- SystemCoreClock, [387](#)
- SystemCoreClockUpdate, [386](#)
- CHIP: LPC5411X System and Control Driver, [324](#)
- CHIP\_SYSCON\_BOOT\_MODE\_REMAP\_T, [330](#)
- CHIP\_SYSCON\_PERIPH\_RESET\_T, [331](#)
- CHIP\_SYSCON\_WAKEUP\_T, [331](#)



- Chip\_SYSCON\_ClearSystemRSTStatus, 333
- Chip\_SYSCON\_DisableAutoClocking, 333
- Chip\_SYSCON\_DisableNMISource, 333
- Chip\_SYSCON\_DisableWakeup, 334
- Chip\_SYSCON\_Enable\_ASYNC\_Syscon, 334
- Chip\_SYSCON\_EnableAutoClocking, 334
- Chip\_SYSCON\_EnableNMISource, 334
- Chip\_SYSCON\_EnableWakeup, 335
- Chip\_SYSCON\_GetCompFreqMeas, 335
- Chip\_SYSCON\_GetDeviceID, 335
- Chip\_SYSCON\_GetMemoryMap, 335
- Chip\_SYSCON\_GetPORPIOStatus, 335
- Chip\_SYSCON\_GetPowerStates, 337
- Chip\_SYSCON\_GetRawFreqMeasCapval, 337
- Chip\_SYSCON\_GetResetPIOStatus, 337
- Chip\_SYSCON\_GetSystemRSTStatus, 337
- Chip\_SYSCON\_IsFreqMeasComplete, 338
- Chip\_SYSCON\_Map, 338
- Chip\_SYSCON\_PeriphReset, 338
- Chip\_SYSCON\_PowerDown, 338
- Chip\_SYSCON\_PowerUp, 339
- Chip\_SYSCON\_SetFLASHAccess, 339
- Chip\_SYSCON\_SetNMISource, 339
- Chip\_SYSCON\_SetSYSTCKCAL, 339
- Chip\_SYSCON\_SetUSARTFRGCtrl, 340
- Chip\_SYSCON\_StartFreqMeas, 340
- FLASHTIM\_20MHZ\_CPU, 333
- REMAP\_BOOT\_LOADER\_MODE, 330
- REMAP\_USER\_FLASH\_MODE, 331
- REMAP\_USER\_RAM\_MODE, 330
- RESET\_ADC, 331
- RESET\_ADC0, 331
- RESET\_CRC, 331
- RESET\_DMA, 331
- RESET\_DMIC, 331
- RESET\_FLASH, 331
- RESET\_FLEXCOMM0, 331
- RESET\_FLEXCOMM1, 331
- RESET\_FLEXCOMM2, 331
- RESET\_FLEXCOMM3, 331
- RESET\_FLEXCOMM4, 331
- RESET\_FLEXCOMM5, 331
- RESET\_FLEXCOMM6, 331
- RESET\_FLEXCOMM7, 331
- RESET\_FMC, 331
- RESET\_GINT, 331
- RESET\_GPIO0, 331
- RESET\_GPIO1, 331
- RESET\_IOCON, 331
- RESET\_MRT, 331
- RESET\_MUX, 331
- RESET\_PINT, 331
- RESET\_SCT, 331
- RESET\_SCT0, 331
- RESET\_SPIFI, 331
- RESET\_TIMER0, 331
- RESET\_TIMER1, 331
- RESET\_TIMER2, 331
- RESET\_TIMER3, 331
- RESET\_TIMER4, 331
- RESET\_USB, 331
- RESET\_UTICK, 331
- RESET\_WWDT, 331
- SYSCON\_AUTOCGOR\_MASK, 326
- SYSCON\_AUTOCGOR\_RAM0X, 326
- SYSCON\_AUTOCGOR\_RAM1, 327
- SYSCON\_AUTOCGOR\_RAM2, 327
- SYSCON\_FLASH\_1CYCLE, 333
- SYSCON\_FLASH\_2CYCLE, 333
- SYSCON\_FLASH\_3CYCLE, 333
- SYSCON\_FLASH\_4CYCLE, 333
- SYSCON\_FLASH\_5CYCLE, 333
- SYSCON\_FLASH\_6CYCLE, 333
- SYSCON\_FLASH\_7CYCLE, 333
- SYSCON\_FLASH\_8CYCLE, 333
- SYSCON\_FLASHTIM\_T, 332
- SYSCON\_FROCTRL\_HSPDCLK, 327
- SYSCON\_FROCTRL\_MASK, 327
- SYSCON\_FROCTRL\_SEL96MHZ, 327
- SYSCON\_FROCTRL\_USBCLKADJ, 327
- SYSCON\_FROCTRL\_USBMODCHG, 327
- SYSCON\_FROCTRL\_WRTRIM, 327
- SYSCON\_NMISRC\_M0\_ENABLE, 327
- SYSCON\_NMISRC\_M4\_ENABLE, 328
- SYSCON\_PDRUNCFG\_LP\_VDDFLASH, 328
- SYSCON\_PDRUNCFG\_PD\_ADC0, 328
- SYSCON\_PDRUNCFG\_PD\_BOD\_INTR, 328
- SYSCON\_PDRUNCFG\_PD\_BOD\_RST, 328
- SYSCON\_PDRUNCFG\_PD\_FLASH, 328
- SYSCON\_PDRUNCFG\_PD\_FRO, 328
- SYSCON\_PDRUNCFG\_PD\_ROM, 328
- SYSCON\_PDRUNCFG\_PD\_SRAM0, 328
- SYSCON\_PDRUNCFG\_PD\_SRAM1, 329
- SYSCON\_PDRUNCFG\_PD\_SRAM2, 329
- SYSCON\_PDRUNCFG\_PD\_SRAMX, 329
- SYSCON\_PDRUNCFG\_PD\_SYS\_PLL, 329
- SYSCON\_PDRUNCFG\_PD\_TS, 329
- SYSCON\_PDRUNCFG\_PD\_USB\_PHY, 329
- SYSCON\_PDRUNCFG\_PD\_VDDA\_ENA, 329
- SYSCON\_PDRUNCFG\_PD\_VDDFLASH, 329
- SYSCON\_PDRUNCFG\_PD\_VDDHV\_ENA, 329
- SYSCON\_PDRUNCFG\_PD\_VREFP, 330
- SYSCON\_PDRUNCFG\_PD\_WDT\_OSC, 330
- SYSCON\_RST\_BOD, 330
- SYSCON\_RST\_EXTRST, 330
- SYSCON\_RST\_POR, 330
- SYSCON\_RST\_SYSRST, 330
- SYSCON\_RST\_WDT, 330
- SYSCON\_STARTER\_ADC0\_SEQA, 332
- SYSCON\_STARTER\_ADC0\_SEQB, 332
- SYSCON\_STARTER\_ADC0\_THCMP, 332
- SYSCON\_STARTER\_DMA, 332
- SYSCON\_STARTER\_DMIC, 332
- SYSCON\_STARTER\_FLEXCOMM0, 332
- SYSCON\_STARTER\_FLEXCOMM1, 332
- SYSCON\_STARTER\_FLEXCOMM2, 332

- SYSCON\_STARTER\_FLEXCOMM3, 332
- SYSCON\_STARTER\_FLEXCOMM4, 332
- SYSCON\_STARTER\_FLEXCOMM5, 332
- SYSCON\_STARTER\_FLEXCOMM6, 332
- SYSCON\_STARTER\_FLEXCOMM7, 332
- SYSCON\_STARTER\_GINT0, 332
- SYSCON\_STARTER\_GINT1, 332
- SYSCON\_STARTER\_HWVAD, 332
- SYSCON\_STARTER\_MAILBOX, 332
- SYSCON\_STARTER\_MRT, 332
- SYSCON\_STARTER\_PINT0, 332
- SYSCON\_STARTER\_PINT1, 332
- SYSCON\_STARTER\_PINT2, 332
- SYSCON\_STARTER\_PINT3, 332
- SYSCON\_STARTER\_PINT4, 332
- SYSCON\_STARTER\_PINT5, 332
- SYSCON\_STARTER\_PINT6, 332
- SYSCON\_STARTER\_PINT7, 332
- SYSCON\_STARTER\_RESERVED0, 332
- SYSCON\_STARTER\_RTC, 332
- SYSCON\_STARTER\_SCT0, 332
- SYSCON\_STARTER\_TIMER0, 332
- SYSCON\_STARTER\_TIMER1, 332
- SYSCON\_STARTER\_TIMER2, 332
- SYSCON\_STARTER\_TIMER3, 332
- SYSCON\_STARTER\_TIMER4, 332
- SYSCON\_STARTER\_USB, 332
- SYSCON\_STARTER\_USBNEEDCLK, 332
- SYSCON\_STARTER\_UTICK, 332
- SYSCON\_STARTER\_WWDT\_BOD, 332
- CHIP: LPC5411X UART Driver, 341
  - Chip\_UART\_AutoBaud, 353
  - Chip\_UART\_ClearFIFOCfg, 354
  - Chip\_UART\_ClearFIFOStatus, 354
  - Chip\_UART\_ClearStatus, 354
  - Chip\_UART\_ConfigDMA, 356
  - Chip\_UART\_ConfigData, 354
  - Chip\_UART\_DeInit, 356
  - Chip\_UART\_Disable, 356
  - Chip\_UART\_DisableFIFOInts, 356
  - Chip\_UART\_Enable, 358
  - Chip\_UART\_EnableFIFOInts, 358
  - Chip\_UART\_FlushFIFOs, 358
  - Chip\_UART\_GetFIFOEnabledInts, 358
  - Chip\_UART\_GetFIFOPendingInts, 359
  - Chip\_UART\_GetFIFOStatus, 359
  - Chip\_UART\_GetIntStatus, 359
  - Chip\_UART\_GetIntsEnabled, 359
  - Chip\_UART\_GetStatus, 361
  - Chip\_UART\_IRQHandlerDMA, 362
  - Chip\_UART\_IRQHandlerRB, 362
  - Chip\_UART\_Init, 361
  - Chip\_UART\_IntDisable, 361
  - Chip\_UART\_IntEnable, 362
  - Chip\_UART\_RXIntHandlerRB, 365
  - Chip\_UART\_Read, 363
  - Chip\_UART\_ReadBlocking, 363
  - Chip\_UART\_ReadByte, 363
  - Chip\_UART\_ReadRB, 365
  - Chip\_UART\_Send, 365
  - Chip\_UART\_SendBlocking, 366
  - Chip\_UART\_SendByte, 366
  - Chip\_UART\_SendRB, 366
  - Chip\_UART\_SetBaud, 368
  - Chip\_UART\_SetFIFOCfg, 368
  - Chip\_UART\_SetFIFOTrigLevel, 368
  - Chip\_UART\_TXDisable, 369
  - Chip\_UART\_TXEnable, 369
  - Chip\_UART\_TXIntHandlerRB, 369
  - ECHO\_DIS, 344
  - ECHO\_EN, 344
  - UART\_CFG\_AUTOADDR, 344
  - UART\_CFG\_BITMASK, 344
  - UART\_CFG\_CLKPOL, 345
  - UART\_CFG\_CTSEN, 345
  - UART\_CFG\_DATALEN\_7, 345
  - UART\_CFG\_DATALEN\_8, 345
  - UART\_CFG\_DATALEN\_9, 345
  - UART\_CFG\_ENABLE, 345
  - UART\_CFG\_IOMODE, 345
  - UART\_CFG\_LINMODE, 345
  - UART\_CFG\_LOOP, 345
  - UART\_CFG\_MODE32K, 346
  - UART\_CFG\_OEPOL, 346
  - UART\_CFG\_OESEL, 346
  - UART\_CFG\_OETA, 346
  - UART\_CFG\_PARITY\_EVEN, 346
  - UART\_CFG\_PARITY\_NONE, 346
  - UART\_CFG\_PARITY\_ODD, 346
  - UART\_CFG\_RXPOL, 346
  - UART\_CFG\_STOPLEN\_1, 346
  - UART\_CFG\_STOPLEN\_2, 347
  - UART\_CFG\_SYNCEN, 347
  - UART\_CFG\_SYNCMST, 347
  - UART\_CFG\_TXPOL, 347
  - UART\_CTRL\_ADDRDET, 347
  - UART\_CTRL\_AUTOBAUD, 347
  - UART\_CTRL\_CC, 347
  - UART\_CTRL\_CLRCONRX, 347
  - UART\_CTRL\_TXBRKEN, 347
  - UART\_CTRL\_TXDIS, 348
  - UART\_FIFO\_DEPTH, 348
  - UART\_FIFOCFG\_BITMASK, 348
  - UART\_FIFOCFG\_DMARX, 348
  - UART\_FIFOCFG\_DMATX, 348
  - UART\_FIFOCFG\_EMPTYRX, 348
  - UART\_FIFOCFG\_EMPTYTX, 348
  - UART\_FIFOCFG\_ENABLERX, 348
  - UART\_FIFOCFG\_ENABLETX, 348
  - UART\_FIFOCFG\_WAKERX, 349
  - UART\_FIFOCFG\_WAKETX, 349
  - UART\_FIFPOINT\_BITMASK, 349
  - UART\_FIFPOINT\_PERINT, 349
  - UART\_FIFPOINT\_RXERR, 349
  - UART\_FIFPOINT\_RXLVL, 349
  - UART\_FIFPOINT\_TXERR, 349



- UART\_FIFOINT\_TXLVL, [349](#)
- UART\_FIFOSTAT\_BITMASK, [349](#)
- UART\_FIFOSTAT\_PERIPH, [349](#)
- UART\_FIFOSTAT\_RXERR, [350](#)
- UART\_FIFOSTAT\_RXFULL, [350](#)
- UART\_FIFOSTAT\_RXLVL, [350](#)
- UART\_FIFOSTAT\_RXNOTEMPTY, [350](#)
- UART\_FIFOSTAT\_TXEMPTY, [350](#)
- UART\_FIFOSTAT\_TXERR, [350](#)
- UART\_FIFOSTAT\_TXLVL, [350](#)
- UART\_FIFOSTAT\_TXNOTFULL, [350](#)
- UART\_FIFOTRIG\_BITMASK, [350](#)
- UART\_FIFOTRIG\_RXLVL, [351](#)
- UART\_FIFOTRIG\_RXLVLENA, [351](#)
- UART\_FIFOTRIG\_TXLVL, [351](#)
- UART\_FIFOTRIG\_TXLVLENA, [351](#)
- UART\_INT\_ABERR, [351](#)
- UART\_INT\_DELTACTS, [351](#)
- UART\_INT\_DELTARXBRK, [351](#)
- UART\_INT\_FRAMERR, [351](#)
- UART\_INT\_PARITYERR, [351](#)
- UART\_INT\_RXNOISE, [352](#)
- UART\_INT\_START, [352](#)
- UART\_INT\_TXDIS, [352](#)
- UART\_INT\_TXIDLE, [352](#)
- UART\_STAT\_ABERR, [352](#)
- UART\_STAT\_CTS, [352](#)
- UART\_STAT\_DELTACTS, [352](#)
- UART\_STAT\_DELTARXBRK, [352](#)
- UART\_STAT\_FRM\_ERRINT, [352](#)
- UART\_STAT\_PAR\_ERRINT, [353](#)
- UART\_STAT\_RXBRK, [353](#)
- UART\_STAT\_RXIDLE, [353](#)
- UART\_STAT\_RXNOISEINT, [353](#)
- UART\_STAT\_START, [353](#)
- UART\_STAT\_TXDISINT, [353](#)
- UART\_STAT\_TXIDLE, [353](#)
- CHIP: LPC5411X Windowed Watchdog driver, [371](#)
  - Chip\_WWDT\_ClearStatusFlag, [372](#)
  - Chip\_WWDT\_DeInit, [373](#)
  - Chip\_WWDT\_Feed, [373](#)
  - Chip\_WWDT\_GetCurrentCount, [373](#)
  - Chip\_WWDT\_GetStatus, [374](#)
  - Chip\_WWDT\_GetWarning, [374](#)
  - Chip\_WWDT\_GetWindow, [374](#)
  - Chip\_WWDT\_Init, [374](#)
  - Chip\_WWDT\_SetOption, [375](#)
  - Chip\_WWDT\_SetTimeOut, [375](#)
  - Chip\_WWDT\_SetWarning, [375](#)
  - Chip\_WWDT\_SetWindow, [376](#)
  - Chip\_WWDT\_Start, [376](#)
  - Chip\_WWDT\_UnsetOption, [376](#)
- WWDT\_WDMOD\_BITMASK, [372](#)
- WWDT\_WDMOD\_LOCK, [372](#)
- WWDT\_WDMOD\_WDEN, [372](#)
- WWDT\_WDMOD\_WDINT, [372](#)
- WWDT\_WDMOD\_WDPROTECT, [372](#)
- WWDT\_WDMOD\_WDRESET, [372](#)
- WWDT\_WDMOD\_WDTOF, [372](#)
- CHIP: LPC5411x Chip driver build time options, [388](#)
  - ExtClockIn, [388](#)
- CHIP: LPC5411x I2C driver, [389](#)
  - CLKDIV, [414](#)
  - Chip\_I2C\_ClearInt, [411](#)
  - Chip\_I2C\_DeInit, [412](#)
  - Chip\_I2C\_DisableInt, [412](#)
  - Chip\_I2C\_EnableInt, [412](#)
  - Chip\_I2C\_GetClockDiv, [412](#)
  - Chip\_I2C\_GetPendingInt, [413](#)
  - Chip\_I2C\_Init, [413](#)
  - Chip\_I2C\_SetClockDiv, [413](#)
  - I2C\_CFG\_MASK, [393](#)
  - I2C\_CFG\_MONCLKSTR, [393](#)
  - I2C\_CFG\_MONEN, [393](#)
  - I2C\_CFG\_MSTEN, [393](#)
  - I2C\_CFG\_SLVEN, [393, 394](#)
  - I2C\_CFG\_TIMEOUTEN, [394](#)
  - I2C\_INTENCLR\_EVENTTIMEOUT, [394](#)
  - I2C\_INTENCLR\_MONIDLE, [394](#)
  - I2C\_INTENCLR\_MONOV, [394](#)
  - I2C\_INTENCLR\_MONRDY, [395](#)
  - I2C\_INTENCLR\_MSTPENDING, [395](#)
  - I2C\_INTENCLR\_MSTRARBLOSS, [395](#)
  - I2C\_INTENCLR\_MSTSTSPERR, [395](#)
  - I2C\_INTENCLR\_SCLTIMEOUT, [395, 396](#)
  - I2C\_INTENCLR\_SLVDESEL, [396](#)
  - I2C\_INTENCLR\_SLVNOTSTR, [396](#)
  - I2C\_INTENCLR\_SLVPENDING, [396](#)
  - I2C\_INTENSET\_EVENTTIMEOUT, [396](#)
  - I2C\_INTENSET\_MONIDLE, [397](#)
  - I2C\_INTENSET\_MONOV, [397](#)
  - I2C\_INTENSET\_MONRDY, [397](#)
  - I2C\_INTENSET\_MSTPENDING, [397](#)
  - I2C\_INTENSET\_MSTRARBLOSS, [397, 398](#)
  - I2C\_INTENSET\_MSTSTSPERR, [398](#)
  - I2C\_INTENSET\_SCLTIMEOUT, [398](#)
  - I2C\_INTENSET\_SLVDESEL, [398](#)
  - I2C\_INTENSET\_SLVNOTSTR, [398](#)
  - I2C\_INTENSET\_SLVPENDING, [399](#)
  - I2C\_INTSTAT\_EVENTTIMEOUT, [399](#)
  - I2C\_INTSTAT\_MONIDLE, [399](#)
  - I2C\_INTSTAT\_MONOV, [399](#)
  - I2C\_INTSTAT\_MONRDY, [399, 400](#)
  - I2C\_INTSTAT\_MSTPENDING, [400](#)
  - I2C\_INTSTAT\_MSTRARBLOSS, [400](#)
  - I2C\_INTSTAT\_MSTSTSPERR, [400](#)
  - I2C\_INTSTAT\_SCLTIMEOUT, [400](#)
  - I2C\_INTSTAT\_SLVDESEL, [401](#)
  - I2C\_INTSTAT\_SLVNOTSTR, [401](#)
  - I2C\_INTSTAT\_SLVPENDING, [401](#)
  - I2C\_MONRXDAT\_DATA, [401](#)
  - I2C\_MONRXDAT\_MONNACK, [401, 402](#)
  - I2C\_MONRXDAT\_MONRESTART, [402](#)
  - I2C\_MONRXDAT\_MONSTART, [402](#)
  - I2C\_MSTCTL\_MSTCONTINUE, [402](#)
  - I2C\_MSTCTL\_MSTDMA, [402](#)

- I2C\_MSTCTL\_MSTSTART, [403](#)
- I2C\_MSTCTL\_MSTSTOP, [403](#)
- I2C\_MSTDAT\_DATAMASK, [403](#)
- I2C\_MSTTIME\_MSTSCLHIGH, [403](#)
- I2C\_MSTTIME\_MSTSCLLOW, [403](#), [404](#)
- I2C\_SLVADR\_MASK, [404](#)
- I2C\_SLVADR\_SADISABLE, [404](#)
- I2C\_SLVADR\_SLVADR, [404](#)
- I2C\_SLVCTL\_SLVCONTINUE, [404](#)
- I2C\_SLVCTL\_SLVDMA, [405](#)
- I2C\_SLVCTL\_SLVNACK, [405](#)
- I2C\_SLVDAT\_DATAMASK, [405](#)
- I2C\_SLVQUAL\_QUALMODE0, [405](#)
- I2C\_SLVQUAL\_SLVQUAL0, [405](#), [406](#)
- I2C\_STAT\_EVENTTIMEOUT, [406](#)
- I2C\_STAT\_MONACTIVE, [406](#)
- I2C\_STAT\_MONIDLE, [406](#)
- I2C\_STAT\_MONOV, [406](#)
- I2C\_STAT\_MONRDY, [407](#)
- I2C\_STAT\_MSTCODE\_IDLE, [407](#)
- I2C\_STAT\_MSTCODE\_NACKADR, [407](#)
- I2C\_STAT\_MSTCODE\_NACKDAT, [407](#)
- I2C\_STAT\_MSTCODE\_RXREADY, [407](#), [408](#)
- I2C\_STAT\_MSTCODE\_TXREADY, [408](#)
- I2C\_STAT\_MSTPENDING, [408](#)
- I2C\_STAT\_MSTRARBLOSS, [408](#)
- I2C\_STAT\_MSTSTATE, [408](#)
- I2C\_STAT\_MSTSTSPERR, [409](#)
- I2C\_STAT\_SCLTIMEOUT, [409](#)
- I2C\_STAT\_SLVCODE\_ADDR, [409](#)
- I2C\_STAT\_SLVCODE\_RX, [409](#)
- I2C\_STAT\_SLVCODE\_TX, [409](#), [410](#)
- I2C\_STAT\_SLVDESEL, [410](#)
- I2C\_STAT\_SLVIDX, [410](#)
- I2C\_STAT\_SLVNOTSTR, [410](#)
- I2C\_STAT\_SLVPENDING, [410](#)
- I2C\_STAT\_SLVSEL, [411](#)
- I2C\_STAT\_SLVSTATE, [411](#)
- I2C\_TIMEOUT\_VAL, [411](#)
- INTENCLR, [414](#)
- INTENSET, [414](#)
- INTSTAT, [414](#)
- LPC\_I2C\_T, [414](#)
- MONRXDAT, [414](#)
- MSTCTL, [414](#)
- MSTDAT, [414](#)
- MSTTIME, [415](#)
- PID, [415](#)
- PSELID, [415](#)
- RESERVED0, [415](#)
- RESERVED1, [415](#)
- RESERVED2, [415](#)
- SLVADR, [415](#)
- SLVCTL, [415](#)
- SLVDAT, [415](#)
- SLVQUAL0, [415](#)
- STAT, [416](#)
- TIMEOUT, [416](#)
- CHIP: RTC tick to (a more) Universal Time conversion functions, [417](#)
  - ConvertRtcTime, [417](#)
  - ConvertTimeRtc, [417](#)
  - TM\_DAYOFWEEK, [417](#)
  - TM\_YEAR\_BASE, [417](#)
- CHIP: Simple ring buffer implementation, [419](#)
  - RB\_VHEAD, [419](#)
  - RB\_VTAIL, [419](#)
  - RingBuffer\_Flush, [420](#)
  - RingBuffer\_GetCount, [420](#)
  - RingBuffer\_GetFree, [420](#)
  - RingBuffer\_GetSize, [420](#)
  - RingBuffer\_Init, [420](#)
  - RingBuffer\_Insert, [422](#)
  - RingBuffer\_InsertMult, [422](#)
  - RingBuffer\_IsEmpty, [422](#)
  - RingBuffer\_IsFull, [423](#)
  - RingBuffer\_Pop, [423](#)
  - RingBuffer\_PopMult, [423](#)
- CHIP: Stopwatch primitives., [424](#)
  - StopWatch\_DelayMs, [424](#)
  - StopWatch\_DelayTicks, [424](#)
  - StopWatch\_DelayUs, [425](#)
  - StopWatch\_Elapsed, [425](#)
  - StopWatch\_Init, [425](#)
  - StopWatch\_MsToTicks, [425](#)
  - StopWatch\_Start, [426](#)
  - StopWatch\_TicksPerSecond, [426](#)
  - StopWatch\_TicksToMs, [426](#)
  - StopWatch\_TicksToUs, [426](#)
  - StopWatch\_UsToTicks, [426](#)
- CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - ADC\_SEQA\_IRQn, [429](#)
  - ADC\_SEQB\_IRQn, [429](#)
  - ADC\_THCMP\_IRQn, [429](#)
  - CT32B0\_IRQn, [428](#)
  - CT32B1\_IRQn, [428](#)
  - CT32B3\_IRQn, [428](#)
  - DMA\_IRQn, [428](#)
  - DMIC\_IRQn, [429](#)
  - FLEXCOMM0\_IRQn, [428](#)
  - FLEXCOMM1\_IRQn, [428](#)
  - FLEXCOMM2\_IRQn, [428](#)
  - FLEXCOMM3\_IRQn, [428](#)
  - FLEXCOMM4\_IRQn, [428](#)
  - FLEXCOMM5\_IRQn, [428](#)
  - FLEXCOMM6\_IRQn, [428](#)
  - FLEXCOMM7\_IRQn, [429](#)
  - GINT0\_IRQn, [428](#)
  - GINT1\_IRQn, [428](#)
  - HWVAD, [429](#)
  - HardFault\_IRQn, [428](#)
  - LPC5411X\_M0\_IRQn\_Type, [428](#)
  - MAILBOX\_IRQn, [429](#)
  - MRT\_IRQn, [428](#)
  - NonMaskableInt\_IRQn, [428](#)

- PIN\_INT0\_IRQn, [428](#)
- PIN\_INT1\_IRQn, [428](#)
- PIN\_INT2\_IRQn, [428](#)
- PIN\_INT3\_IRQn, [428](#)
- PendSV\_IRQn, [428](#)
- RTC\_IRQn, [429](#)
- Reserved\_IRQn, [429](#)
- Reset\_IRQn, [428](#)
- SCT0\_IRQn, [428](#)
- SVCaI\_IRQn, [428](#)
- SysTick\_IRQn, [428](#)
- USB\_IRQn, [429](#)
- USBACT\_IRQn, [429](#)
- UTICK\_IRQn, [428](#)
- WDTBOD\_IRQn, [428](#)
- CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- ADC\_SEQA\_IRQn, [431](#)
- ADC\_SEQB\_IRQn, [431](#)
- ADC\_THCMP\_IRQn, [431](#)
- BusFault\_IRQn, [430](#)
- CT32B0\_IRQn, [430](#)
- CT32B1\_IRQn, [430](#)
- CT32B2\_IRQn, [431](#)
- CT32B3\_IRQn, [430](#)
- CT32B4\_IRQn, [431](#)
- DMA\_IRQn, [430](#)
- DMIC\_IRQn, [431](#)
- DebugMonitor\_IRQn, [430](#)
- FLEXCOMM0\_IRQn, [431](#)
- FLEXCOMM1\_IRQn, [431](#)
- FLEXCOMM2\_IRQn, [431](#)
- FLEXCOMM3\_IRQn, [431](#)
- FLEXCOMM4\_IRQn, [431](#)
- FLEXCOMM5\_IRQn, [431](#)
- FLEXCOMM6\_IRQn, [431](#)
- FLEXCOMM7\_IRQn, [431](#)
- GINT0\_IRQn, [430](#)
- GINT1\_IRQn, [430](#)
- HWVAD\_IRQn, [431](#)
- HardFault\_IRQn, [430](#)
- LPC5411X\_IRQn\_Type, [430](#)
- MAILBOX\_IRQn, [431](#)
- MRT\_IRQn, [430](#)
- MemoryManagement\_IRQn, [430](#)
- NonMaskableInt\_IRQn, [430](#)
- PIN\_INT0\_IRQn, [430](#)
- PIN\_INT1\_IRQn, [430](#)
- PIN\_INT2\_IRQn, [430](#)
- PIN\_INT3\_IRQn, [430](#)
- PIN\_INT4\_IRQn, [431](#)
- PIN\_INT5\_IRQn, [431](#)
- PIN\_INT6\_IRQn, [431](#)
- PIN\_INT7\_IRQn, [431](#)
- PendSV\_IRQn, [430](#)
- RTC\_IRQn, [431](#)
- Reserved1\_IRQn, [431](#)
- Reserved\_IRQn, [431](#)
- Reset\_IRQn, [430](#)
- SCT0\_IRQn, [430](#)
- SPIFI\_IRQn, [431](#)
- SVCaI\_IRQn, [430](#)
- SysTick\_IRQn, [430](#)
- USB\_IRQn, [431](#)
- USBACT\_IRQn, [431](#)
- UTICK\_IRQn, [430](#)
- UsageFault\_IRQn, [430](#)
- WDTBOD\_IRQn, [430](#)
- CHIP\_ASYNC\_SYSCON\_SRC\_T
  - CHIP: LPC5411X Clock Driver, [78](#)
- CHIP\_PMU\_BODRINTVAL\_T
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- CHIP\_PMU\_BODRSTLVL\_T
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- CHIP\_SCT\_EVENT\_T
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- CHIP\_SCT\_MATCH\_REG\_T
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- CHIP\_SYSCON\_ADCCLKSELSRC\_T
  - CHIP: LPC5411X Clock Driver, [78](#)
- CHIP\_SYSCON\_BOOT\_MODE\_REMAP\_T
  - CHIP: LPC5411X System and Control Driver, [330](#)
- CHIP\_SYSCON\_CLKOUTSRC\_T
  - CHIP: LPC5411X Clock Driver, [78](#)
- CHIP\_SYSCON\_CLOCK\_T
  - CHIP: LPC5411X Clock Driver, [78](#)
- CHIP\_SYSCON\_FLEXCOMMCLKSELSRC\_T
  - CHIP: LPC5411X Clock Driver, [79](#)
- CHIP\_SYSCON\_FRGCLKSRC\_T
  - CHIP: LPC5411X Clock Driver, [80](#)
- CHIP\_SYSCON\_MAIN\_A\_CLKSRC\_T
  - CHIP: LPC5411X Clock Driver, [80](#)
- CHIP\_SYSCON\_MAIN\_B\_CLKSRC\_T
  - CHIP: LPC5411X Clock Driver, [80](#)
- CHIP\_SYSCON\_MAINCLKSRC\_T
  - CHIP: LPC5411X Clock Driver, [80](#)
- CHIP\_SYSCON\_MCLKSRC\_T
  - CHIP: LPC5411X Clock Driver, [81](#)
- CHIP\_SYSCON\_PERIPH\_RESET\_T
  - CHIP: LPC5411X System and Control Driver, [331](#)
- CHIP\_SYSCON\_PLLCLKSRC\_T
  - CHIP: LPC5411X PLL Driver, [227](#)
- CHIP\_SYSCON\_USBCLKSRC\_T
  - CHIP: LPC5411X Clock Driver, [81](#)
- CHIP\_SYSCON\_WAKEUP\_T
  - CHIP: LPC5411X System and Control Driver, [331](#)
- CID0
  - ITM\_Type, [567](#)
- CID1
  - ITM\_Type, [567](#)
- CID2
  - ITM\_Type, [567](#)

- CID3
  - ITM\_Type, [567](#)
- CIENF
  - LPC\_PIN\_INT\_T, [592](#)
- CIENR
  - LPC\_PIN\_INT\_T, [592](#)
- CLAIMCLR
  - TPI\_Type, [645](#)
- CLAIMSET
  - TPI\_Type, [645](#)
- CLKDIV
  - CHIP: LPC5411x I2C driver, [414](#)
- CLKOUTDIV
  - LPC\_SYSCON\_T, [610](#)
- CLKOUTSELA
  - LPC\_SYSCON\_T, [610](#)
- CLR
  - LPC\_GPIO\_T, [581](#)
  - LPC\_SCT\_T, [599](#)
- CMSIS Core Instruction Interface, [432](#)
- CMSIS Core Register Access Functions, [433](#)
- CMSIS Global Defines, [434](#)
- CMSIS SIMD Intrinsics, [435](#)
- CMSIS support, [436](#)
- COMP0
  - DWT\_Type, [557](#)
- COMP1
  - DWT\_Type, [557](#)
- COMP2
  - DWT\_Type, [557](#)
- COMP3
  - DWT\_Type, [557](#)
- COMPENSATION\_T
  - CHIP: LPC5411X DMIC driver, [137](#)
- COMPILE\_TIME\_ASSERT
  - error.h, [705](#)
- CONEN
  - LPC\_SCT\_T, [599](#)
- CONFIG
  - LPC\_SCT\_T, [599](#)
- CONFIG\_SCT\_nEV
  - CHIP: LPC5411X State Configurable Timer driver, [309](#)
- CONFIG\_SCT\_nIN
  - CHIP: LPC5411X State Configurable Timer driver, [309](#)
- CONFIG\_SCT\_nOU
  - CHIP: LPC5411X State Configurable Timer driver, [310](#)
- CONFIG\_SCT\_nRG
  - CHIP: LPC5411X State Configurable Timer driver, [310](#)
- CONFLAG
  - LPC\_SCT\_T, [599](#)
- CONTROL\_Type, [551](#)
  - \_reserved0, [552](#)
  - b, [552](#)
  - FPCA, [552](#)
  - nPRIV, [552](#)
  - SPSEL, [552](#)
  - w, [553](#)
- CORESELECT\_M0PLUS
  - CHIP: LPC5411X CPU multi-core support driver, [72](#)
- CORESELECT\_M4
  - CHIP: LPC5411X CPU multi-core support driver, [72](#)
- CORESELECT\_T
  - CHIP: LPC5411X CPU multi-core support driver, [72](#)
- COUNT
  - LPC\_RTC\_T, [596](#)
- COUNT\_H
  - LPC\_SCT\_T, [599](#)
- COUNT\_L
  - LPC\_SCT\_T, [599](#)
- COUNT\_U
  - LPC\_SCT\_T, [600](#)
- COUNTUP\_TO
  - CHIP: LPC5411X State Configurable Timer driver, [310](#)
- COUNTUP\_TO\_LIMIT\_THEN\_CLEAR\_TO\_ZERO
  - CHIP: LPC5411X State Configurable Timer driver, [310](#)
- CPACR
  - SCB\_Type, [635](#)
- CPBOOT
  - LPC\_SYSCON\_T, [610](#)
- CPCTRL
  - LPC\_SYSCON\_T, [610](#)
- CPICNT
  - DWT\_Type, [557](#)
- CPSTACK
  - LPC\_SYSCON\_T, [610](#)
- CPSTAT
  - LPC\_SYSCON\_T, [610](#)
- CPUID
  - SCB\_Type, [635](#)
- CR
  - LPC\_TIMER\_T, [619](#)
- CRC\_MODE\_POLY\_BITMASK
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [98](#)
- CRC\_MODE\_POLY\_CCITT
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [98](#)
- CRC\_MODE\_POLY\_CRC16
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [98](#)
- CRC\_MODE\_POLY\_CRC32
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [98](#)
- CRC\_MODE\_SUM\_BIT\_RVS
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [98](#)
- CRC\_MODE\_SUM\_BITMASK

- CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [98](#)
- CRC\_MODE\_SUM\_CMPL
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [98](#)
- CRC\_MODE\_WRDATA\_BIT\_RVS
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CRC\_MODE\_WRDATA\_BITMASK
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CRC\_MODE\_WRDATA\_CMPL
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CRC\_POLY\_CCITT
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CRC\_POLY\_CRC16
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CRC\_POLY\_CRC32
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CRC\_POLY\_LAST
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [100](#)
- CRC\_POLY\_T
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CRC\_SEED\_CCITT
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CRC\_SEED\_CRC16
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CRC\_SEED\_CRC32
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- CSPSR
  - TPI\_Type, [645](#)
- CT32B0\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- CT32B1\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- CT32B2\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- CT32B3\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- CT32B4\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- CTCR
  - LPC\_TIMER\_T, [619](#)
- CTL
  - LPC\_USART\_T, [622](#)
- CTLSTAT
  - LPC\_DMA\_CHANNEL\_T, [575](#)
- CTRL
  - DWT\_Type, [557](#)
  - LPC\_ADC\_T, [571](#)
  - LPC\_DMA\_T, [579](#)
  - LPC\_GPIOGROUPINT\_T, [583](#)
  - LPC\_MRT\_CH\_T, [590](#)
  - LPC\_RTC\_T, [596](#)
  - LPC\_SCT\_T, [600](#)
  - LPC\_UTICK\_T, [624](#)
  - SysTick\_Type, [643](#)
- CTRL\_H
  - LPC\_SCT\_T, [600](#)
- CTRL\_L
  - LPC\_SCT\_T, [600](#)
- CTRL\_U
  - LPC\_SCT\_T, [600](#)
- CYCCNT
  - DWT\_Type, [557](#)
- channel
  - CHIP: LPC5411X DMA Service driver, [132](#)
- ChannelNumber
  - I2S\_AUDIO\_FORMAT\_T, [562](#)
- Chip specific drivers, [437](#)
- Chip\_ADC\_Calibration
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- Chip\_ADC\_ClearFlags
  - CHIP: LPC5411X A/D conversion driver, [30](#)
- Chip\_ADC\_ClearSequencerBits
  - CHIP: LPC5411X A/D conversion driver, [31](#)
- Chip\_ADC\_ClearTHRSELBits
  - CHIP: LPC5411X A/D conversion driver, [31](#)
- Chip\_ADC\_Delnit
  - CHIP: LPC5411X A/D conversion driver, [31](#)
- Chip\_ADC\_DisableInt
  - CHIP: LPC5411X A/D conversion driver, [32](#)
- Chip\_ADC\_DisableSequencer
  - CHIP: LPC5411X A/D conversion driver, [32](#)
- Chip\_ADC\_EnableInt
  - CHIP: LPC5411X A/D conversion driver, [32](#)
- Chip\_ADC\_EnableSequencer
  - CHIP: LPC5411X A/D conversion driver, [33](#)
- Chip\_ADC\_GetDataReg
  - CHIP: LPC5411X A/D conversion driver, [33](#)
- Chip\_ADC\_GetDivider
  - CHIP: LPC5411X A/D conversion driver, [33](#)
- Chip\_ADC\_GetFlags
  - CHIP: LPC5411X A/D conversion driver, [35](#)
- Chip\_ADC\_GetGlobalDataReg
  - CHIP: LPC5411X A/D conversion driver, [35](#)

- Chip\_ADC\_GetSequencerCtrl
  - CHIP: LPC5411X A/D conversion driver, [36](#)
- Chip\_ADC\_Init
  - CHIP: LPC5411X A/D conversion driver, [36](#)
- Chip\_ADC\_SelectTH0Channels
  - CHIP: LPC5411X A/D conversion driver, [37](#)
- Chip\_ADC\_SelectTH1Channels
  - CHIP: LPC5411X A/D conversion driver, [37](#)
- Chip\_ADC\_SelectTempSensorInput
  - CHIP: LPC5411X A/D conversion driver, [36](#)
- Chip\_ADC\_SetClockRate
  - CHIP: LPC5411X A/D conversion driver, [37](#)
- Chip\_ADC\_SetDivider
  - CHIP: LPC5411X A/D conversion driver, [38](#)
- Chip\_ADC\_SetSequencerBits
  - CHIP: LPC5411X A/D conversion driver, [38](#)
- Chip\_ADC\_SetTHRSELBits
  - CHIP: LPC5411X A/D conversion driver, [40](#)
- Chip\_ADC\_SetThrHighValue
  - CHIP: LPC5411X A/D conversion driver, [40](#)
- Chip\_ADC\_SetThrLowValue
  - CHIP: LPC5411X A/D conversion driver, [40](#)
- Chip\_ADC\_SetThresholdInt
  - CHIP: LPC5411X A/D conversion driver, [38](#)
- Chip\_ADC\_SetupSequencer
  - CHIP: LPC5411X A/D conversion driver, [40](#)
- Chip\_ADC\_StartBurstSequencer
  - CHIP: LPC5411X A/D conversion driver, [42](#)
- Chip\_ADC\_StartSequencer
  - CHIP: LPC5411X A/D conversion driver, [42](#)
- Chip\_ADC\_StopBurstSequencer
  - CHIP: LPC5411X A/D conversion driver, [43](#)
- Chip\_CPU\_CM0Boot
  - CHIP: LPC5411X CPU multi-core support driver, [72](#)
- Chip\_CPU\_CM4Boot
  - CHIP: LPC5411X CPU multi-core support driver, [73](#)
- Chip\_CPU\_IsM4Core
  - CHIP: LPC5411X CPU multi-core support driver, [73](#)
- Chip\_CPU\_IsMasterCore
  - CHIP: LPC5411X CPU multi-core support driver, [73](#)
- Chip\_CPU\_SelectMasterCore
  - CHIP: LPC5411X CPU multi-core support driver, [73](#)
- Chip\_CRC\_CRC16
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [100](#)
- Chip\_CRC\_CRC32
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [100](#)
- Chip\_CRC\_CRC8
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [100](#)
- Chip\_CRC\_Deinit
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [100](#)
- Chip\_CRC\_GetMode
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [101](#)
- Chip\_CRC\_GetSeed
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [101](#)
- Chip\_CRC\_Init
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [101](#)
- Chip\_CRC\_SetMode
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [101](#)
- Chip\_CRC\_SetPoly
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [102](#)
- Chip\_CRC\_SetSeed
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [102](#)
- Chip\_CRC\_Sum
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [102](#)
- Chip\_CRC\_UseCCITT
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [103](#)
- Chip\_CRC\_UseCRC16
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [103](#)
- Chip\_CRC\_UseCRC32
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [103](#)
- Chip\_CRC\_UseDefaultConfig
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [103](#)
- Chip\_CRC\_Write16
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [104](#)
- Chip\_CRC\_Write32
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [104](#)
- Chip\_CRC\_Write8
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [104](#)
- Chip\_Clock\_DisablePeriphClock
  - CHIP: LPC5411X Clock Driver, [82](#)
- Chip\_Clock\_DisableRTCOsc
  - CHIP: LPC5411X Clock Driver, [82](#)
- Chip\_Clock\_EnablePeriphClock
  - CHIP: LPC5411X Clock Driver, [82](#)
- Chip\_Clock\_EnableRTCOsc
  - CHIP: LPC5411X Clock Driver, [84](#)
- Chip\_Clock\_GetADCClockDiv
  - CHIP: LPC5411X Clock Driver, [84](#)
- Chip\_Clock\_GetADCClockRate
  - CHIP: LPC5411X Clock Driver, [84](#)
- Chip\_Clock\_GetADCClockSource
  - CHIP: LPC5411X Clock Driver, [84](#)



- Chip\_Clock\_GetAsyncSyscon\_ClockRate
  - CHIP: LPC5411X Clock Driver, [84](#)
- Chip\_Clock\_GetAsyncSyscon\_ClockRate\_NoDiv
  - clock\_5411x.c, [777](#)
- Chip\_Clock\_GetAsyncSysconClockSource
  - CHIP: LPC5411X Clock Driver, [85](#)
- Chip\_Clock\_GetCLKOUTDiv
  - CHIP: LPC5411X Clock Driver, [85](#)
- Chip\_Clock\_GetCLKOUTSource
  - CHIP: LPC5411X Clock Driver, [85](#)
- Chip\_Clock\_GetExtClockInRate
  - CHIP: LPC5411X Clock Driver, [85](#)
- Chip\_Clock\_GetFLEXCOMMClockRate
  - CHIP: LPC5411X Clock Driver, [85](#)
- Chip\_Clock\_GetFLEXCOMMClockSource
  - CHIP: LPC5411X Clock Driver, [86](#)
- Chip\_Clock\_GetFRGClockRate
  - CHIP: LPC5411X Clock Driver, [86](#)
- Chip\_Clock\_GetFRGClockSource
  - CHIP: LPC5411X Clock Driver, [86](#)
- Chip\_Clock\_GetFRGInClockRate
  - CHIP: LPC5411X Clock Driver, [86](#)
- Chip\_Clock\_GetFROHFRate
  - CHIP: LPC5411X Clock Driver, [86](#)
- Chip\_Clock\_GetIntOscRate
  - CHIP: LPC5411X Clock Driver, [77](#)
- Chip\_Clock\_GetMCLKClockRate
  - clock\_5411x.c, [777](#)
- Chip\_Clock\_GetMCLKDir
  - CHIP: LPC5411X Clock Driver, [88](#)
- Chip\_Clock\_GetMCLKDiv
  - CHIP: LPC5411X Clock Driver, [88](#)
- Chip\_Clock\_GetMCLKSource
  - CHIP: LPC5411X Clock Driver, [88](#)
- Chip\_Clock\_GetMain\_A\_ClockRate
  - CHIP: LPC5411X Clock Driver, [87](#)
- Chip\_Clock\_GetMain\_A\_ClockSource
  - CHIP: LPC5411X Clock Driver, [87](#)
- Chip\_Clock\_GetMain\_B\_ClockRate
  - CHIP: LPC5411X Clock Driver, [87](#)
- Chip\_Clock\_GetMain\_B\_ClockSource
  - CHIP: LPC5411X Clock Driver, [87](#)
- Chip\_Clock\_GetMainClockRate
  - CHIP: LPC5411X Clock Driver, [87](#)
- Chip\_Clock\_GetMainClockSource
  - CHIP: LPC5411X Clock Driver, [87](#)
- Chip\_Clock\_GetPIIConfig
  - pll\_5411x.c, [795](#)
- Chip\_Clock\_GetRTCOsc
  - CHIP: LPC5411X Clock Driver, [88](#)
- Chip\_Clock\_GetRTCOscRate
  - CHIP: LPC5411X Clock Driver, [88](#)
- Chip\_Clock\_GetStoredPLLClockRate
  - CHIP: LPC5411X PLL Driver, [228](#)
- Chip\_Clock\_GetSysClockDiv
  - CHIP: LPC5411X Clock Driver, [88](#)
- Chip\_Clock\_GetSysTickClockDiv
  - CHIP: LPC5411X Clock Driver, [89](#)
- Chip\_Clock\_GetSysTickClockRate
  - CHIP: LPC5411X Clock Driver, [89](#)
- Chip\_Clock\_GetSystemClockRate
  - CHIP: LPC5411X Clock Driver, [89](#)
- Chip\_Clock\_GetSystemPLLInClockRate
  - CHIP: LPC5411X PLL Driver, [228](#)
- Chip\_Clock\_GetSystemPLLOutClockRate
  - CHIP: LPC5411X PLL Driver, [228](#)
- Chip\_Clock\_GetSystemPLLOutFromSetup
  - CHIP: LPC5411X PLL Driver, [229](#)
- Chip\_Clock\_GetSystemPLLOutFromSetupUpdate
  - pll\_5411x.c, [795](#)
- Chip\_Clock\_GetUSBClockDiv
  - CHIP: LPC5411X Clock Driver, [89](#)
- Chip\_Clock\_GetUSBClockSource
  - CHIP: LPC5411X Clock Driver, [89](#)
- Chip\_Clock\_GetWDTOSCRate
  - CHIP: LPC5411X Clock Driver, [90](#)
- Chip\_Clock\_IsSystemPLLLocked
  - CHIP: LPC5411X PLL Driver, [229](#)
- Chip\_Clock\_SetADCClockDiv
  - CHIP: LPC5411X Clock Driver, [90](#)
- Chip\_Clock\_SetADCClockSource
  - CHIP: LPC5411X Clock Driver, [90](#)
- Chip\_Clock\_SetAsyncSysconClockSource
  - CHIP: LPC5411X Clock Driver, [90](#)
- Chip\_Clock\_SetBypassPLL
  - CHIP: LPC5411X PLL Driver, [229](#)
- Chip\_Clock\_SetCLKOUTSource
  - CHIP: LPC5411X Clock Driver, [91](#)
- Chip\_Clock\_SetFLEXCOMMClockSource
  - CHIP: LPC5411X Clock Driver, [91](#)
- Chip\_Clock\_SetFRGClockRate
  - CHIP: LPC5411X Clock Driver, [91](#)
- Chip\_Clock\_SetFRGClockSource
  - CHIP: LPC5411X Clock Driver, [91](#)
- Chip\_Clock\_SetMCLKClockSource
  - CHIP: LPC5411X Clock Driver, [93](#)
- Chip\_Clock\_SetMCLKDir
  - CHIP: LPC5411X Clock Driver, [93](#)
- Chip\_Clock\_SetMCLKDirInput
  - CHIP: LPC5411X Clock Driver, [93](#)
- Chip\_Clock\_SetMCLKDirOutput
  - CHIP: LPC5411X Clock Driver, [93](#)
- Chip\_Clock\_SetMain\_A\_ClockSource
  - CHIP: LPC5411X Clock Driver, [92](#)
- Chip\_Clock\_SetMain\_B\_ClockSource
  - CHIP: LPC5411X Clock Driver, [92](#)
- Chip\_Clock\_SetMainClockSource
  - CHIP: LPC5411X Clock Driver, [92](#)
- Chip\_Clock\_SetPLLFreq
  - CHIP: LPC5411X PLL Driver, [229](#)
- Chip\_Clock\_SetStoredPLLClockRate
  - CHIP: LPC5411X PLL Driver, [230](#)
- Chip\_Clock\_SetSysClockDiv
  - CHIP: LPC5411X Clock Driver, [93](#)
- Chip\_Clock\_SetSysTickClockDiv
  - CHIP: LPC5411X Clock Driver, [95](#)

- Chip\_Clock\_SetSystemPLLSource
  - CHIP: LPC5411X PLL Driver, [230](#)
- Chip\_Clock\_SetUSBClockSource
  - CHIP: LPC5411X Clock Driver, [95](#)
- Chip\_Clock\_SetWDTOSCRate
  - CHIP: LPC5411X Clock Driver, [95](#)
- Chip\_Clock\_SetupPLLData
  - CHIP: LPC5411X PLL Driver, [230](#)
- Chip\_Clock\_SetupSystemPLL
  - CHIP: LPC5411X PLL Driver, [231](#)
- Chip\_Clock\_SetupSystemPLLPre
  - CHIP: LPC5411X PLL Driver, [231](#)
- Chip\_DMA\_AbortChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [111](#)
- Chip\_DMA\_ClearActiveIntAChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [112](#)
- Chip\_DMA\_ClearActiveIntBChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [112](#)
- Chip\_DMA\_ClearErrorIntChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [112](#)
- Chip\_DMA\_ClearTranBits
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [105](#)
- Chip\_DMA\_DeInit
  - CHIP: LPC5411X DMA Controller driver common functions (legacy), [118](#)
- Chip\_DMA\_Disable
  - CHIP: LPC5411X DMA Controller driver common functions (legacy), [118](#)
- Chip\_DMA\_DisableChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [113](#)
- Chip\_DMA\_DisableIntChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [113](#)
- Chip\_DMA\_Enable
  - CHIP: LPC5411X DMA Controller driver common functions (legacy), [118](#)
- Chip\_DMA\_EnableChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [113](#)
- Chip\_DMA\_EnableIntChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [113](#)
- Chip\_DMA\_GetActiveChannels
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [114](#)
- Chip\_DMA\_GetActiveIntAChannels
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [114](#)
- Chip\_DMA\_GetActiveIntBChannels
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [114](#)
- Chip\_DMA\_GetBusyChannels
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [115](#)
- Chip\_DMA\_GetChannelStatus
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [106](#)
- Chip\_DMA\_GetEnableIntChannels
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [115](#)
- Chip\_DMA\_GetEnabledChannels
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [115](#)
- Chip\_DMA\_GetErrorIntChannels
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [116](#)
- Chip\_DMA\_GetIntStatus
  - CHIP: LPC5411X DMA Controller driver common functions (legacy), [119](#)
- Chip\_DMA\_GetSRAMBase
  - CHIP: LPC5411X DMA Controller driver common functions (legacy), [119](#)
- Chip\_DMA\_Init
  - CHIP: LPC5411X DMA Controller driver common functions (legacy), [119](#)
- Chip\_DMA\_SWTriggerChannel
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [110](#)
- Chip\_DMA\_SetChannelInvalid
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [106](#)
- Chip\_DMA\_SetChannelValid
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [106](#)
- Chip\_DMA\_SetSRAMBase
  - CHIP: LPC5411X DMA Controller driver common functions (legacy), [119](#)
- Chip\_DMA\_SetTranBits
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [106](#)
- Chip\_DMA\_SetTrigChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [116](#)
- Chip\_DMA\_SetValidChannel
  - CHIP: LPC5411X DMA Controller driver common channel functions (legacy), [116](#)
- Chip\_DMA\_SetupChannelConfig
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [108](#)
- Chip\_DMA\_SetupChannelTransfer
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [108](#)
- Chip\_DMA\_SetupChannelTransferSize
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [109](#)
- Chip\_DMA\_SetupTranChannel
  - CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [109](#)
- Chip\_DMA\_Table



- CHIP: LPC5411X DMA Controller driver channel specific functions (legacy), [110](#)
- Chip\_DMASERVICE\_DoubleBuffer
  - CHIP: LPC5411X DMA Service driver, [129](#)
- Chip\_DMASERVICE\_ErrorHandler
  - dma\_service\_5411x.c, [779](#)
- Chip\_DMASERVICE\_Init
  - CHIP: LPC5411X DMA Service driver, [130](#)
- Chip\_DMASERVICE\_Isr
  - CHIP: LPC5411X DMA Service driver, [130](#)
- Chip\_DMASERVICE\_RegisterCb
  - CHIP: LPC5411X DMA Service driver, [130](#)
- Chip\_DMASERVICE\_SingleBuffer
  - CHIP: LPC5411X DMA Service driver, [130](#)
- Chip\_DMIC\_CfgChannel
  - CHIP: LPC5411X DMIC driver, [138](#)
- Chip\_DMIC\_CfgChannelDc
  - CHIP: LPC5411X DMIC driver, [139](#)
- Chip\_DMIC\_CfgIO
  - CHIP: LPC5411X DMIC driver, [139](#)
- Chip\_DMIC\_EnableChannnel
  - CHIP: LPC5411X DMIC driver, [139](#)
- Chip\_DMIC\_FifoChannel
  - CHIP: LPC5411X DMIC driver, [139](#)
- Chip\_DMIC\_FifoClearStatus
  - CHIP: LPC5411X DMIC driver, [141](#)
- Chip\_DMIC\_FifoGetData
  - CHIP: LPC5411X DMIC driver, [141](#)
- Chip\_DMIC\_FifoGetStatus
  - CHIP: LPC5411X DMIC driver, [141](#)
- Chip\_DMIC\_Init
  - CHIP: LPC5411X DMIC driver, [141](#)
- Chip\_DMIC\_SetOpMode
  - CHIP: LPC5411X DMIC driver, [143](#)
- Chip\_DMIC\_Use2fs
  - CHIP: LPC5411X DMIC driver, [143](#)
- Chip\_FLEXCOMM\_DelInit
  - CHIP: LPC5411X flexcomm API, [380](#)
- Chip\_FLEXCOMM\_GetFunc
  - CHIP: LPC5411X flexcomm API, [380](#)
- Chip\_FLEXCOMM\_GetIndex
  - CHIP: LPC5411X flexcomm API, [380](#)
- Chip\_FLEXCOMM\_Init
  - CHIP: LPC5411X flexcomm API, [382](#)
- Chip\_FLEXCOMM\_IsLocked
  - CHIP: LPC5411X flexcomm API, [382](#)
- Chip\_FLEXCOMM\_Lock
  - CHIP: LPC5411X flexcomm API, [382](#)
- Chip\_FLEXCOMM\_SetPeriph
  - CHIP: LPC5411X flexcomm API, [382](#)
- Chip\_GPIO\_ClearValue
  - CHIP: LPC5411X GPIO driver, [151](#)
- Chip\_GPIO\_DelInit
  - CHIP: LPC5411X GPIO driver, [151](#)
- Chip\_GPIO\_GetMaskedPortValue
  - CHIP: LPC5411X GPIO driver, [153](#)
- Chip\_GPIO\_GetPinDIR
  - CHIP: LPC5411X GPIO driver, [153](#)
- Chip\_GPIO\_GetPinState
  - CHIP: LPC5411X GPIO driver, [153](#)
- Chip\_GPIO\_GetPortDIR
  - CHIP: LPC5411X GPIO driver, [153](#)
- Chip\_GPIO\_GetPortMask
  - CHIP: LPC5411X GPIO driver, [155](#)
- Chip\_GPIO\_GetPortValue
  - CHIP: LPC5411X GPIO driver, [155](#)
- Chip\_GPIO\_Init
  - CHIP: LPC5411X GPIO driver, [155](#)
- Chip\_GPIO\_ReadDirBit
  - CHIP: LPC5411X GPIO driver, [156](#)
- Chip\_GPIO\_ReadPortBit
  - CHIP: LPC5411X GPIO driver, [156](#)
- Chip\_GPIO\_ReadValue
  - CHIP: LPC5411X GPIO driver, [156](#)
- Chip\_GPIO\_SetDir
  - CHIP: LPC5411X GPIO driver, [157](#)
- Chip\_GPIO\_SetMaskedPortValue
  - CHIP: LPC5411X GPIO driver, [157](#)
- Chip\_GPIO\_SetPinDIR
  - CHIP: LPC5411X GPIO driver, [157](#)
- Chip\_GPIO\_SetPinDIRInput
  - CHIP: LPC5411X GPIO driver, [157](#)
- Chip\_GPIO\_SetPinDIROutput
  - CHIP: LPC5411X GPIO driver, [158](#)
- Chip\_GPIO\_SetPinOutHigh
  - CHIP: LPC5411X GPIO driver, [158](#)
- Chip\_GPIO\_SetPinOutLow
  - CHIP: LPC5411X GPIO driver, [158](#)
- Chip\_GPIO\_SetPinState
  - CHIP: LPC5411X GPIO driver, [159](#)
- Chip\_GPIO\_SetPinToggle
  - CHIP: LPC5411X GPIO driver, [159](#)
- Chip\_GPIO\_SetPortDIR
  - CHIP: LPC5411X GPIO driver, [159](#)
- Chip\_GPIO\_SetPortDIRInput
  - CHIP: LPC5411X GPIO driver, [160](#)
- Chip\_GPIO\_SetPortDIROutput
  - CHIP: LPC5411X GPIO driver, [160](#)
- Chip\_GPIO\_SetPortMask
  - CHIP: LPC5411X GPIO driver, [160](#)
- Chip\_GPIO\_SetPortOutHigh
  - CHIP: LPC5411X GPIO driver, [162](#)
- Chip\_GPIO\_SetPortOutLow
  - CHIP: LPC5411X GPIO driver, [162](#)
- Chip\_GPIO\_SetPortToggle
  - CHIP: LPC5411X GPIO driver, [162](#)
- Chip\_GPIO\_SetPortValue
  - CHIP: LPC5411X GPIO driver, [163](#)
- Chip\_GPIO\_SetValue
  - CHIP: LPC5411X GPIO driver, [163](#)
- Chip\_GPIO\_WriteDirBit
  - CHIP: LPC5411X GPIO driver, [163](#)
- Chip\_GPIO\_WritePortBit
  - CHIP: LPC5411X GPIO driver, [164](#)
- Chip\_GPIOGP\_ClearIntStatus
  - CHIP: LPC5411X GPIO group driver, [166](#)

- Chip\_GPIOGP\_DeInit
  - CHIP: LPC5411X GPIO group driver, [166](#)
- Chip\_GPIOGP\_DisableGroupPins
  - CHIP: LPC5411X GPIO group driver, [166](#)
- Chip\_GPIOGP\_EnableGroupPins
  - CHIP: LPC5411X GPIO group driver, [167](#)
- Chip\_GPIOGP\_GetIntStatus
  - CHIP: LPC5411X GPIO group driver, [167](#)
- Chip\_GPIOGP\_Init
  - CHIP: LPC5411X GPIO group driver, [167](#)
- Chip\_GPIOGP\_SelectAndMode
  - CHIP: LPC5411X GPIO group driver, [168](#)
- Chip\_GPIOGP\_SelectEdgeMode
  - CHIP: LPC5411X GPIO group driver, [168](#)
- Chip\_GPIOGP\_SelectHighLevel
  - CHIP: LPC5411X GPIO group driver, [168](#)
- Chip\_GPIOGP\_SelectLevelMode
  - CHIP: LPC5411X GPIO group driver, [168](#)
- Chip\_GPIOGP\_SelectLowLevel
  - CHIP: LPC5411X GPIO group driver, [170](#)
- Chip\_GPIOGP\_SelectOrMode
  - CHIP: LPC5411X GPIO group driver, [170](#)
- Chip\_I2C\_ClearInt
  - CHIP: LPC5411x I2C driver, [411](#)
- Chip\_I2C\_DeInit
  - CHIP: LPC5411x I2C driver, [412](#)
- Chip\_I2C\_DisableInt
  - CHIP: LPC5411x I2C driver, [412](#)
- Chip\_I2C\_EnableInt
  - CHIP: LPC5411x I2C driver, [412](#)
- Chip\_I2C\_GetClockDiv
  - CHIP: LPC5411x I2C driver, [412](#)
- Chip\_I2C\_GetPendingInt
  - CHIP: LPC5411x I2C driver, [413](#)
- Chip\_I2C\_Init
  - CHIP: LPC5411x I2C driver, [413](#)
- Chip\_I2C\_SetClockDiv
  - CHIP: LPC5411x I2C driver, [413](#)
- Chip\_I2CM\_ClearStatus
  - CHIP: LPC5411X I2C master-only driver, [173](#)
- Chip\_I2CM\_Disable
  - CHIP: LPC5411X I2C master-only driver, [173](#)
- Chip\_I2CM\_Enable
  - CHIP: LPC5411X I2C master-only driver, [173](#)
- Chip\_I2CM\_GetMasterState
  - CHIP: LPC5411X I2C master-only driver, [173](#)
- Chip\_I2CM\_GetStatus
  - CHIP: LPC5411X I2C master-only driver, [174](#)
- Chip\_I2CM\_IsMasterPending
  - CHIP: LPC5411X I2C master-only driver, [174](#)
- Chip\_I2CM\_MasterContinue
  - CHIP: LPC5411X I2C master-only driver, [174](#)
- Chip\_I2CM\_ReadByte
  - CHIP: LPC5411X I2C master-only driver, [175](#)
- Chip\_I2CM\_SendStart
  - CHIP: LPC5411X I2C master-only driver, [175](#)
- Chip\_I2CM\_SendStop
  - CHIP: LPC5411X I2C master-only driver, [175](#)
- Chip\_I2CM\_SetBusSpeed
  - CHIP: LPC5411X I2C master-only driver, [176](#)
- Chip\_I2CM\_SetDutyCycle
  - CHIP: LPC5411X I2C master-only driver, [176](#)
- Chip\_I2CM\_WriteByte
  - CHIP: LPC5411X I2C master-only driver, [177](#)
- Chip\_I2CM\_Xfer
  - CHIP: LPC5411X I2C master-only driver, [177](#)
- Chip\_I2CM\_XferBlocking
  - CHIP: LPC5411X I2C master-only driver, [178](#)
- Chip\_I2CM\_XferHandler
  - CHIP: LPC5411X I2C master-only driver, [178](#)
- Chip\_I2CS\_ClearStatus
  - CHIP: LPC5411X I2C slave-only driver, [181](#)
- Chip\_I2CS\_Disable
  - CHIP: LPC5411X I2C slave-only driver, [181](#)
- Chip\_I2CS\_DisableSlaveAddr
  - CHIP: LPC5411X I2C slave-only driver, [181](#)
- Chip\_I2CS\_Enable
  - CHIP: LPC5411X I2C slave-only driver, [182](#)
- Chip\_I2CS\_EnableSlaveAddr
  - CHIP: LPC5411X I2C slave-only driver, [182](#)
- Chip\_I2CS\_GetSlaveAddr
  - CHIP: LPC5411X I2C slave-only driver, [182](#)
- Chip\_I2CS\_GetSlaveMatchIndex
  - CHIP: LPC5411X I2C slave-only driver, [182](#)
- Chip\_I2CS\_GetSlaveState
  - CHIP: LPC5411X I2C slave-only driver, [183](#)
- Chip\_I2CS\_GetStatus
  - CHIP: LPC5411X I2C slave-only driver, [183](#)
- Chip\_I2CS\_IsSlaveDeSelected
  - CHIP: LPC5411X I2C slave-only driver, [183](#)
- Chip\_I2CS\_IsSlavePending
  - CHIP: LPC5411X I2C slave-only driver, [183](#)
- Chip\_I2CS\_IsSlaveSelected
  - CHIP: LPC5411X I2C slave-only driver, [184](#)
- Chip\_I2CS\_ReadByte
  - CHIP: LPC5411X I2C slave-only driver, [184](#)
- Chip\_I2CS\_SetSlaveAddr
  - CHIP: LPC5411X I2C slave-only driver, [184](#)
- Chip\_I2CS\_SetSlaveQual0
  - CHIP: LPC5411X I2C slave-only driver, [185](#)
- Chip\_I2CS\_SlaveContinue
  - CHIP: LPC5411X I2C slave-only driver, [185](#)
- Chip\_I2CS\_SlaveDisableDMA
  - CHIP: LPC5411X I2C slave-only driver, [185](#)
- Chip\_I2CS\_SlaveEnableDMA
  - CHIP: LPC5411X I2C slave-only driver, [186](#)
- Chip\_I2CS\_SlaveNACK
  - CHIP: LPC5411X I2C slave-only driver, [186](#)
- Chip\_I2CS\_WriteByte
  - CHIP: LPC5411X I2C slave-only driver, [186](#)
- Chip\_I2CS\_XferHandler
  - CHIP: LPC5411X I2C slave-only driver, [187](#)
- Chip\_I2S\_ClearStatus
  - i2s\_5411x.h, [730](#)
- Chip\_I2S\_ClrFIFOStatus
  - i2s\_5411x.h, [730](#)

- Chip\_I2S\_Config
  - i2s\_5411x.c, [781](#)
  - i2s\_5411x.h, [731](#)
- Chip\_I2S\_DeInit
  - i2s\_5411x.h, [731](#)
- Chip\_I2S\_ErrorHandler
  - i2s\_5411x.c, [781](#)
  - i2s\_5411x.h, [731](#)
- Chip\_I2S\_FIFO\_ClearStatus
  - i2s\_5411x.h, [731](#)
- Chip\_I2S\_FIFO\_ClrInterrupt
  - i2s\_5411x.h, [733](#)
- Chip\_I2S\_FIFO\_Config
  - i2s\_5411x.c, [782](#)
  - i2s\_5411x.h, [733](#)
- Chip\_I2S\_FIFO\_Control
  - i2s\_5411x.c, [782](#)
  - i2s\_5411x.h, [733](#)
- Chip\_I2S\_FIFO\_GetPendingInts
  - i2s\_5411x.h, [733](#)
- Chip\_I2S\_FIFO\_SetInterrupt
  - i2s\_5411x.h, [734](#)
- Chip\_I2S\_GetFIFORxLevel
  - i2s\_5411x.h, [734](#)
- Chip\_I2S\_GetFIFOStatus
  - i2s\_5411x.h, [734](#)
- Chip\_I2S\_GetFIFOTrigLevel
  - i2s\_5411x.h, [734](#)
- Chip\_I2S\_GetFIFOTxLevel
  - i2s\_5411x.h, [735](#)
- Chip\_I2S\_GetStatus
  - i2s\_5411x.h, [735](#)
- Chip\_I2S\_Init
  - i2s\_5411x.c, [782](#)
  - i2s\_5411x.h, [735](#)
- Chip\_I2S\_Pause
  - i2s\_5411x.h, [735](#)
- Chip\_I2S\_Play
  - i2s\_5411x.h, [736](#)
- Chip\_I2S\_RX\_Init
  - i2s\_5411x.h, [736](#)
- Chip\_I2S\_Receive
  - i2s\_5411x.h, [736](#)
- Chip\_I2S\_Send
  - i2s\_5411x.h, [736](#)
- Chip\_I2S\_SetFIFOTrigLevel
  - i2s\_5411x.h, [737](#)
- Chip\_I2S\_Start
  - i2s\_5411x.h, [737](#)
- Chip\_I2S\_Stop
  - i2s\_5411x.h, [737](#)
- Chip\_I2S\_TX\_Init
  - i2s\_5411x.h, [738](#)
- Chip\_IAP\_BlankCheckSector
  - CHIP: Common Chip ISP/IAP commands and return codes, [50](#)
- Chip\_IAP\_Compare
  - CHIP: Common Chip ISP/IAP commands and return codes, [51](#)
- Chip\_IAP\_CopyRamToFlash
  - CHIP: Common Chip ISP/IAP commands and return codes, [51](#)
- Chip\_IAP\_ErasePage
  - CHIP: Common Chip ISP/IAP commands and return codes, [51](#)
- Chip\_IAP\_EraseSector
  - CHIP: Common Chip ISP/IAP commands and return codes, [52](#)
- Chip\_IAP\_PreSectorForReadWrite
  - CHIP: Common Chip ISP/IAP commands and return codes, [52](#)
- Chip\_IAP\_ReadBootCode
  - CHIP: Common Chip ISP/IAP commands and return codes, [52](#)
- Chip\_IAP\_ReadPID
  - CHIP: Common Chip ISP/IAP commands and return codes, [53](#)
- Chip\_IAP\_ReadUID
  - CHIP: Common Chip ISP/IAP commands and return codes, [53](#)
- Chip\_IAP\_ReinvokeISP
  - CHIP: Common Chip ISP/IAP commands and return codes, [53](#)
- Chip\_INMUX\_PinIntSel
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- Chip\_INMUX\_SetDMAOutMux
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- Chip\_INMUX\_SetDMATrigger
  - CHIP: LPC5411X Input Mux Registers and Driver, [196](#)
- Chip\_INMUX\_SetFreqMeasRefClock
  - CHIP: LPC5411X Input Mux Registers and Driver, [196](#)
- Chip\_INMUX\_SetFreqMeasTargClock
  - CHIP: LPC5411X Input Mux Registers and Driver, [196](#)
- Chip\_IOCON\_PinMux
  - CHIP: LPC5411X IOCON register block and driver, [192](#)
- Chip\_IOCON\_PinMuxSet
  - CHIP: LPC5411X IOCON register block and driver, [192](#)
- Chip\_IOCON\_SetPinMuxing
  - CHIP: LPC5411X IOCON register block and driver, [192](#)
- Chip\_MBOX\_ClearValueBits
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [206](#)
- Chip\_MBOX\_DeInit
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [206](#)
- Chip\_MBOX\_GetMutex
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [206](#)
- Chip\_MBOX\_GetValue
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [206](#)

- Chip\_MBOX\_Init
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [208](#)
- Chip\_MBOX\_SetMutex
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [208](#)
- Chip\_MBOX\_SetValue
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [208](#)
- Chip\_MBOX\_SetValueBits
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [209](#)
- Chip\_MRT\_ClearIntPending
  - CHIP: LPC5411X Multi-Rate Timer driver, [217](#)
- Chip\_MRT\_DeInit
  - CHIP: LPC5411X Multi-Rate Timer driver, [217](#)
- Chip\_MRT\_GetEnabled
  - CHIP: LPC5411X Multi-Rate Timer driver, [217](#)
- Chip\_MRT\_GetIdleChannel
  - CHIP: LPC5411X Multi-Rate Timer driver, [218](#)
- Chip\_MRT\_GetIdleChannelShifted
  - CHIP: LPC5411X Multi-Rate Timer driver, [218](#)
- Chip\_MRT\_GetIntPending
  - CHIP: LPC5411X Multi-Rate Timer driver, [218](#)
- Chip\_MRT\_GetIntPendingByChannel
  - CHIP: LPC5411X Multi-Rate Timer driver, [218](#)
- Chip\_MRT\_GetInterval
  - CHIP: LPC5411X Multi-Rate Timer driver, [218](#)
- Chip\_MRT\_GetMode
  - CHIP: LPC5411X Multi-Rate Timer driver, [219](#)
- Chip\_MRT\_GetRegPtr
  - CHIP: LPC5411X Multi-Rate Timer driver, [219](#)
- Chip\_MRT\_GetTimer
  - CHIP: LPC5411X Multi-Rate Timer driver, [219](#)
- Chip\_MRT\_Init
  - CHIP: LPC5411X Multi-Rate Timer driver, [219](#)
- Chip\_MRT\_IntClear
  - CHIP: LPC5411X Multi-Rate Timer driver, [219](#)
- Chip\_MRT\_IntPending
  - CHIP: LPC5411X Multi-Rate Timer driver, [220](#)
- Chip\_MRT\_IsOneShotMode
  - CHIP: LPC5411X Multi-Rate Timer driver, [220](#)
- Chip\_MRT\_IsRepeatMode
  - CHIP: LPC5411X Multi-Rate Timer driver, [220](#)
- Chip\_MRT\_Running
  - CHIP: LPC5411X Multi-Rate Timer driver, [220](#)
- Chip\_MRT\_SetDisabled
  - CHIP: LPC5411X Multi-Rate Timer driver, [221](#)
- Chip\_MRT\_SetEnabled
  - CHIP: LPC5411X Multi-Rate Timer driver, [221](#)
- Chip\_MRT\_SetInterval
  - CHIP: LPC5411X Multi-Rate Timer driver, [221](#)
- Chip\_MRT\_SetMode
  - CHIP: LPC5411X Multi-Rate Timer driver, [221](#)
- Chip\_PININT\_BITSLICE\_CFG\_T
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- Chip\_PININT\_BITSLICE\_T
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- Chip\_PININT\_ClearFallStates
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- Chip\_PININT\_ClearIntStatus
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- Chip\_PININT\_ClearRiseStates
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- Chip\_PININT\_DeInit
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [244](#)
- Chip\_PININT\_DisableIntHigh
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [244](#)
- Chip\_PININT\_DisableIntLow
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [244](#)
- Chip\_PININT\_DisablePatternMatch
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [244](#)
- Chip\_PININT\_DisablePatternMatchRxEv
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [245](#)
- Chip\_PININT\_EnableIntHigh
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [245](#)
- Chip\_PININT\_EnableIntLow
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [245](#)
- Chip\_PININT\_EnablePatternMatch
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [245](#)
- Chip\_PININT\_EnablePatternMatchRxEv
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [246](#)
- Chip\_PININT\_GetFallStates
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [246](#)
- Chip\_PININT\_GetHighEnabled
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [246](#)
- Chip\_PININT\_GetIntStatus
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [246](#)
- Chip\_PININT\_GetLowEnabled
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [247](#)
- Chip\_PININT\_GetPatternMatchState
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [247](#)
- Chip\_PININT\_GetPinMode
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [247](#)
- Chip\_PININT\_GetRiseStates
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [247](#)
- Chip\_PININT\_Init

- CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [248](#)
- Chip\_PININT\_SELECT\_T
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- Chip\_PININT\_SetPatternMatchConfig
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [248](#)
- Chip\_PININT\_SetPatternMatchSrc
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [248](#)
- Chip\_PININT\_SetPinModeEdge
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [249](#)
- Chip\_PININT\_SetPinModeLevel
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [249](#)
- Chip\_PMU\_DisableBODInt
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- Chip\_PMU\_DisableBODReset
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- Chip\_PMU\_EnableBODInt
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- Chip\_PMU\_EnableBODReset
  - CHIP: LPC5411X Power Management declarations and functions, [256](#)
- Chip\_PMU\_SetBODLevels
  - CHIP: LPC5411X Power Management declarations and functions, [256](#)
- Chip\_POWER\_EnterPowerMode
  - CHIP: LPC5411X Power LIBRARY functions, [252](#)
- Chip\_POWER\_GetROMVersion
  - CHIP: LPC5411X Power LIBRARY functions, [252](#)
- Chip\_POWER\_SetFROHFRate
  - CHIP: LPC5411X Power LIBRARY functions, [252](#)
- Chip\_POWER\_SetLowPowerVoltage
  - CHIP: LPC5411X Power LIBRARY functions, [252](#)
- Chip\_POWER\_SetPLL
  - CHIP: LPC5411X Power LIBRARY functions, [253](#)
- Chip\_POWER\_SetVoltage
  - CHIP: LPC5411X Power LIBRARY functions, [253](#)
- Chip\_RTC\_ClearStatus
  - CHIP: LPC5411X Real Time clock, [260](#)
- Chip\_RTC\_DeInit
  - CHIP: LPC5411X Real Time clock, [260](#)
- Chip\_RTC\_Disable
  - CHIP: LPC5411X Real Time clock, [260](#)
- Chip\_RTC\_Disable1KHZ
  - CHIP: LPC5411X Real Time clock, [261](#)
- Chip\_RTC\_DisableOptions
  - CHIP: LPC5411X Real Time clock, [261](#)
- Chip\_RTC\_DisableWakeup
  - CHIP: LPC5411X Real Time clock, [261](#)
- Chip\_RTC\_Enable
  - CHIP: LPC5411X Real Time clock, [262](#)
- Chip\_RTC\_Enable1KHZ
  - CHIP: LPC5411X Real Time clock, [262](#)
- Chip\_RTC\_EnableOptions
  - CHIP: LPC5411X Real Time clock, [262](#)
- Chip\_RTC\_EnableWakeup
  - CHIP: LPC5411X Real Time clock, [263](#)
- Chip\_RTC\_GetAlarm
  - CHIP: LPC5411X Real Time clock, [263](#)
- Chip\_RTC\_GetCount
  - CHIP: LPC5411X Real Time clock, [263](#)
- Chip\_RTC\_GetStatus
  - CHIP: LPC5411X Real Time clock, [264](#)
- Chip\_RTC\_GetWake
  - CHIP: LPC5411X Real Time clock, [264](#)
- Chip\_RTC\_Init
  - CHIP: LPC5411X Real Time clock, [264](#)
- Chip\_RTC\_PowerDown
  - CHIP: LPC5411X Real Time clock, [264](#)
- Chip\_RTC\_PowerUp
  - CHIP: LPC5411X Real Time clock, [265](#)
- Chip\_RTC\_Reset
  - CHIP: LPC5411X Real Time clock, [265](#)
- Chip\_RTC\_SetAlarm
  - CHIP: LPC5411X Real Time clock, [265](#)
- Chip\_RTC\_SetCount
  - CHIP: LPC5411X Real Time clock, [265](#)
- Chip\_RTC\_SetWake
  - CHIP: LPC5411X Real Time clock, [267](#)
- Chip\_SCT\_ClearControl
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- Chip\_SCT\_ClearEventFlag
  - CHIP: LPC5411X State Configurable Timer driver, [318](#)
- Chip\_SCT\_Config
  - CHIP: LPC5411X State Configurable Timer driver, [318](#)
- Chip\_SCT\_DeInit
  - CHIP: LPC5411X State Configurable Timer driver, [318](#)
- Chip\_SCT\_DisableEventInt
  - CHIP: LPC5411X State Configurable Timer driver, [318](#)
- Chip\_SCT\_EnableEventInt
  - CHIP: LPC5411X State Configurable Timer driver, [319](#)
- Chip\_SCT\_EventControl
  - CHIP: LPC5411X State Configurable Timer driver, [319](#)
- Chip\_SCT\_EventStateMask
  - CHIP: LPC5411X State Configurable Timer driver, [319](#)
- Chip\_SCT\_Init
  - CHIP: LPC5411X State Configurable Timer driver, [319](#)
- Chip\_SCT\_Limit
  - CHIP: LPC5411X State Configurable Timer driver, [321](#)



- Chip\_SCT\_SetClrControl
  - CHIP: LPC5411X State Configurable Timer driver, [321](#)
- Chip\_SCT\_SetConflictResolution
  - CHIP: LPC5411X State Configurable Timer driver, [321](#)
- Chip\_SCT\_SetControl
  - CHIP: LPC5411X State Configurable Timer driver, [322](#)
- Chip\_SCT\_SetCount
  - CHIP: LPC5411X State Configurable Timer driver, [322](#)
- Chip\_SCT\_SetCountH
  - CHIP: LPC5411X State Configurable Timer driver, [322](#)
- Chip\_SCT\_SetCountL
  - CHIP: LPC5411X State Configurable Timer driver, [322](#)
- Chip\_SCT\_SetMatchCount
  - CHIP: LPC5411X State Configurable Timer driver, [323](#)
- Chip\_SCT\_SetMatchReload
  - CHIP: LPC5411X State Configurable Timer driver, [323](#)
- Chip\_SCTPWM\_GetDutyCycle
  - CHIP: LPC5411X State Configurable Timer PWM driver, [303](#)
- Chip\_SCTPWM\_GetTicksPerCycle
  - CHIP: LPC5411X State Configurable Timer PWM driver, [303](#)
- Chip\_SCTPWM\_Init
  - CHIP: LPC5411X State Configurable Timer PWM driver, [304](#)
- Chip\_SCTPWM\_PercentageToTicks
  - CHIP: LPC5411X State Configurable Timer PWM driver, [304](#)
- Chip\_SCTPWM\_SetDutyCycle
  - CHIP: LPC5411X State Configurable Timer PWM driver, [304](#)
- Chip\_SCTPWM\_SetOutPin
  - CHIP: LPC5411X State Configurable Timer PWM driver, [305](#)
- Chip\_SCTPWM\_SetRate
  - CHIP: LPC5411X State Configurable Timer PWM driver, [305](#)
- Chip\_SCTPWM\_Start
  - CHIP: LPC5411X State Configurable Timer PWM driver, [305](#)
- Chip\_SCTPWM\_Stop
  - CHIP: LPC5411X State Configurable Timer PWM driver, [306](#)
- Chip\_SPI\_ClearCFGRegBits
  - CHIP: LPC5411X SPI driver, [280](#)
- Chip\_SPI\_ClearFIFOCfg
  - CHIP: LPC5411X SPI driver, [281](#)
- Chip\_SPI\_ClearFIFOStatus
  - CHIP: LPC5411X SPI driver, [281](#)
- Chip\_SPI\_ClearStatus
  - CHIP: LPC5411X SPI driver, [281](#)
- Chip\_SPI\_ConfigureSPI
  - CHIP: LPC5411X SPI driver, [281](#)
- Chip\_SPI\_DeInit
  - CHIP: LPC5411X SPI driver, [283](#)
- Chip\_SPI\_Disable
  - CHIP: LPC5411X SPI driver, [283](#)
- Chip\_SPI\_DisableFIFOInts
  - CHIP: LPC5411X SPI driver, [283](#)
- Chip\_SPI\_DisableInts
  - CHIP: LPC5411X SPI driver, [283](#)
- Chip\_SPI\_Enable
  - CHIP: LPC5411X SPI driver, [284](#)
- Chip\_SPI\_EnableFIFOInts
  - CHIP: LPC5411X SPI driver, [284](#)
- Chip\_SPI\_EnableInts
  - CHIP: LPC5411X SPI driver, [284](#)
- Chip\_SPI\_EnableLSBFirst
  - CHIP: LPC5411X SPI driver, [284](#)
- Chip\_SPI\_EnableMSBFirst
  - CHIP: LPC5411X SPI driver, [285](#)
- Chip\_SPI\_EnableSlaveMode
  - CHIP: LPC5411X SPI driver, [285](#)
- Chip\_SPI\_FlushFIFOs
  - CHIP: LPC5411X SPI driver, [272](#)
- Chip\_SPI\_FlushFifos
  - CHIP: LPC5411X SPI driver, [285](#)
- Chip\_SPI\_GetEnabledInts
  - CHIP: LPC5411X SPI driver, [285](#)
- Chip\_SPI\_GetFIFOEnabledInts
  - CHIP: LPC5411X SPI driver, [286](#)
- Chip\_SPI\_GetFIFOPendingInts
  - CHIP: LPC5411X SPI driver, [286](#)
- Chip\_SPI\_GetFIFOStatus
  - CHIP: LPC5411X SPI driver, [286](#)
- Chip\_SPI\_GetFIFOTrigLevel
  - CHIP: LPC5411X SPI driver, [286](#)
- Chip\_SPI\_GetPendingInts
  - CHIP: LPC5411X SPI driver, [288](#)
- Chip\_SPI\_GetStatus
  - CHIP: LPC5411X SPI driver, [288](#)
- Chip\_SPI\_Init
  - CHIP: LPC5411X SPI driver, [288](#)
- Chip\_SPI\_ReadFIFO
  - CHIP: LPC5411X SPI driver, [272](#)
- Chip\_SPI\_ReadFIFOdata
  - CHIP: LPC5411X SPI driver, [272](#)
- Chip\_SPI\_ReadRXData
  - CHIP: LPC5411X SPI driver, [289](#)
- Chip\_SPI\_ReadRawRXFifo
  - CHIP: LPC5411X SPI driver, [288](#)
- Chip\_SPI\_SetCFGRegBits
  - CHIP: LPC5411X SPI driver, [289](#)
- Chip\_SPI\_SetCSPolHigh
  - CHIP: LPC5411X SPI driver, [289](#)
- Chip\_SPI\_SetCSPolLow
  - CHIP: LPC5411X SPI driver, [290](#)
- Chip\_SPI\_SetFIFOCfg

- CHIP: LPC5411X SPI driver, [290](#)
- Chip\_SPI\_SetFIFOTrigLevel
  - CHIP: LPC5411X SPI driver, [290](#)
- Chip\_SPI\_SetSPIMode
  - CHIP: LPC5411X SPI driver, [291](#)
- Chip\_SPI\_SetTXCTRLData
  - CHIP: LPC5411X SPI driver, [291](#)
- Chip\_SPI\_WriteFIFO
  - CHIP: LPC5411X SPI driver, [291](#)
- Chip\_SPI\_WriteFIFOcd
  - CHIP: LPC5411X SPI driver, [272](#)
- Chip\_SPI\_WriteFIFOdata
  - CHIP: LPC5411X SPI driver, [272](#)
- Chip\_SPI\_WriteTXData
  - CHIP: LPC5411X SPI driver, [291](#)
- Chip\_SPIM\_DelayConfig
  - CHIP: LPC5411X SPI master driver, [295](#)
- Chip\_SPIM\_DisableLoopBack
  - CHIP: LPC5411X SPI master driver, [296](#)
- Chip\_SPIM\_EnableLoopBack
  - CHIP: LPC5411X SPI master driver, [296](#)
- Chip\_SPIM\_ForceEndOfTransfer
  - CHIP: LPC5411X SPI master driver, [296](#)
- Chip\_SPIM\_GetClockRate
  - CHIP: LPC5411X SPI master driver, [296](#)
- Chip\_SPIM\_SetClockRate
  - CHIP: LPC5411X SPI master driver, [297](#)
- Chip\_SPIM\_Xfer
  - CHIP: LPC5411X SPI master driver, [297](#)
- Chip\_SPIM\_XferBlocking
  - CHIP: LPC5411X SPI master driver, [298](#)
- Chip\_SPIM\_XferFIFO
  - CHIP: LPC5411X SPI master driver, [298](#)
- Chip\_SPIM\_XferHandler
  - CHIP: LPC5411X SPI master driver, [299](#)
- Chip\_SPIS\_DisableInts
  - CHIP: LPC5411X SPI slave driver, [300](#)
- Chip\_SPIS\_EnableInts
  - CHIP: LPC5411X SPI slave driver, [301](#)
- Chip\_SPIS\_Init
  - CHIP: LPC5411X SPI slave driver, [301](#)
- Chip\_SPIS\_LoadFIFO
  - CHIP: LPC5411X SPI slave driver, [301](#)
- Chip\_SPIS\_ReadFIFO
  - CHIP: LPC5411X SPI slave driver, [302](#)
- Chip\_SPIS\_XferHandler
  - CHIP: LPC5411X SPI slave driver, [302](#)
- Chip\_SYSCON\_ClearSystemRSTStatus
  - CHIP: LPC5411X System and Control Driver, [333](#)
- Chip\_SYSCON\_DisableAutoClocking
  - CHIP: LPC5411X System and Control Driver, [333](#)
- Chip\_SYSCON\_DisableNMISource
  - CHIP: LPC5411X System and Control Driver, [333](#)
- Chip\_SYSCON\_DisableWakeup
  - CHIP: LPC5411X System and Control Driver, [334](#)
- Chip\_SYSCON\_Enable\_ASYNC\_Syscon
  - CHIP: LPC5411X System and Control Driver, [334](#)
- Chip\_SYSCON\_EnableAutoClocking
  - CHIP: LPC5411X System and Control Driver, [334](#)
- Chip\_SYSCON\_EnableNMISource
  - CHIP: LPC5411X System and Control Driver, [334](#)
- Chip\_SYSCON\_EnableWakeup
  - CHIP: LPC5411X System and Control Driver, [335](#)
- Chip\_SYSCON\_GetCompFreqMeas
  - CHIP: LPC5411X System and Control Driver, [335](#)
- Chip\_SYSCON\_GetDeviceID
  - CHIP: LPC5411X System and Control Driver, [335](#)
- Chip\_SYSCON\_GetMemoryMap
  - CHIP: LPC5411X System and Control Driver, [335](#)
- Chip\_SYSCON\_GetPORPIOSStatus
  - CHIP: LPC5411X System and Control Driver, [335](#)
- Chip\_SYSCON\_GetPowerStates
  - CHIP: LPC5411X System and Control Driver, [337](#)
- Chip\_SYSCON\_GetRawFreqMeasCapval
  - CHIP: LPC5411X System and Control Driver, [337](#)
- Chip\_SYSCON\_GetResetPIOStatus
  - CHIP: LPC5411X System and Control Driver, [337](#)
- Chip\_SYSCON\_GetSystemRSTStatus
  - CHIP: LPC5411X System and Control Driver, [337](#)
- Chip\_SYSCON\_IsFreqMeasComplete
  - CHIP: LPC5411X System and Control Driver, [338](#)
- Chip\_SYSCON\_Map
  - CHIP: LPC5411X System and Control Driver, [338](#)
- Chip\_SYSCON\_PeriphReset
  - CHIP: LPC5411X System and Control Driver, [338](#)
- Chip\_SYSCON\_PowerDown
  - CHIP: LPC5411X System and Control Driver, [338](#)
- Chip\_SYSCON\_PowerUp
  - CHIP: LPC5411X System and Control Driver, [339](#)
- Chip\_SYSCON\_SetFLASHAccess
  - CHIP: LPC5411X System and Control Driver, [339](#)
- Chip\_SYSCON\_SetNMISource
  - CHIP: LPC5411X System and Control Driver, [339](#)
- Chip\_SYSCON\_SetSYSTCKCAL
  - CHIP: LPC5411X System and Control Driver, [339](#)
- Chip\_SYSCON\_SetUSARTFRGCtrl
  - CHIP: LPC5411X System and Control Driver, [340](#)
- Chip\_SYSCON\_StartFreqMeas
  - CHIP: LPC5411X System and Control Driver, [340](#)
- Chip\_SetupExtInClocking
  - CHIP: LPC5411X support functions, [385](#)
- Chip\_SetupFROClocking
  - CHIP: LPC5411X support functions, [385](#)
- Chip\_SetupIrcClocking
  - CHIP: LPC5411X support functions, [385](#)
- Chip\_SystemInit
  - CHIP: LPC5411X support functions, [386](#)
- Chip\_TIMER\_CaptureDisableInt
  - CHIP: LPC5411X 32-bit Timer driver, [61](#)
- Chip\_TIMER\_CaptureEnableInt
  - CHIP: LPC5411X 32-bit Timer driver, [61](#)
- Chip\_TIMER\_CaptureFallingEdgeDisable
  - CHIP: LPC5411X 32-bit Timer driver, [61](#)
- Chip\_TIMER\_CaptureFallingEdgeEnable
  - CHIP: LPC5411X 32-bit Timer driver, [62](#)
- Chip\_TIMER\_CapturePending

- CHIP: LPC5411X 32-bit Timer driver, [62](#)
- Chip\_TIMER\_CaptureRisingEdgeDisable
  - CHIP: LPC5411X 32-bit Timer driver, [62](#)
- Chip\_TIMER\_CaptureRisingEdgeEnable
  - CHIP: LPC5411X 32-bit Timer driver, [62](#)
- Chip\_TIMER\_ClearCapture
  - CHIP: LPC5411X 32-bit Timer driver, [64](#)
- Chip\_TIMER\_ClearMatch
  - CHIP: LPC5411X 32-bit Timer driver, [64](#)
- Chip\_TIMER\_DeInit
  - CHIP: LPC5411X 32-bit Timer driver, [64](#)
- Chip\_TIMER\_Disable
  - CHIP: LPC5411X 32-bit Timer driver, [65](#)
- Chip\_TIMER\_Enable
  - CHIP: LPC5411X 32-bit Timer driver, [65](#)
- Chip\_TIMER\_ExtMatchControlSet
  - CHIP: LPC5411X 32-bit Timer driver, [65](#)
- Chip\_TIMER\_Init
  - CHIP: LPC5411X 32-bit Timer driver, [66](#)
- Chip\_TIMER\_MatchDisableInt
  - CHIP: LPC5411X 32-bit Timer driver, [66](#)
- Chip\_TIMER\_MatchEnableInt
  - CHIP: LPC5411X 32-bit Timer driver, [66](#)
- Chip\_TIMER\_MatchPending
  - CHIP: LPC5411X 32-bit Timer driver, [67](#)
- Chip\_TIMER\_PrescaleSet
  - CHIP: LPC5411X 32-bit Timer driver, [67](#)
- Chip\_TIMER\_ReadCapture
  - CHIP: LPC5411X 32-bit Timer driver, [67](#)
- Chip\_TIMER\_ReadCount
  - CHIP: LPC5411X 32-bit Timer driver, [68](#)
- Chip\_TIMER\_ReadPrescale
  - CHIP: LPC5411X 32-bit Timer driver, [68](#)
- Chip\_TIMER\_Reset
  - CHIP: LPC5411X 32-bit Timer driver, [68](#)
- Chip\_TIMER\_ResetOnMatchDisable
  - CHIP: LPC5411X 32-bit Timer driver, [69](#)
- Chip\_TIMER\_ResetOnMatchEnable
  - CHIP: LPC5411X 32-bit Timer driver, [69](#)
- Chip\_TIMER\_SetMatch
  - CHIP: LPC5411X 32-bit Timer driver, [69](#)
- Chip\_TIMER\_StopOnMatchDisable
  - CHIP: LPC5411X 32-bit Timer driver, [69](#)
- Chip\_TIMER\_StopOnMatchEnable
  - CHIP: LPC5411X 32-bit Timer driver, [71](#)
- Chip\_TIMER\_TIMER\_SetCountClockSrc
  - CHIP: LPC5411X 32-bit Timer driver, [71](#)
- Chip\_UART\_AutoBaud
  - CHIP: LPC5411X UART Driver, [353](#)
- Chip\_UART\_CalculateBaud
  - uart\_5411x.c, [804](#)
- Chip\_UART\_ClearFIFOCfg
  - CHIP: LPC5411X UART Driver, [354](#)
- Chip\_UART\_ClearFIFOStatus
  - CHIP: LPC5411X UART Driver, [354](#)
- Chip\_UART\_ClearStatus
  - CHIP: LPC5411X UART Driver, [354](#)
- Chip\_UART\_ConfigDMA
  - CHIP: LPC5411X UART Driver, [356](#)
- Chip\_UART\_ConfigData
  - CHIP: LPC5411X UART Driver, [354](#)
- Chip\_UART\_DeInit
  - CHIP: LPC5411X UART Driver, [356](#)
- Chip\_UART\_Disable
  - CHIP: LPC5411X UART Driver, [356](#)
- Chip\_UART\_DisableFIFOInts
  - CHIP: LPC5411X UART Driver, [356](#)
- Chip\_UART\_Enable
  - CHIP: LPC5411X UART Driver, [358](#)
- Chip\_UART\_EnableFIFOInts
  - CHIP: LPC5411X UART Driver, [358](#)
- Chip\_UART\_FlushFIFOs
  - CHIP: LPC5411X UART Driver, [358](#)
- Chip\_UART\_GetFIFOEnabledInts
  - CHIP: LPC5411X UART Driver, [358](#)
- Chip\_UART\_GetFIFOPendingInts
  - CHIP: LPC5411X UART Driver, [359](#)
- Chip\_UART\_GetFIFOStatus
  - CHIP: LPC5411X UART Driver, [359](#)
- Chip\_UART\_GetIntStatus
  - CHIP: LPC5411X UART Driver, [359](#)
- Chip\_UART\_GetIntsEnabled
  - CHIP: LPC5411X UART Driver, [359](#)
- Chip\_UART\_GetStatus
  - CHIP: LPC5411X UART Driver, [361](#)
- Chip\_UART\_IRQHandlerDMA
  - CHIP: LPC5411X UART Driver, [362](#)
- Chip\_UART\_IRQHandlerRB
  - CHIP: LPC5411X UART Driver, [362](#)
- Chip\_UART\_Init
  - CHIP: LPC5411X UART Driver, [361](#)
- Chip\_UART\_IntDisable
  - CHIP: LPC5411X UART Driver, [361](#)
- Chip\_UART\_IntEnable
  - CHIP: LPC5411X UART Driver, [362](#)
- Chip\_UART\_RXIntHandlerRB
  - CHIP: LPC5411X UART Driver, [365](#)
- Chip\_UART\_Read
  - CHIP: LPC5411X UART Driver, [363](#)
- Chip\_UART\_ReadBlocking
  - CHIP: LPC5411X UART Driver, [363](#)
- Chip\_UART\_ReadByte
  - CHIP: LPC5411X UART Driver, [363](#)
- Chip\_UART\_ReadRB
  - CHIP: LPC5411X UART Driver, [365](#)
- Chip\_UART\_Send
  - CHIP: LPC5411X UART Driver, [365](#)
- Chip\_UART\_SendBlocking
  - CHIP: LPC5411X UART Driver, [366](#)
- Chip\_UART\_SendByte
  - CHIP: LPC5411X UART Driver, [366](#)
- Chip\_UART\_SendRB
  - CHIP: LPC5411X UART Driver, [366](#)
- Chip\_UART\_SetBaud
  - CHIP: LPC5411X UART Driver, [368](#)
- Chip\_UART\_SetFIFOCfg



- CHIP: LPC5411X UART Driver, [368](#)
- Chip\_UART\_SetFIFOTrigLevel
  - CHIP: LPC5411X UART Driver, [368](#)
- Chip\_UART\_TXDisable
  - CHIP: LPC5411X UART Driver, [369](#)
- Chip\_UART\_TXEnable
  - CHIP: LPC5411X UART Driver, [369](#)
- Chip\_UART\_TXIntHandlerRB
  - CHIP: LPC5411X UART Driver, [369](#)
- Chip\_USB\_Init
  - CHIP: LPC5411X support functions, [386](#)
- Chip\_USB\_TrimOff
  - CHIP: LPC5411X support functions, [386](#)
- Chip\_UTICK\_ClearInterrupt
  - CHIP: LPC5411X Micro Tick driver, [211](#)
- Chip\_UTICK\_DeInit
  - CHIP: LPC5411X Micro Tick driver, [211](#)
- Chip\_UTICK\_GetStatus
  - CHIP: LPC5411X Micro Tick driver, [211](#)
- Chip\_UTICK\_GetTick
  - CHIP: LPC5411X Micro Tick driver, [212](#)
- Chip\_UTICK\_Halt
  - CHIP: LPC5411X Micro Tick driver, [212](#)
- Chip\_UTICK\_Init
  - CHIP: LPC5411X Micro Tick driver, [212](#)
- Chip\_UTICK\_SetDelayMs
  - CHIP: LPC5411X Micro Tick driver, [212](#)
- Chip\_UTICK\_SetTick
  - CHIP: LPC5411X Micro Tick driver, [213](#)
- Chip\_WWDT\_ClearStatusFlag
  - CHIP: LPC5411X Windowed Watchdog driver, [372](#)
- Chip\_WWDT\_DeInit
  - CHIP: LPC5411X Windowed Watchdog driver, [373](#)
- Chip\_WWDT\_Feed
  - CHIP: LPC5411X Windowed Watchdog driver, [373](#)
- Chip\_WWDT\_GetCurrentCount
  - CHIP: LPC5411X Windowed Watchdog driver, [373](#)
- Chip\_WWDT\_GetStatus
  - CHIP: LPC5411X Windowed Watchdog driver, [374](#)
- Chip\_WWDT\_GetWarning
  - CHIP: LPC5411X Windowed Watchdog driver, [374](#)
- Chip\_WWDT\_GetWindow
  - CHIP: LPC5411X Windowed Watchdog driver, [374](#)
- Chip\_WWDT\_Init
  - CHIP: LPC5411X Windowed Watchdog driver, [374](#)
- Chip\_WWDT\_SetOption
  - CHIP: LPC5411X Windowed Watchdog driver, [375](#)
- Chip\_WWDT\_SetTimeOut
  - CHIP: LPC5411X Windowed Watchdog driver, [375](#)
- Chip\_WWDT\_SetWarning
  - CHIP: LPC5411X Windowed Watchdog driver, [375](#)
- Chip\_WWDT\_SetWindow
  - CHIP: LPC5411X Windowed Watchdog driver, [376](#)
- Chip\_WWDT\_Start
  - CHIP: LPC5411X Windowed Watchdog driver, [376](#)
- Chip\_WWDT\_UnsetOption
  - CHIP: LPC5411X Windowed Watchdog driver, [376](#)
- clk
  - UART\_BAUD\_T, [648](#)
- clock\_5411x.c
  - Chip\_Clock\_GetAsyncSyscon\_ClockRate\_NoDiv, [777](#)
  - Chip\_Clock\_GetMCLKClockRate, [777](#)
  - mclk\_in\_rate, [777](#)
  - WDT\_FREQ\_LOOKUP, [777](#)
- Code Red LPCXpresso support in LPCOpen, [438](#)
- Common components used with chip drivers, [440](#)
- Common FreeRTOS functions shared with multiple platforms, [439](#)
- Community support for LPCOpen, [441](#)
- ConvertRtcTime
  - CHIP: RTC tick to (a more) Universal Time conversion functions, [417](#)
- ConvertTimeRtc
  - CHIP: RTC tick to (a more) Universal Time conversion functions, [417](#)
- copy
  - RINGBUFF\_T, [633](#)
- Copyright (C) 2013 NXP Semiconductors. All rights reserved., [442](#)
- Core Debug Registers (CoreDebug), [443](#)
  - CoreDebug\_DCRSR\_REGSEL\_Msk, [444](#)
  - CoreDebug\_DCRSR\_REGSEL\_Pos, [444](#)
  - CoreDebug\_DCRSR\_REGWnR\_Msk, [444](#)
  - CoreDebug\_DCRSR\_REGWnR\_Pos, [444](#)
  - CoreDebug\_DEMCR\_MON\_EN\_Msk, [444](#)
  - CoreDebug\_DEMCR\_MON\_EN\_Pos, [444](#)
  - CoreDebug\_DEMCR\_MON\_PEND\_Msk, [444](#)
  - CoreDebug\_DEMCR\_MON\_PEND\_Pos, [445](#)
  - CoreDebug\_DEMCR\_MON\_REQ\_Msk, [445](#)
  - CoreDebug\_DEMCR\_MON\_REQ\_Pos, [445](#)
  - CoreDebug\_DEMCR\_MON\_STEP\_Msk, [445](#)
  - CoreDebug\_DEMCR\_MON\_STEP\_Pos, [445](#)
  - CoreDebug\_DEMCR\_TRCENA\_Msk, [445](#)
  - CoreDebug\_DEMCR\_TRCENA\_Pos, [445](#)
  - CoreDebug\_DEMCR\_VC\_BUSERR\_Msk, [445](#)
  - CoreDebug\_DEMCR\_VC\_BUSERR\_Pos, [445](#)
  - CoreDebug\_DEMCR\_VC\_CHKERR\_Msk, [446](#)
  - CoreDebug\_DEMCR\_VC\_CHKERR\_Pos, [446](#)
  - CoreDebug\_DEMCR\_VC\_CORERESET\_Msk, [446](#)
  - CoreDebug\_DEMCR\_VC\_CORERESET\_Pos, [446](#)
  - CoreDebug\_DEMCR\_VC\_HARDERR\_Msk, [446](#)
  - CoreDebug\_DEMCR\_VC\_HARDERR\_Pos, [446](#)
  - CoreDebug\_DEMCR\_VC\_INTERR\_Msk, [446](#)
  - CoreDebug\_DEMCR\_VC\_INTERR\_Pos, [446](#)
  - CoreDebug\_DEMCR\_VC\_MMERR\_Msk, [446](#)
  - CoreDebug\_DEMCR\_VC\_MMERR\_Pos, [447](#)
  - CoreDebug\_DEMCR\_VC\_NOCERR\_Msk, [447](#)
  - CoreDebug\_DEMCR\_VC\_NOCERR\_Pos, [447](#)
  - CoreDebug\_DEMCR\_VC\_STATERR\_Msk, [447](#)
  - CoreDebug\_DEMCR\_VC\_STATERR\_Pos, [447](#)
  - CoreDebug\_DHCSR\_C\_DEBUGEN\_Msk, [447](#)
  - CoreDebug\_DHCSR\_C\_DEBUGEN\_Pos, [447](#)
  - CoreDebug\_DHCSR\_C\_HALT\_Msk, [447](#)
  - CoreDebug\_DHCSR\_C\_HALT\_Pos, [447](#)

- CoreDebug\_DHCSR\_C\_MASKINTS\_Msk, [448](#)
- CoreDebug\_DHCSR\_C\_MASKINTS\_Pos, [448](#)
- CoreDebug\_DHCSR\_C\_SNAPSTALL\_Msk, [448](#)
- CoreDebug\_DHCSR\_C\_SNAPSTALL\_Pos, [448](#)
- CoreDebug\_DHCSR\_C\_STEP\_Msk, [448](#)
- CoreDebug\_DHCSR\_C\_STEP\_Pos, [448](#)
- CoreDebug\_DHCSR\_DBGKEY\_Msk, [448](#)
- CoreDebug\_DHCSR\_DBGKEY\_Pos, [448](#)
- CoreDebug\_DHCSR\_S\_HALT\_Msk, [448](#)
- CoreDebug\_DHCSR\_S\_HALT\_Pos, [449](#)
- CoreDebug\_DHCSR\_S\_LOCKUP\_Msk, [449](#)
- CoreDebug\_DHCSR\_S\_LOCKUP\_Pos, [449](#)
- CoreDebug\_DHCSR\_S\_REGRDY\_Msk, [449](#)
- CoreDebug\_DHCSR\_S\_REGRDY\_Pos, [449](#)
- CoreDebug\_DHCSR\_S\_RESET\_ST\_Msk, [449](#)
- CoreDebug\_DHCSR\_S\_RESET\_ST\_Pos, [449](#)
- CoreDebug\_DHCSR\_S\_RETIRE\_ST\_Msk, [449](#)
- CoreDebug\_DHCSR\_S\_RETIRE\_ST\_Pos, [449](#)
- CoreDebug\_DHCSR\_S\_SLEEP\_Msk, [450](#)
- CoreDebug\_DHCSR\_S\_SLEEP\_Pos, [450](#)
- Core Definitions, [451](#)
- CoreDebug, [451](#)
- CoreDebug\_BASE, [451](#)
- DWT, [451](#)
- DWT\_BASE, [451](#)
- ITM, [452](#)
- ITM\_BASE, [452](#)
- NVIC, [452](#)
- NVIC\_BASE, [452](#)
- SCB, [452](#)
- SCB\_BASE, [452](#), [453](#)
- SCS\_BASE, [453](#)
- SCnSCB, [453](#)
- SysTick, [453](#)
- SysTick\_BASE, [453](#)
- TPI, [453](#)
- TPI\_BASE, [454](#)
- core\_cm0plus.h
  - \_\_CM0PLUS\_CMSIS\_VERSION, [684](#)
  - \_\_CM0PLUS\_CMSIS\_VERSION\_MAIN, [684](#)
  - \_\_CM0PLUS\_CMSIS\_VERSION\_SUB, [684](#)
  - \_\_CORE\_CM0PLUS\_H\_DEPENDANT, [684](#)
  - \_\_CORE\_CM0PLUS\_H\_GENERIC, [684](#)
  - \_\_CORTEX\_M, [684](#)
  - \_\_FPU\_USED, [684](#)
  - \_\_I, [684](#)
  - \_\_IO, [684](#)
  - \_\_O, [685](#)
- core\_cm4.h
  - \_\_CM4\_CMSIS\_VERSION, [694](#)
  - \_\_CM4\_CMSIS\_VERSION\_MAIN, [694](#)
  - \_\_CM4\_CMSIS\_VERSION\_SUB, [694](#)
  - \_\_CORE\_CM4\_H\_DEPENDANT, [694](#)
  - \_\_CORE\_CM4\_H\_GENERIC, [694](#)
  - \_\_CORTEX\_M, [694](#)
  - \_\_I, [695](#)
  - \_\_IO, [695](#)
  - \_\_O, [695](#)
- core\_cm4\_simd.h
  - \_\_CORE\_CM4\_SIMD\_H, [695](#)
- CoreDebug
  - Core Definitions, [451](#)
- CoreDebug\_BASE
  - Core Definitions, [451](#)
- CoreDebug\_DCRSR\_REGSEL\_Msk
  - Core Debug Registers (CoreDebug), [444](#)
- CoreDebug\_DCRSR\_REGSEL\_Pos
  - Core Debug Registers (CoreDebug), [444](#)
- CoreDebug\_DCRSR\_REGWnR\_Msk
  - Core Debug Registers (CoreDebug), [444](#)
- CoreDebug\_DCRSR\_REGWnR\_Pos
  - Core Debug Registers (CoreDebug), [444](#)
- CoreDebug\_DEMCR\_MON\_EN\_Msk
  - Core Debug Registers (CoreDebug), [444](#)
- CoreDebug\_DEMCR\_MON\_EN\_Pos
  - Core Debug Registers (CoreDebug), [444](#)
- CoreDebug\_DEMCR\_MON\_PEND\_Msk
  - Core Debug Registers (CoreDebug), [444](#)
- CoreDebug\_DEMCR\_MON\_PEND\_Pos
  - Core Debug Registers (CoreDebug), [445](#)
- CoreDebug\_DEMCR\_MON\_REQ\_Msk
  - Core Debug Registers (CoreDebug), [445](#)
- CoreDebug\_DEMCR\_MON\_REQ\_Pos
  - Core Debug Registers (CoreDebug), [445](#)
- CoreDebug\_DEMCR\_MON\_STEP\_Msk
  - Core Debug Registers (CoreDebug), [445](#)
- CoreDebug\_DEMCR\_MON\_STEP\_Pos
  - Core Debug Registers (CoreDebug), [445](#)
- CoreDebug\_DEMCR\_TRCENA\_Msk
  - Core Debug Registers (CoreDebug), [445](#)
- CoreDebug\_DEMCR\_TRCENA\_Pos
  - Core Debug Registers (CoreDebug), [445](#)
- CoreDebug\_DEMCR\_VC\_BUSERR\_Msk
  - Core Debug Registers (CoreDebug), [445](#)
- CoreDebug\_DEMCR\_VC\_BUSERR\_Pos
  - Core Debug Registers (CoreDebug), [445](#)
- CoreDebug\_DEMCR\_VC\_CHKERR\_Msk
  - Core Debug Registers (CoreDebug), [446](#)
- CoreDebug\_DEMCR\_VC\_CHKERR\_Pos
  - Core Debug Registers (CoreDebug), [446](#)
- CoreDebug\_DEMCR\_VC\_CORERESET\_Msk
  - Core Debug Registers (CoreDebug), [446](#)
- CoreDebug\_DEMCR\_VC\_CORERESET\_Pos
  - Core Debug Registers (CoreDebug), [446](#)
- CoreDebug\_DEMCR\_VC\_HARDERR\_Msk
  - Core Debug Registers (CoreDebug), [446](#)
- CoreDebug\_DEMCR\_VC\_HARDERR\_Pos
  - Core Debug Registers (CoreDebug), [446](#)
- CoreDebug\_DEMCR\_VC\_INTERR\_Msk
  - Core Debug Registers (CoreDebug), [446](#)
- CoreDebug\_DEMCR\_VC\_INTERR\_Pos
  - Core Debug Registers (CoreDebug), [446](#)
- CoreDebug\_DEMCR\_VC\_MMERR\_Msk
  - Core Debug Registers (CoreDebug), [446](#)
- CoreDebug\_DEMCR\_VC\_MMERR\_Pos
  - Core Debug Registers (CoreDebug), [447](#)

- CoreDebug\_DEMCR\_VC\_NOCERR\_Msk
  - Core Debug Registers (CoreDebug), [447](#)
- CoreDebug\_DEMCR\_VC\_NOCERR\_Pos
  - Core Debug Registers (CoreDebug), [447](#)
- CoreDebug\_DEMCR\_VC\_STATERR\_Msk
  - Core Debug Registers (CoreDebug), [447](#)
- CoreDebug\_DEMCR\_VC\_STATERR\_Pos
  - Core Debug Registers (CoreDebug), [447](#)
- CoreDebug\_DHCSR\_C\_DEBUGEN\_Msk
  - Core Debug Registers (CoreDebug), [447](#)
- CoreDebug\_DHCSR\_C\_DEBUGEN\_Pos
  - Core Debug Registers (CoreDebug), [447](#)
- CoreDebug\_DHCSR\_C\_HALT\_Msk
  - Core Debug Registers (CoreDebug), [447](#)
- CoreDebug\_DHCSR\_C\_HALT\_Pos
  - Core Debug Registers (CoreDebug), [447](#)
- CoreDebug\_DHCSR\_C\_MASKINTS\_Msk
  - Core Debug Registers (CoreDebug), [448](#)
- CoreDebug\_DHCSR\_C\_MASKINTS\_Pos
  - Core Debug Registers (CoreDebug), [448](#)
- CoreDebug\_DHCSR\_C\_SNAPSTALL\_Msk
  - Core Debug Registers (CoreDebug), [448](#)
- CoreDebug\_DHCSR\_C\_SNAPSTALL\_Pos
  - Core Debug Registers (CoreDebug), [448](#)
- CoreDebug\_DHCSR\_C\_STEP\_Msk
  - Core Debug Registers (CoreDebug), [448](#)
- CoreDebug\_DHCSR\_C\_STEP\_Pos
  - Core Debug Registers (CoreDebug), [448](#)
- CoreDebug\_DHCSR\_DBGKEY\_Msk
  - Core Debug Registers (CoreDebug), [448](#)
- CoreDebug\_DHCSR\_DBGKEY\_Pos
  - Core Debug Registers (CoreDebug), [448](#)
- CoreDebug\_DHCSR\_S\_HALT\_Msk
  - Core Debug Registers (CoreDebug), [448](#)
- CoreDebug\_DHCSR\_S\_HALT\_Pos
  - Core Debug Registers (CoreDebug), [449](#)
- CoreDebug\_DHCSR\_S\_LOCKUP\_Msk
  - Core Debug Registers (CoreDebug), [449](#)
- CoreDebug\_DHCSR\_S\_LOCKUP\_Pos
  - Core Debug Registers (CoreDebug), [449](#)
- CoreDebug\_DHCSR\_S\_REGRDY\_Msk
  - Core Debug Registers (CoreDebug), [449](#)
- CoreDebug\_DHCSR\_S\_REGRDY\_Pos
  - Core Debug Registers (CoreDebug), [449](#)
- CoreDebug\_DHCSR\_S\_RESET\_ST\_Msk
  - Core Debug Registers (CoreDebug), [449](#)
- CoreDebug\_DHCSR\_S\_RESET\_ST\_Pos
  - Core Debug Registers (CoreDebug), [449](#)
- CoreDebug\_DHCSR\_S\_RETIRE\_ST\_Msk
  - Core Debug Registers (CoreDebug), [449](#)
- CoreDebug\_DHCSR\_S\_RETIRE\_ST\_Pos
  - Core Debug Registers (CoreDebug), [449](#)
- CoreDebug\_DHCSR\_S\_SLEEP\_Msk
  - Core Debug Registers (CoreDebug), [450](#)
- CoreDebug\_DHCSR\_S\_SLEEP\_Pos
  - Core Debug Registers (CoreDebug), [450](#)
- CoreDebug\_Type, [553](#)
  - DCRDR, [553](#)
  - DCRSR, [553](#)
  - DEMCR, [553](#)
  - DHCSR, [553](#)
- count
  - RINGBUFF\_T, [633](#)
- cr\_startup\_lpc5411x-m0.c
  - \_\_attribute\_\_, [806](#)
  - \_\_bss\_section\_table, [806](#)
  - \_\_bss\_section\_table\_end, [806](#)
  - \_\_data\_section\_table, [806](#)
  - \_\_data\_section\_table\_end, [806](#)
  - ALIAS, [805](#)
  - HardFault\_Handler, [806](#)
  - IntDefaultHandler, [806](#)
  - NMI\_Handler, [806](#)
  - PendSV\_Handler, [806](#)
  - ResetISR, [806](#)
  - SVC\_Handler, [806](#)
  - SysTick\_Handler, [806](#)
  - WDT\_BOD\_IRQHandler, [806](#)
  - WEAK, [805](#)
- cr\_startup\_lpc5411x.c
  - \_\_SWVtrace\_Enabled, [808](#)
  - \_\_attribute\_\_, [807](#)
  - \_\_bss\_section\_table, [808](#)
  - \_\_bss\_section\_table\_end, [808](#)
  - \_\_data\_section\_table, [808](#)
  - \_\_data\_section\_table\_end, [808](#)
  - ALIAS, [807](#)
  - BusFault\_Handler, [807](#)
  - DebugMon\_Handler, [807](#)
  - HardFault\_Handler, [807](#)
  - IntDefaultHandler, [807](#)
  - MemManage\_Handler, [807](#)
  - NMI\_Handler, [807](#)
  - PendSV\_Handler, [807](#)
  - ResetISR, [808](#)
  - SVC\_Handler, [808](#)
  - SysTick\_Handler, [808](#)
  - UsageFault\_Handler, [808](#)
  - WDT\_BOD\_IRQHandler, [808](#)
  - WEAK, [807](#)
- crc32\_len
  - PINTABLE\_T, [628](#)
- crc32\_val
  - PINTABLE\_T, [628](#)
- curPllRate
  - pll\_5411x.c, [796](#)
- DAT
  - LPC\_ADC\_T, [571](#)
- DAYSPEERLEAPYEAR
  - rtc\_ut.c, [797](#)
- DAYSPEERWEEK
  - rtc\_ut.c, [797](#)
- DAYSPEERYEAR
  - rtc\_ut.c, [797](#)
- DC\_CTRL
  - CHIP: LPC5411X DMIC driver, [143](#)

- DC\_REMOVAL\_T
  - CHIP: LPC5411X DMIC driver, [137](#)
- DCRDR
  - CoreDebug\_Type, [553](#)
- DCRSR
  - CoreDebug\_Type, [553](#)
- DEMCR
  - CoreDebug\_Type, [553](#)
- DEVICE\_ID
  - LPC\_SYSCON\_T, [610](#)
- DEVID
  - TPI\_Type, [645](#)
- DEVTYPE
  - TPI\_Type, [645](#)
- DFR
  - SCB\_Type, [635](#)
- DFSR
  - SCB\_Type, [635](#)
- DHCSR
  - CoreDebug\_Type, [553](#)
- DIR
  - LPC\_GPIO\_T, [582](#)
- DISABLE
  - LPC Public Types, [480](#)
- DIV
  - LPC\_I2S\_T, [584](#)
  - LPC\_SPI\_T, [604](#)
- DIVHFCLK
  - CHIP: LPC5411X DMIC driver, [143](#)
- DLY
  - LPC\_SPI\_T, [605](#)
- DMA0REQUEST
  - LPC\_SCT\_T, [600](#)
- DMA1REQUEST
  - LPC\_SCT\_T, [600](#)
- DMA\_ADDR
  - CHIP: LPC5411X DMA Engine driver (legacy), [122](#)
- DMA\_CALLBACK\_T
  - CHIP: LPC5411X DMA Service driver, [129](#)
- DMA\_CFG\_BURSTPOWER
  - CHIP: LPC5411X DMA Engine driver (legacy), [122](#)
- DMA\_CFG\_BURSTPOWER\_1
  - CHIP: LPC5411X DMA Engine driver (legacy), [122](#)
- DMA\_CFG\_BURSTPOWER\_1024
  - CHIP: LPC5411X DMA Engine driver (legacy), [122](#)
- DMA\_CFG\_BURSTPOWER\_128
  - CHIP: LPC5411X DMA Engine driver (legacy), [123](#)
- DMA\_CFG\_BURSTPOWER\_16
  - CHIP: LPC5411X DMA Engine driver (legacy), [123](#)
- DMA\_CFG\_BURSTPOWER\_2
  - CHIP: LPC5411X DMA Engine driver (legacy), [123](#)
- DMA\_CFG\_BURSTPOWER\_256
  - CHIP: LPC5411X DMA Engine driver (legacy), [123](#)
- DMA\_CFG\_BURSTPOWER\_32
  - CHIP: LPC5411X DMA Engine driver (legacy), [123](#)
- DMA\_CFG\_BURSTPOWER\_4
  - CHIP: LPC5411X DMA Engine driver (legacy), [123](#)
- DMA\_CFG\_BURSTPOWER\_512
  - CHIP: LPC5411X DMA Engine driver (legacy), [123](#)
- DMA\_CFG\_BURSTPOWER\_64
  - CHIP: LPC5411X DMA Engine driver (legacy), [123](#)
- DMA\_CFG\_BURSTPOWER\_8
  - CHIP: LPC5411X DMA Engine driver (legacy), [123](#)
- DMA\_CFG\_CHPRIORITY
  - CHIP: LPC5411X DMA Engine driver (legacy), [124](#)
- DMA\_CFG\_DSTBURSTWRAP
  - CHIP: LPC5411X DMA Engine driver (legacy), [124](#)
- DMA\_CFG\_HWTRIGEN
  - CHIP: LPC5411X DMA Engine driver (legacy), [124](#)
- DMA\_CFG\_PERIPHREQEN
  - CHIP: LPC5411X DMA Engine driver (legacy), [124](#)
- DMA\_CFG\_SRCBURSTWRAP
  - CHIP: LPC5411X DMA Engine driver (legacy), [124](#)
- DMA\_CFG\_TRIGBURST\_BURST
  - CHIP: LPC5411X DMA Engine driver (legacy), [124](#)
- DMA\_CFG\_TRIGBURST\_SNGL
  - CHIP: LPC5411X DMA Engine driver (legacy), [124](#)
- DMA\_CFG\_TRIGPOL\_HIGH
  - CHIP: LPC5411X DMA Engine driver (legacy), [124](#)
- DMA\_CFG\_TRIGPOL\_LOW
  - CHIP: LPC5411X DMA Engine driver (legacy), [124](#)
- DMA\_CFG\_TRIGTYPE\_EDGE
  - CHIP: LPC5411X DMA Engine driver (legacy), [125](#)
- DMA\_CFG\_TRIGTYPE\_LEVEL
  - CHIP: LPC5411X DMA Engine driver (legacy), [125](#)
- DMA\_CH0
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_CH1
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_CH10
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH11
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH12
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH13
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH14
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH15
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH16
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH17
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH18
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH19
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH2
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_CH3
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_CH4
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH5

- CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH6
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH7
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH8
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CH9
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMA\_CHDESC\_T, [554](#)
  - dest, [554](#)
  - next, [554](#)
  - source, [554](#)
  - xfercfg, [554](#)
- DMA\_CHID\_T
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_CTLSTAT\_TRIG
  - CHIP: LPC5411X DMA Engine driver (legacy), [125](#)
- DMA\_CTLSTAT\_VALIDPENDING
  - CHIP: LPC5411X DMA Engine driver (legacy), [125](#)
- DMA\_DUAL\_DESCRIPTOR\_T, [555](#)
- DMA\_INTSTAT\_ACTIVEERRINT
  - CHIP: LPC5411X DMA Engine driver (legacy), [125](#)
- DMA\_INTSTAT\_ACTIVEINT
  - CHIP: LPC5411X DMA Engine driver (legacy), [125](#)
- DMA\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- DMA\_ITRIG\_INMUX
  - LPC\_INMUX\_T, [587](#)
- DMA\_OTRIG\_INMUX
  - LPC\_INMUX\_T, [587](#)
- DMA\_PERIPHERAL\_CONTEXT\_T, [555](#)
- DMA\_TRIGSRC\_T
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMA\_XFERCFG\_CFGVALID
  - CHIP: LPC5411X DMA Engine driver (legacy), [125](#)
- DMA\_XFERCFG\_CLRTRIG
  - CHIP: LPC5411X DMA Engine driver (legacy), [125](#)
- DMA\_XFERCFG\_DSTINC
  - dma\_service\_5411x.c, [778](#)
- DMA\_XFERCFG\_DSTINC\_0
  - CHIP: LPC5411X DMA Engine driver (legacy), [125](#)
- DMA\_XFERCFG\_DSTINC\_1
  - CHIP: LPC5411X DMA Engine driver (legacy), [126](#)
- DMA\_XFERCFG\_DSTINC\_2
  - CHIP: LPC5411X DMA Engine driver (legacy), [126](#)
- DMA\_XFERCFG\_DSTINC\_4
  - CHIP: LPC5411X DMA Engine driver (legacy), [126](#)
- DMA\_XFERCFG\_RELOAD
  - CHIP: LPC5411X DMA Engine driver (legacy), [126](#)
- DMA\_XFERCFG\_SETINTA
  - CHIP: LPC5411X DMA Engine driver (legacy), [126](#)
- DMA\_XFERCFG\_SETINTB
  - CHIP: LPC5411X DMA Engine driver (legacy), [126](#)
- DMA\_XFERCFG\_SRCINC
  - dma\_service\_5411x.c, [778](#)
- DMA\_XFERCFG\_SRCINC\_0
  - CHIP: LPC5411X DMA Engine driver (legacy), [126](#)
- DMA\_XFERCFG\_SRCINC\_1
  - CHIP: LPC5411X DMA Engine driver (legacy), [126](#)
- DMA\_XFERCFG\_SRCINC\_2
  - CHIP: LPC5411X DMA Engine driver (legacy), [126](#)
- DMA\_XFERCFG\_SRCINC\_4
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_XFERCFG\_SWTRIG
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_XFERCFG\_WIDTH
  - dma\_service\_5411x.c, [778](#)
- DMA\_XFERCFG\_WIDTH\_16
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_XFERCFG\_WIDTH\_32
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_XFERCFG\_WIDTH\_8
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMA\_XFERCFG\_XFERCOUNT
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- DMACH
  - LPC\_DMA\_T, [579](#)
- DMACOMMON
  - LPC\_DMA\_T, [579](#)
- DMAREQ\_DMIC0
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_DMIC1
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM0\_RX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM0\_TX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM1\_RX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM1\_TX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM2\_RX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM2\_TX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM3\_RX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM3\_TX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM4\_RX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM4\_TX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM5\_RX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM5\_TX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM6\_RX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM6\_TX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)



- DMAREQ\_FLEXCOMM7\_RX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_FLEXCOMM7\_TX
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMAREQ\_I2S6\_RX
  - i2s\_5411x.h, [723](#)
- DMAREQ\_I2S6\_TX
  - i2s\_5411x.h, [723](#)
- DMAREQ\_I2S7\_RX
  - i2s\_5411x.h, [723](#)
- DMAREQ\_I2S7\_TX
  - i2s\_5411x.h, [723](#)
- DMAREQ\_SPIFI
  - CHIP: LPC5411X DMA Engine driver (legacy), [128](#)
- DMATRIG\_ADC0\_SEQA\_IRQ
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMATRIG\_ADC0\_SEQB\_IRQ
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMATRIG\_OUTMUX0
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_OUTMUX1
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_OUTMUX2
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_OUTMUX3
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_PININT0
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_PININT1
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_PININT2
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_PININT3
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_SCT0\_DMA0
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMATRIG\_SCT0\_DMA1
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMATRIG\_TIMER0\_MATCH0
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMATRIG\_TIMER0\_MATCH1
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMATRIG\_TIMER1\_MATCH0
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMATRIG\_TIMER2\_MATCH0
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMATRIG\_TIMER2\_MATCH1
  - CHIP: LPC5411X Input Mux Registers and Driver, [194](#)
- DMATRIG\_TIMER3\_MATCH0
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_TIMER4\_MATCH0
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMATRIG\_TIMER4\_MATCH1
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- DMIC\_CHANNEL\_CONFIG\_T, [555](#)
- DMIC\_COMP0\_0
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_COMP0\_13
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_COMP0\_15
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_COMP0\_16
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_DC\_CUT155
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_DC\_CUT39
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_DC\_CUT78
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_DC\_NOREMOVE
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_DCGAIN\_REDUCE\_P
  - CHIP: LPC5411X DMIC driver, [135](#)
- DMIC\_DCPOLE\_P
  - CHIP: LPC5411X DMIC driver, [135](#)
- DMIC\_FIFO\_DMAEN
  - CHIP: LPC5411X DMIC driver, [135](#)
- DMIC\_FIFO\_DMAEN\_P
  - CHIP: LPC5411X DMIC driver, [135](#)
- DMIC\_FIFO\_ENABLE
  - CHIP: LPC5411X DMIC driver, [135](#)
- DMIC\_FIFO\_ENABLE\_P
  - CHIP: LPC5411X DMIC driver, [135](#)
- DMIC\_FIFO\_INT
  - CHIP: LPC5411X DMIC driver, [135](#)
- DMIC\_FIFO\_INT\_P
  - CHIP: LPC5411X DMIC driver, [135](#)
- DMIC\_FIFO\_INTREN
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_FIFO\_INTREN\_P
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_FIFO\_OVERRUN
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_FIFO\_OVERRUN\_P
  - CHIP: LPC5411X DMIC driver, [136](#)

- DMIC\_FIFO\_RESETN
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_FIFO\_RESETN\_P
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_FIFO\_TLVL\_P
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_FIFO\_UNDERRUN
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_FIFO\_UNDERRUN\_P
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_IO\_T
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- DMIC\_LEFT
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_OP\_DMA
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_OP\_INTR
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_OP\_POLL
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV1
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV12
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV128
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV16
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV2
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV24
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV3
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV32
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV4
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV48
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV6
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV64
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV8
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PDM\_DIV96
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_PHY\_FALL
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_PHY\_FALL\_P
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_PHY\_HALF
  - CHIP: LPC5411X DMIC driver, [136](#)
- DMIC\_PHY\_HALF\_P
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_RIGHT
  - CHIP: LPC5411X DMIC driver, [138](#)
- DMIC\_SATURATE\_AT16BIT\_P
  - CHIP: LPC5411X DMIC driver, [137](#)
- DMIC\_STATISTICS\_T, [556](#)
- DMICCLKDIV
  - LPC\_SYSCON\_T, [610](#)
- DMICCLKSEL
  - LPC\_SYSCON\_T, [611](#)
- DSP\_WS\_50
  - i2s\_5411x.h, [730](#)
- DSP\_WS\_LONG
  - i2s\_5411x.h, [730](#)
- DSP\_WS\_SHORT
  - i2s\_5411x.h, [730](#)
- DWT
  - Core Definitions, [451](#)
- DWT\_BASE
  - Core Definitions, [451](#)
- DWT\_CPICNT\_CPICNT\_Msk
  - Data Watchpoint and Trace (DWT), [456](#)
- DWT\_CPICNT\_CPICNT\_Pos
  - Data Watchpoint and Trace (DWT), [456](#)
- DWT\_CTRL\_CPIEVTENA\_Msk
  - Data Watchpoint and Trace (DWT), [456](#)
- DWT\_CTRL\_CPIEVTENA\_Pos
  - Data Watchpoint and Trace (DWT), [456](#)
- DWT\_CTRL\_CYCCNTENA\_Msk
  - Data Watchpoint and Trace (DWT), [456](#)
- DWT\_CTRL\_CYCCNTENA\_Pos
  - Data Watchpoint and Trace (DWT), [457](#)
- DWT\_CTRL\_CYCEVTENA\_Msk
  - Data Watchpoint and Trace (DWT), [457](#)
- DWT\_CTRL\_CYCEVTENA\_Pos
  - Data Watchpoint and Trace (DWT), [457](#)
- DWT\_CTRL\_CYCTAP\_Msk
  - Data Watchpoint and Trace (DWT), [457](#)
- DWT\_CTRL\_CYCTAP\_Pos
  - Data Watchpoint and Trace (DWT), [457](#)
- DWT\_CTRL\_EXCEVTENA\_Msk
  - Data Watchpoint and Trace (DWT), [457](#)
- DWT\_CTRL\_EXCEVTENA\_Pos
  - Data Watchpoint and Trace (DWT), [457](#)
- DWT\_CTRL\_EXCTRCENA\_Msk
  - Data Watchpoint and Trace (DWT), [457](#)
- DWT\_CTRL\_EXCTRCENA\_Pos
  - Data Watchpoint and Trace (DWT), [457](#)
- DWT\_CTRL\_FOLDEVTENA\_Msk
  - Data Watchpoint and Trace (DWT), [458](#)
- DWT\_CTRL\_FOLDEVTENA\_Pos
  - Data Watchpoint and Trace (DWT), [458](#)
- DWT\_CTRL\_LSUEVTENA\_Msk
  - Data Watchpoint and Trace (DWT), [458](#)
- DWT\_CTRL\_LSUEVTENA\_Pos
  - Data Watchpoint and Trace (DWT), [458](#)

- DWT\_CTRL\_NOCYCCNT\_Msk
  - Data Watchpoint and Trace (DWT), [458](#)
- DWT\_CTRL\_NOCYCCNT\_Pos
  - Data Watchpoint and Trace (DWT), [458](#)
- DWT\_CTRL\_NOEXTTRIG\_Msk
  - Data Watchpoint and Trace (DWT), [458](#)
- DWT\_CTRL\_NOEXTTRIG\_Pos
  - Data Watchpoint and Trace (DWT), [458](#)
- DWT\_CTRL\_NOPRFCNT\_Msk
  - Data Watchpoint and Trace (DWT), [458](#)
- DWT\_CTRL\_NOPRFCNT\_Pos
  - Data Watchpoint and Trace (DWT), [459](#)
- DWT\_CTRL\_NOTRCPKT\_Msk
  - Data Watchpoint and Trace (DWT), [459](#)
- DWT\_CTRL\_NOTRCPKT\_Pos
  - Data Watchpoint and Trace (DWT), [459](#)
- DWT\_CTRL\_NUMCOMP\_Msk
  - Data Watchpoint and Trace (DWT), [459](#)
- DWT\_CTRL\_NUMCOMP\_Pos
  - Data Watchpoint and Trace (DWT), [459](#)
- DWT\_CTRL\_PCSAMPLENA\_Msk
  - Data Watchpoint and Trace (DWT), [459](#)
- DWT\_CTRL\_PCSAMPLENA\_Pos
  - Data Watchpoint and Trace (DWT), [459](#)
- DWT\_CTRL\_POSTINIT\_Msk
  - Data Watchpoint and Trace (DWT), [459](#)
- DWT\_CTRL\_POSTINIT\_Pos
  - Data Watchpoint and Trace (DWT), [459](#)
- DWT\_CTRL\_POSTPRESET\_Msk
  - Data Watchpoint and Trace (DWT), [460](#)
- DWT\_CTRL\_POSTPRESET\_Pos
  - Data Watchpoint and Trace (DWT), [460](#)
- DWT\_CTRL\_SLEEPEVTENA\_Msk
  - Data Watchpoint and Trace (DWT), [460](#)
- DWT\_CTRL\_SLEEPEVTENA\_Pos
  - Data Watchpoint and Trace (DWT), [460](#)
- DWT\_CTRL\_SYNCTAP\_Msk
  - Data Watchpoint and Trace (DWT), [460](#)
- DWT\_CTRL\_SYNCTAP\_Pos
  - Data Watchpoint and Trace (DWT), [460](#)
- DWT\_EXCCNT\_EXCCNT\_Msk
  - Data Watchpoint and Trace (DWT), [460](#)
- DWT\_EXCCNT\_EXCCNT\_Pos
  - Data Watchpoint and Trace (DWT), [460](#)
- DWT\_FOLDCNT\_FOLDCNT\_Msk
  - Data Watchpoint and Trace (DWT), [460](#)
- DWT\_FOLDCNT\_FOLDCNT\_Pos
  - Data Watchpoint and Trace (DWT), [461](#)
- DWT\_FUNCTION\_CYCMATCH\_Msk
  - Data Watchpoint and Trace (DWT), [461](#)
- DWT\_FUNCTION\_CYCMATCH\_Pos
  - Data Watchpoint and Trace (DWT), [461](#)
- DWT\_FUNCTION\_DATAVADDR0\_Msk
  - Data Watchpoint and Trace (DWT), [461](#)
- DWT\_FUNCTION\_DATAVADDR0\_Pos
  - Data Watchpoint and Trace (DWT), [461](#)
- DWT\_FUNCTION\_DATAVADDR1\_Msk
  - Data Watchpoint and Trace (DWT), [461](#)
- DWT\_FUNCTION\_DATAVADDR1\_Pos
  - Data Watchpoint and Trace (DWT), [461](#)
- DWT\_FUNCTION\_DATAVMATCH\_Msk
  - Data Watchpoint and Trace (DWT), [461](#)
- DWT\_FUNCTION\_DATAVMATCH\_Pos
  - Data Watchpoint and Trace (DWT), [461](#)
- DWT\_FUNCTION\_DATAVSIZE\_Msk
  - Data Watchpoint and Trace (DWT), [462](#)
- DWT\_FUNCTION\_DATAVSIZE\_Pos
  - Data Watchpoint and Trace (DWT), [462](#)
- DWT\_FUNCTION\_EMITRANGE\_Msk
  - Data Watchpoint and Trace (DWT), [462](#)
- DWT\_FUNCTION\_EMITRANGE\_Pos
  - Data Watchpoint and Trace (DWT), [462](#)
- DWT\_FUNCTION\_FUNCTION\_Msk
  - Data Watchpoint and Trace (DWT), [462](#)
- DWT\_FUNCTION\_FUNCTION\_Pos
  - Data Watchpoint and Trace (DWT), [462](#)
- DWT\_FUNCTION\_LNK1ENA\_Msk
  - Data Watchpoint and Trace (DWT), [462](#)
- DWT\_FUNCTION\_LNK1ENA\_Pos
  - Data Watchpoint and Trace (DWT), [462](#)
- DWT\_FUNCTION\_MATCHED\_Msk
  - Data Watchpoint and Trace (DWT), [462](#)
- DWT\_FUNCTION\_MATCHED\_Pos
  - Data Watchpoint and Trace (DWT), [463](#)
- DWT\_LSUCNT\_LSUCNT\_Msk
  - Data Watchpoint and Trace (DWT), [463](#)
- DWT\_LSUCNT\_LSUCNT\_Pos
  - Data Watchpoint and Trace (DWT), [463](#)
- DWT\_MASK\_MASK\_Msk
  - Data Watchpoint and Trace (DWT), [463](#)
- DWT\_MASK\_MASK\_Pos
  - Data Watchpoint and Trace (DWT), [463](#)
- DWT\_SLEEP\_CNT\_SLEEP\_CNT\_Msk
  - Data Watchpoint and Trace (DWT), [463](#)
- DWT\_SLEEP\_CNT\_SLEEP\_CNT\_Pos
  - Data Watchpoint and Trace (DWT), [463](#)
- DWT\_Type, [556](#)
  - COMP0, [557](#)
  - COMP1, [557](#)
  - COMP2, [557](#)
  - COMP3, [557](#)
  - CPICNT, [557](#)
  - CTRL, [557](#)
  - CYCCNT, [557](#)
  - EXCCNT, [557](#)
  - FOLDCNT, [558](#)
  - FUNCTION0, [558](#)
  - FUNCTION1, [558](#)
  - FUNCTION2, [558](#)
  - FUNCTION3, [558](#)
  - LSUCNT, [558](#)
  - MASK0, [558](#)
  - MASK1, [558](#)
  - MASK2, [558](#)
  - MASK3, [559](#)
  - PCSR, [559](#)



- RESERVED0, [559](#)
- RESERVED1, [559](#)
- RESERVED2, [559](#)
- SLEEPcnt, [559](#)
- data
  - RINGBUFF\_T, [633](#)
- Data Watchpoint and Trace (DWT), [455](#)
  - DWT\_CPICNT\_CPICNT\_Msk, [456](#)
  - DWT\_CPICNT\_CPICNT\_Pos, [456](#)
  - DWT\_CTRL\_CPIEVTENA\_Msk, [456](#)
  - DWT\_CTRL\_CPIEVTENA\_Pos, [456](#)
  - DWT\_CTRL\_CYCCNTENA\_Msk, [456](#)
  - DWT\_CTRL\_CYCCNTENA\_Pos, [457](#)
  - DWT\_CTRL\_CYCEVTENA\_Msk, [457](#)
  - DWT\_CTRL\_CYCEVTENA\_Pos, [457](#)
  - DWT\_CTRL\_CYCTAP\_Msk, [457](#)
  - DWT\_CTRL\_CYCTAP\_Pos, [457](#)
  - DWT\_CTRL\_EXCEVTENA\_Msk, [457](#)
  - DWT\_CTRL\_EXCEVTENA\_Pos, [457](#)
  - DWT\_CTRL\_EXCTRCENA\_Msk, [457](#)
  - DWT\_CTRL\_EXCTRCENA\_Pos, [457](#)
  - DWT\_CTRL\_FOLDEVTENA\_Msk, [458](#)
  - DWT\_CTRL\_FOLDEVTENA\_Pos, [458](#)
  - DWT\_CTRL\_LSUEVTENA\_Msk, [458](#)
  - DWT\_CTRL\_LSUEVTENA\_Pos, [458](#)
  - DWT\_CTRL\_NOCYCCNT\_Msk, [458](#)
  - DWT\_CTRL\_NOCYCCNT\_Pos, [458](#)
  - DWT\_CTRL\_NOEXTTRIG\_Msk, [458](#)
  - DWT\_CTRL\_NOEXTTRIG\_Pos, [458](#)
  - DWT\_CTRL\_NOPRFCNT\_Msk, [458](#)
  - DWT\_CTRL\_NOPRFCNT\_Pos, [459](#)
  - DWT\_CTRL\_NOTRCPKT\_Msk, [459](#)
  - DWT\_CTRL\_NOTRCPKT\_Pos, [459](#)
  - DWT\_CTRL\_NUMCOMP\_Msk, [459](#)
  - DWT\_CTRL\_NUMCOMP\_Pos, [459](#)
  - DWT\_CTRL\_PCSAMPLENA\_Msk, [459](#)
  - DWT\_CTRL\_PCSAMPLENA\_Pos, [459](#)
  - DWT\_CTRL\_POSTINIT\_Msk, [459](#)
  - DWT\_CTRL\_POSTINIT\_Pos, [459](#)
  - DWT\_CTRL\_POSTPRESET\_Msk, [460](#)
  - DWT\_CTRL\_POSTPRESET\_Pos, [460](#)
  - DWT\_CTRL\_SLEEPEVTENA\_Msk, [460](#)
  - DWT\_CTRL\_SLEEPEVTENA\_Pos, [460](#)
  - DWT\_CTRL\_SYNCTAP\_Msk, [460](#)
  - DWT\_CTRL\_SYNCTAP\_Pos, [460](#)
  - DWT\_EXCCNT\_EXCCNT\_Msk, [460](#)
  - DWT\_EXCCNT\_EXCCNT\_Pos, [460](#)
  - DWT\_FOLDcnt\_FOLDcnt\_Msk, [460](#)
  - DWT\_FOLDcnt\_FOLDcnt\_Pos, [461](#)
  - DWT\_FUNCTION\_CYCMATCH\_Msk, [461](#)
  - DWT\_FUNCTION\_CYCMATCH\_Pos, [461](#)
  - DWT\_FUNCTION\_DATAVADDR0\_Msk, [461](#)
  - DWT\_FUNCTION\_DATAVADDR0\_Pos, [461](#)
  - DWT\_FUNCTION\_DATAVADDR1\_Msk, [461](#)
  - DWT\_FUNCTION\_DATAVADDR1\_Pos, [461](#)
  - DWT\_FUNCTION\_DATAVMATCH\_Msk, [461](#)
  - DWT\_FUNCTION\_DATAVMATCH\_Pos, [461](#)
  - DWT\_FUNCTION\_DATAVSIZE\_Msk, [462](#)
  - DWT\_FUNCTION\_DATAVSIZE\_Pos, [462](#)
  - DWT\_FUNCTION\_EMITRANGE\_Msk, [462](#)
  - DWT\_FUNCTION\_EMITRANGE\_Pos, [462](#)
  - DWT\_FUNCTION\_FUNCTION\_Msk, [462](#)
  - DWT\_FUNCTION\_FUNCTION\_Pos, [462](#)
  - DWT\_FUNCTION\_LNK1ENA\_Msk, [462](#)
  - DWT\_FUNCTION\_LNK1ENA\_Pos, [462](#)
  - DWT\_FUNCTION\_MATCHED\_Msk, [462](#)
  - DWT\_FUNCTION\_MATCHED\_Pos, [463](#)
  - DWT\_LSUCNT\_LSUCNT\_Msk, [463](#)
  - DWT\_LSUCNT\_LSUCNT\_Pos, [463](#)
  - DWT\_MASK\_MASK\_Msk, [463](#)
  - DWT\_MASK\_MASK\_Pos, [463](#)
  - DWT\_SLEEPcnt\_SLEEPcnt\_Msk, [463](#)
  - DWT\_SLEEPcnt\_SLEEPcnt\_Pos, [463](#)
- DataPos
  - I2S\_AUDIO\_FORMAT\_T, [562](#)
- dataWidth
  - SPIM\_XFER\_T, [640](#)
  - SPIS\_XFER\_T, [642](#)
- daysPerMonth
  - rtc\_ut.c, [798](#)
- DebugMon\_Handler
  - cr\_startup\_lpc5411x.c, [807](#)
- DebugMonitor\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- Defines and Type Definitions, [464](#)
- descr
  - CHIP: LPC5411X DMA Service driver, [132](#)
- desiredRate
  - PLL\_CONFIG\_T, [630](#)
- dest
  - DMA\_CHDESC\_T, [554](#)
- Dir
  - \_BM\_T, [551](#)
- Direction
  - I2S\_AUDIO\_FORMAT\_T, [562](#)
- div
  - UART\_BAUD\_T, [648](#)
- divhclk
  - CHIP: LPC5411X DMIC driver, [144](#)
- Divider
  - I2S\_AUDIO\_FORMAT\_T, [562](#)
- dma\_pCallback\_array
  - dma\_service\_5411x.c, [779](#)
- dma\_service\_5411x.c
  - Chip\_DMASERVICE\_ErrorHandler, [779](#)
  - DMA\_XFERCFG\_DSTINC, [778](#)
  - DMA\_XFERCFG\_SRCINC, [778](#)
  - DMA\_XFERCFG\_WIDTH, [778](#)
  - dma\_pCallback\_array, [779](#)
  - dma\_service\_error\_cb, [779](#)
- dma\_service\_error\_cb
  - dma\_service\_5411x.c, [779](#)
- dmic\_ch0\_dma\_context
  - CHIP: LPC5411X DMIC driver, [144](#)
- dmic\_ch0\_dma\_interleaved\_context

- CHIP: LPC5411X DMIC driver, [144](#)
- dmic\_ch1\_dma\_context
  - CHIP: LPC5411X DMIC driver, [144](#)
- dmic\_ch1\_dma\_interleaved\_context
  - CHIP: LPC5411X DMIC driver, [144](#)
- dst\_increment
  - CHIP: LPC5411X DMA Service driver, [132](#)
- ECHO\_DIS
  - CHIP: LPC5411X UART Driver, [344](#)
- ECHO\_EN
  - CHIP: LPC5411X UART Driver, [344](#)
- EMR
  - LPC\_TIMER\_T, [619](#)
- ENABLE
  - LPC Public Types, [480](#)
- ENABLECLR
  - LPC\_DMA\_COMMON\_T, [577](#)
- ENABLESET
  - LPC\_DMA\_COMMON\_T, [577](#)
- ERR\_ADC\_BASE
  - error.h, [709](#)
- ERR\_ADC\_INVALID\_CHANNEL
  - error.h, [709](#)
- ERR\_ADC\_INVALID\_LENGTH
  - error.h, [709](#)
- ERR\_ADC\_INVALID\_SEQUENCE
  - error.h, [709](#)
- ERR\_ADC\_INVALID\_SETUP
  - error.h, [709](#)
- ERR\_ADC\_NO\_POWER
  - error.h, [709](#)
- ERR\_ADC\_OVERRUN
  - error.h, [709](#)
- ERR\_ADC\_PARAM
  - error.h, [709](#)
- ERR\_API\_BASE
  - error.h, [706](#)
- ERR\_API\_INVALID\_PARAM1
  - error.h, [706](#)
- ERR\_API\_INVALID\_PARAM2
  - error.h, [706](#)
- ERR\_API\_INVALID\_PARAM3
  - error.h, [706](#)
- ERR\_API\_INVALID\_PARAMS
  - error.h, [706](#)
- ERR\_API\_MOD\_INIT
  - error.h, [706](#)
- ERR\_BUSY
  - error.h, [706](#)
- ERR\_CAN\_BAD\_MEM\_BUF
  - error.h, [708](#)
- ERR\_CAN\_BASE
  - error.h, [708](#)
- ERR\_CAN\_INIT\_FAIL
  - error.h, [708](#)
- ERR\_CANOPEN\_INIT\_FAIL
  - error.h, [708](#)
- ERR\_CGU\_BASE
  - error.h, [707](#)
- ERR\_CGU\_DIV\_SRC
  - error.h, [707](#)
- ERR\_CGU\_DIV\_VAL
  - error.h, [707](#)
- ERR\_CGU\_INVALID\_PARAM
  - error.h, [707](#)
- ERR\_CGU\_INVALID\_SLICE
  - error.h, [707](#)
- ERR\_CGU\_NOT\_IMPL
  - error.h, [707](#)
- ERR\_CGU\_OUTPUT\_GEN
  - error.h, [707](#)
- ERR\_CGU\_SRC
  - error.h, [707](#)
- ERR\_CLK\_BASE
  - error.h, [708](#)
- ERR\_CLK\_BASE\_OFF
  - error.h, [709](#)
- ERR\_CLK\_CFG
  - error.h, [709](#)
- ERR\_CLK\_DIV\_SRC
  - error.h, [708](#)
- ERR\_CLK\_DIV\_VAL
  - error.h, [708](#)
- ERR\_CLK\_INVALID\_PARAM
  - error.h, [708](#)
- ERR\_CLK\_INVALID\_SLICE
  - error.h, [708](#)
- ERR\_CLK\_NOT\_IMPL
  - error.h, [708](#)
- ERR\_CLK\_OFF\_DEADLOCK
  - error.h, [709](#)
- ERR\_CLK\_OSC\_FREQ
  - error.h, [709](#)
- ERR\_CLK\_OUTPUT\_GEN
  - error.h, [708](#)
- ERR\_CLK\_PLL\_FIN\_TOO\_LARGE
  - error.h, [708](#)
- ERR\_CLK\_PLL\_FIN\_TOO\_SMALL
  - error.h, [708](#)
- ERR\_CLK\_PLL\_FOUT\_TOO\_LARGE
  - error.h, [708](#)
- ERR\_CLK\_PLL\_FOUT\_TOO\_SMALL
  - error.h, [708](#)
- ERR\_CLK\_PLL\_MAX\_PCT
  - error.h, [709](#)
- ERR\_CLK\_PLL\_MIN\_PCT
  - error.h, [709](#)
- ERR\_CLK\_PLL\_NO\_SOLUTION
  - error.h, [709](#)
- ERR\_CLK\_SRC
  - error.h, [708](#)
- ERR\_CLK\_TIMEOUT
  - error.h, [709](#)
- ERR\_DM\_BASE
  - error.h, [709](#)
- ERR\_DM\_COMM\_FAIL

- error.h, [709](#)
- ERR\_DM\_NOT\_ENTERED
  - error.h, [709](#)
- ERR\_DM\_UNKNOWN\_CMD
  - error.h, [709](#)
- ERR\_DMA\_BASE
  - error.h, [709](#)
- ERR\_DMA\_BUSY
  - error.h, [709](#)
- ERR\_DMA\_CHANNEL\_DISABLED
  - error.h, [709](#)
- ERR\_DMA\_CHANNEL\_NUMBER
  - error.h, [709](#)
- ERR\_DMA\_CHANNEL\_VALID\_PENDING
  - error.h, [709](#)
- ERR\_DMA\_ERROR\_INT
  - error.h, [709](#)
- ERR\_DMA\_GENERAL
  - error.h, [709](#)
- ERR\_DMA\_NOT\_ALIGNMENT
  - error.h, [709](#)
- ERR\_DMA\_PARAM
  - error.h, [709](#)
- ERR\_DMA\_PING\_PONG\_EN
  - error.h, [709](#)
- ERR\_DMA\_QUEUE\_EMPTY
  - error.h, [709](#)
- ERR\_FAILED
  - error.h, [706](#)
- ERR\_FLEXCOMM\_FUNCNOTSUPPORTED
  - CHIP: LPC5411X flexcomm API, [379](#)
- ERR\_FLEXCOMM\_INVALIDBASE
  - CHIP: LPC5411X flexcomm API, [379](#)
- ERR\_FLEXCOMM\_NOTFREE
  - CHIP: LPC5411X flexcomm API, [379](#)
- ERR\_I2C\_BASE
  - error.h, [707](#)
- ERR\_I2C\_BUFFER\_OVERFLOW
  - error.h, [707](#)
- ERR\_I2C\_BUFFER\_UNDERFLOW
  - error.h, [708](#)
- ERR\_I2C\_BUSY
  - error.h, [707](#)
- ERR\_I2C\_BYTE\_COUNT\_ERR
  - error.h, [707](#)
- ERR\_I2C\_GENERAL\_FAILURE
  - error.h, [707](#)
- ERR\_I2C\_LOSS\_OF\_ARBITRATION
  - error.h, [707](#)
- ERR\_I2C\_LOSS\_OF\_ARBITRATION\_NAK\_BIT
  - error.h, [707](#)
- ERR\_I2C\_NAK
  - error.h, [707](#)
- ERR\_I2C\_PARAM
  - error.h, [708](#)
- ERR\_I2C\_REGS\_SET\_TO\_DEFAULT
  - error.h, [708](#)
- ERR\_I2C\_SLAVE\_NOT\_ADDRESSED
  - error.h, [707](#)
- ERR\_I2C\_TIMEOUT
  - error.h, [708](#)
- ERR\_ISP\_ADDR\_ERROR
  - error.h, [706](#)
- ERR\_ISP\_ADDR\_NOT\_MAPPED
  - error.h, [706](#)
- ERR\_ISP\_BASE
  - error.h, [706](#)
- ERR\_ISP\_BUSY
  - error.h, [706](#)
- ERR\_ISP\_CMD\_LOCKED
  - error.h, [706](#)
- ERR\_ISP\_CODE\_READ\_PROTECTION\_ENABLED
  - error.h, [706](#)
- ERR\_ISP\_COMPARE\_ERROR
  - error.h, [706](#)
- ERR\_ISP\_COUNT\_ERROR
  - error.h, [706](#)
- ERR\_ISP\_DST\_ADDR\_ERROR
  - error.h, [706](#)
- ERR\_ISP\_DST\_ADDR\_NOT\_MAPPED
  - error.h, [706](#)
- ERR\_ISP\_EEPROM\_NO\_CLOCK
  - error.h, [706](#)
- ERR\_ISP\_EEPROM\_NO\_POWER
  - error.h, [706](#)
- ERR\_ISP\_FLASH\_NO\_CLOCK
  - error.h, [706](#)
- ERR\_ISP\_FLASH\_NO\_POWER
  - error.h, [706](#)
- ERR\_ISP\_INVALID\_BAUD\_RATE
  - error.h, [706](#)
- ERR\_ISP\_INVALID\_CODE
  - error.h, [706](#)
- ERR\_ISP\_INVALID\_COMMAND
  - error.h, [706](#)
- ERR\_ISP\_INVALID\_FLASH\_UNIT
  - error.h, [706](#)
- ERR\_ISP\_INVALID\_SECTOR
  - error.h, [706](#)
- ERR\_ISP\_INVALID\_STOP\_BIT
  - error.h, [706](#)
- ERR\_ISP\_IRC\_NO\_POWER
  - error.h, [706](#)
- ERR\_ISP\_PARAM\_ERROR
  - error.h, [706](#)
- ERR\_ISP\_REINVOKE\_ISP\_CONFIG
  - error.h, [706](#)
- ERR\_ISP\_SECTOR\_NOT\_BLANK
  - error.h, [706](#)
- ERR\_ISP\_SECTOR\_NOT\_PREPARED\_FOR\_WRITE↔
  - E\_OPERATION
  - error.h, [706](#)
- ERR\_ISP\_SETTING\_ACTIVE\_PARTITION
  - error.h, [706](#)
- ERR\_ISP\_SRC\_ADDR\_ERROR
  - error.h, [706](#)

|                                                                      |                                                                     |
|----------------------------------------------------------------------|---------------------------------------------------------------------|
| ERR_ISP_SRC_ADDR_NOT_MAPPED<br>error.h, <a href="#">706</a>          | ERR_SPIFI_DEVICE_ERROR<br>error.h, <a href="#">706</a>              |
| ERR_ISP_USER_CODE_CHECKSUM<br>error.h, <a href="#">706</a>           | ERR_SPIFI_ERASE_NEEDED<br>error.h, <a href="#">707</a>              |
| ERR_OTP_AES_KEYS_ENABLED<br>error.h, <a href="#">708</a>             | ERR_SPIFI_INTERNAL_ERROR<br>error.h, <a href="#">706</a>            |
| ERR_OTP_ALL_DATA_OR_MASK_ZERO<br>error.h, <a href="#">708</a>        | ERR_SPIFI_LITE_BASE<br>error.h, <a href="#">708</a>                 |
| ERR_OTP_BASE<br>error.h, <a href="#">708</a>                         | ERR_SPIFI_LITE_BUSY<br>error.h, <a href="#">708</a>                 |
| ERR_OTP_ETH_MAC_ENABLED<br>error.h, <a href="#">708</a>              | ERR_SPIFI_LITE_IN_DMA<br>error.h, <a href="#">708</a>               |
| ERR_OTP_ILLEGAL_BANK<br>error.h, <a href="#">708</a>                 | ERR_SPIFI_LITE_INVALID_ARGUMENTS<br>error.h, <a href="#">708</a>    |
| ERR_OTP_READ_DATA_MISMATCH<br>error.h, <a href="#">708</a>           | ERR_SPIFI_LITE_MEMORY_MODE_OFF<br>error.h, <a href="#">708</a>      |
| ERR_OTP_SOME_BITS_ALREADY_PROGRAMMED<br>error.h, <a href="#">708</a> | ERR_SPIFI_LITE_MEMORY_MODE_ON<br>error.h, <a href="#">708</a>       |
| ERR_OTP_USB_ID_ENABLED<br>error.h, <a href="#">708</a>               | ERR_SPIFI_LITE_NOT_IN_DMA<br>error.h, <a href="#">708</a>           |
| ERR_OTP_WR_ENABLE_INVALID<br>error.h, <a href="#">708</a>            | ERR_SPIFI_NO_DEVICE<br>error.h, <a href="#">707</a>                 |
| ERR_OTP_WRITE_ACCESS_LOCKED<br>error.h, <a href="#">708</a>          | ERR_SPIFI_OPERAND_ERROR<br>error.h, <a href="#">707</a>             |
| ERR_PWR_BASE<br>error.h, <a href="#">709</a>                         | ERR_SPIFI_STATUS_PROBLEM<br>error.h, <a href="#">707</a>            |
| ERR_SEC_AES_BASE<br>error.h, <a href="#">707</a>                     | ERR_SPIFI_TIMEOUT<br>error.h, <a href="#">706</a>                   |
| ERR_SEC_AES_DMA_CHANNEL_CFG<br>error.h, <a href="#">707</a>          | ERR_SPIFI_UNKNOWN_EXT<br>error.h, <a href="#">707</a>               |
| ERR_SEC_AES_DMA_MUX_CFG<br>error.h, <a href="#">707</a>              | ERR_SPIFI_UNKNOWN_ID<br>error.h, <a href="#">707</a>                |
| ERR_SEC_AES_KEY_ALREADY_PROGRAMMED<br>error.h, <a href="#">707</a>   | ERR_SPIFI_UNKNOWN_MFG<br>error.h, <a href="#">707</a>               |
| ERR_SEC_AES_NOT_SUPPORTED<br>error.h, <a href="#">707</a>            | ERR_SPIFI_UNKNOWN_TYPE<br>error.h, <a href="#">707</a>              |
| ERR_SEC_AES_WRONG_CMD<br>error.h, <a href="#">707</a>                | ERR_TIME_OUT<br>error.h, <a href="#">706</a>                        |
| ERR_SPI_BASE<br>error.h, <a href="#">709</a>                         | ERR_UART_BASE<br>error.h, <a href="#">708</a>                       |
| ERR_SPI_BUSY<br>error.h, <a href="#">709</a>                         | ERR_UART_BAUDRATE<br>error.h, <a href="#">708</a>                   |
| ERR_SPI_CLKSTALL<br>error.h, <a href="#">709</a>                     | ERR_UART_OVERRUN_FRAME_PARITY_NOISE<br>error.h, <a href="#">708</a> |
| ERR_SPI_INVALID_LENGTH<br>error.h, <a href="#">709</a>               | ERR_UART_PARAM<br>error.h, <a href="#">708</a>                      |
| ERR_SPI_PARAM<br>error.h, <a href="#">709</a>                        | ERR_UART_RXD_BUSY<br>error.h, <a href="#">708</a>                   |
| ERR_SPI_RXOVERRUN<br>error.h, <a href="#">709</a>                    | ERR_UART_TXD_BUSY<br>error.h, <a href="#">708</a>                   |
| ERR_SPI_SELNASSERT<br>error.h, <a href="#">709</a>                   | ERR_UART_UNDERRUN<br>error.h, <a href="#">708</a>                   |
| ERR_SPI_SELNDEASSERT<br>error.h, <a href="#">709</a>                 | ERR_USBD_BAD_CFG_DESC<br>error.h, <a href="#">707</a>               |
| ERR_SPI_TXUNDERRUN<br>error.h, <a href="#">709</a>                   | ERR_USBD_BAD_DESC<br>error.h, <a href="#">707</a>                   |
| ERR_SPIFI_BASE<br>error.h, <a href="#">706</a>                       | ERR_USBD_BAD_EP_DESC<br>error.h, <a href="#">707</a>                |

- ERR\_USBD\_BAD\_INTF\_DESC
  - error.h, [707](#)
- ERR\_USBD\_BAD\_MEM\_BUF
  - error.h, [707](#)
- ERR\_USBD\_BASE
  - error.h, [707](#)
- ERR\_USBD\_INVALID\_REQ
  - error.h, [707](#)
- ERR\_USBD\_SEND\_DATA
  - error.h, [707](#)
- ERR\_USBD\_SEND\_ZLP
  - error.h, [707](#)
- ERR\_USBD\_STALL
  - error.h, [707](#)
- ERR\_USBD\_TOO\_MANY\_CLASS\_HDLR
  - error.h, [707](#)
- ERR\_USBD\_UNHANDLED
  - error.h, [707](#)
- ERRINT
  - LPC\_DMA\_COMMON\_T, [577](#)
- ERROR
  - LPC Public Types, [480](#)
- EVEN
  - LPC\_SCT\_T, [600](#)
- EVENT
  - LPC\_SCT\_T, [600](#)
- EVFLAG
  - LPC\_SCT\_T, [601](#)
- EXCCNT
  - DWT\_Type, [557](#)
- EXT\_SCLK\_MASTER
  - i2s\_5411x.h, [730](#)
- EXTERN
  - LPC Public Macros, [475](#)
- emWin download and installation, [548](#)
- error.h
  - COMPILE\_TIME\_ASSERT, [705](#)
  - ERR\_ADC\_BASE, [709](#)
  - ERR\_ADC\_INVALID\_CHANNEL, [709](#)
  - ERR\_ADC\_INVALID\_LENGTH, [709](#)
  - ERR\_ADC\_INVALID\_SEQUENCE, [709](#)
  - ERR\_ADC\_INVALID\_SETUP, [709](#)
  - ERR\_ADC\_NO\_POWER, [709](#)
  - ERR\_ADC\_OVERRUN, [709](#)
  - ERR\_ADC\_PARAM, [709](#)
  - ERR\_API\_BASE, [706](#)
  - ERR\_API\_INVALID\_PARAM1, [706](#)
  - ERR\_API\_INVALID\_PARAM2, [706](#)
  - ERR\_API\_INVALID\_PARAM3, [706](#)
  - ERR\_API\_INVALID\_PARAMS, [706](#)
  - ERR\_API\_MOD\_INIT, [706](#)
  - ERR\_BUSY, [706](#)
  - ERR\_CAN\_BAD\_MEM\_BUF, [708](#)
  - ERR\_CAN\_BASE, [708](#)
  - ERR\_CAN\_INIT\_FAIL, [708](#)
  - ERR\_CANOPEN\_INIT\_FAIL, [708](#)
  - ERR\_CGU\_BASE, [707](#)
  - ERR\_CGU\_DIV\_SRC, [707](#)
  - ERR\_CGU\_DIV\_VAL, [707](#)
  - ERR\_CGU\_INVALID\_PARAM, [707](#)
  - ERR\_CGU\_INVALID\_SLICE, [707](#)
  - ERR\_CGU\_NOT\_IMPL, [707](#)
  - ERR\_CGU\_OUTPUT\_GEN, [707](#)
  - ERR\_CGU\_SRC, [707](#)
  - ERR\_CLK\_BASE, [708](#)
  - ERR\_CLK\_BASE\_OFF, [709](#)
  - ERR\_CLK\_CFG, [709](#)
  - ERR\_CLK\_DIV\_SRC, [708](#)
  - ERR\_CLK\_DIV\_VAL, [708](#)
  - ERR\_CLK\_INVALID\_PARAM, [708](#)
  - ERR\_CLK\_INVALID\_SLICE, [708](#)
  - ERR\_CLK\_NOT\_IMPL, [708](#)
  - ERR\_CLK\_OFF\_DEADLOCK, [709](#)
  - ERR\_CLK\_OSC\_FREQ, [709](#)
  - ERR\_CLK\_OUTPUT\_GEN, [708](#)
  - ERR\_CLK\_PLL\_FIN\_TOO\_LARGE, [708](#)
  - ERR\_CLK\_PLL\_FIN\_TOO\_SMALL, [708](#)
  - ERR\_CLK\_PLL\_FOUT\_TOO\_LARGE, [708](#)
  - ERR\_CLK\_PLL\_FOUT\_TOO\_SMALL, [708](#)
  - ERR\_CLK\_PLL\_MAX\_PCT, [709](#)
  - ERR\_CLK\_PLL\_MIN\_PCT, [709](#)
  - ERR\_CLK\_PLL\_NO\_SOLUTION, [709](#)
  - ERR\_CLK\_SRC, [708](#)
  - ERR\_CLK\_TIMEOUT, [709](#)
  - ERR\_DM\_BASE, [709](#)
  - ERR\_DM\_COMM\_FAIL, [709](#)
  - ERR\_DM\_NOT\_ENTERED, [709](#)
  - ERR\_DM\_UNKNOWN\_CMD, [709](#)
  - ERR\_DMA\_BASE, [709](#)
  - ERR\_DMA\_BUSY, [709](#)
  - ERR\_DMA\_CHANNEL\_DISABLED, [709](#)
  - ERR\_DMA\_CHANNEL\_NUMBER, [709](#)
  - ERR\_DMA\_CHANNEL\_VALID\_PENDING, [709](#)
  - ERR\_DMA\_ERROR\_INT, [709](#)
  - ERR\_DMA\_GENERAL, [709](#)
  - ERR\_DMA\_NOT\_ALIGNMENT, [709](#)
  - ERR\_DMA\_PARAM, [709](#)
  - ERR\_DMA\_PING\_PONG\_EN, [709](#)
  - ERR\_DMA\_QUEUE\_EMPTY, [709](#)
  - ERR\_FAILED, [706](#)
  - ERR\_I2C\_BASE, [707](#)
  - ERR\_I2C\_BUFFER\_OVERFLOW, [707](#)
  - ERR\_I2C\_BUFFER\_UNDERFLOW, [708](#)
  - ERR\_I2C\_BUSY, [707](#)
  - ERR\_I2C\_BYTE\_COUNT\_ERR, [707](#)
  - ERR\_I2C\_GENERAL\_FAILURE, [707](#)
  - ERR\_I2C\_LOSS\_OF\_ARBITRATION, [707](#)
  - ERR\_I2C\_LOSS\_OF\_ARBITRATION\_NAK\_BIT, [707](#)
  - ERR\_I2C\_NAK, [707](#)
  - ERR\_I2C\_PARAM, [708](#)
  - ERR\_I2C\_REGS\_SET\_TO\_DEFAULT, [708](#)
  - ERR\_I2C\_SLAVE\_NOT\_ADDRESSED, [707](#)
  - ERR\_I2C\_TIMEOUT, [708](#)
  - ERR\_ISP\_ADDR\_ERROR, [706](#)
  - ERR\_ISP\_ADDR\_NOT\_MAPPED, [706](#)

- ERR\_ISP\_BASE, [706](#)
- ERR\_ISP\_BUSY, [706](#)
- ERR\_ISP\_CMD\_LOCKED, [706](#)
- ERR\_ISP\_CODE\_READ\_PROTECTION\_ENAB↵LED, [706](#)
- ERR\_ISP\_COMPARE\_ERROR, [706](#)
- ERR\_ISP\_COUNT\_ERROR, [706](#)
- ERR\_ISP\_DST\_ADDR\_ERROR, [706](#)
- ERR\_ISP\_DST\_ADDR\_NOT\_MAPPED, [706](#)
- ERR\_ISP\_EEPROM\_NO\_CLOCK, [706](#)
- ERR\_ISP\_EEPROM\_NO\_POWER, [706](#)
- ERR\_ISP\_FLASH\_NO\_CLOCK, [706](#)
- ERR\_ISP\_FLASH\_NO\_POWER, [706](#)
- ERR\_ISP\_INVALID\_BAUD\_RATE, [706](#)
- ERR\_ISP\_INVALID\_CODE, [706](#)
- ERR\_ISP\_INVALID\_COMMAND, [706](#)
- ERR\_ISP\_INVALID\_FLASH\_UNIT, [706](#)
- ERR\_ISP\_INVALID\_SECTOR, [706](#)
- ERR\_ISP\_INVALID\_STOP\_BIT, [706](#)
- ERR\_ISP\_IRC\_NO\_POWER, [706](#)
- ERR\_ISP\_PARAM\_ERROR, [706](#)
- ERR\_ISP\_REINVOKE\_ISP\_CONFIG, [706](#)
- ERR\_ISP\_SECTOR\_NOT\_BLANK, [706](#)
- ERR\_ISP\_SECTOR\_NOT\_PREPARED\_FOR↵WRITE OPERATION, [706](#)
- ERR\_ISP\_SETTING\_ACTIVE\_PARTITION, [706](#)
- ERR\_ISP\_SRC\_ADDR\_ERROR, [706](#)
- ERR\_ISP\_SRC\_ADDR\_NOT\_MAPPED, [706](#)
- ERR\_ISP\_USER\_CODE\_CHECKSUM, [706](#)
- ERR\_OTP\_AES\_KEYS\_ENABLED, [708](#)
- ERR\_OTP\_ALL\_DATA\_OR\_MASK\_ZERO, [708](#)
- ERR\_OTP\_BASE, [708](#)
- ERR\_OTP\_ETH\_MAC\_ENABLED, [708](#)
- ERR\_OTP\_ILLEGAL\_BANK, [708](#)
- ERR\_OTP\_READ\_DATA\_MISMATCH, [708](#)
- ERR\_OTP\_SOME\_BITS\_ALREADY\_PROGRA↵MMED, [708](#)
- ERR\_OTP\_USB\_ID\_ENABLED, [708](#)
- ERR\_OTP\_WR\_ENABLE\_INVALID, [708](#)
- ERR\_OTP\_WRITE\_ACCESS\_LOCKED, [708](#)
- ERR\_PWR\_BASE, [709](#)
- ERR\_SEC\_AES\_BASE, [707](#)
- ERR\_SEC\_AES\_DMA\_CHANNEL\_CFG, [707](#)
- ERR\_SEC\_AES\_DMA\_MUX\_CFG, [707](#)
- ERR\_SEC\_AES\_KEY\_ALREADY\_PROGRAM↵MED, [707](#)
- ERR\_SEC\_AES\_NOT\_SUPPORTED, [707](#)
- ERR\_SEC\_AES\_WRONG\_CMD, [707](#)
- ERR\_SPI\_BASE, [709](#)
- ERR\_SPI\_BUSY, [709](#)
- ERR\_SPI\_CLKSTALL, [709](#)
- ERR\_SPI\_INVALID\_LENGTH, [709](#)
- ERR\_SPI\_PARAM, [709](#)
- ERR\_SPI\_RXOVERRUN, [709](#)
- ERR\_SPI\_SELNASSERT, [709](#)
- ERR\_SPI\_SELNDEASSERT, [709](#)
- ERR\_SPI\_TXUNDERRUN, [709](#)
- ERR\_SPIFI\_BASE, [706](#)
- ERR\_SPIFI\_DEVICE\_ERROR, [706](#)
- ERR\_SPIFI\_ERASE\_NEEDED, [707](#)
- ERR\_SPIFI\_INTERNAL\_ERROR, [706](#)
- ERR\_SPIFI\_LITE\_BASE, [708](#)
- ERR\_SPIFI\_LITE\_BUSY, [708](#)
- ERR\_SPIFI\_LITE\_IN\_DMA, [708](#)
- ERR\_SPIFI\_LITE\_INVALID\_ARGUMENTS, [708](#)
- ERR\_SPIFI\_LITE\_MEMORY\_MODE\_OFF, [708](#)
- ERR\_SPIFI\_LITE\_MEMORY\_MODE\_ON, [708](#)
- ERR\_SPIFI\_LITE\_NOT\_IN\_DMA, [708](#)
- ERR\_SPIFI\_NO\_DEVICE, [707](#)
- ERR\_SPIFI\_OPERAND\_ERROR, [707](#)
- ERR\_SPIFI\_STATUS\_PROBLEM, [707](#)
- ERR\_SPIFI\_TIMEOUT, [706](#)
- ERR\_SPIFI\_UNKNOWN\_EXT, [707](#)
- ERR\_SPIFI\_UNKNOWN\_ID, [707](#)
- ERR\_SPIFI\_UNKNOWN\_MFG, [707](#)
- ERR\_SPIFI\_UNKNOWN\_TYPE, [707](#)
- ERR\_TIME\_OUT, [706](#)
- ERR\_UART\_BASE, [708](#)
- ERR\_UART\_BAUDRATE, [708](#)
- ERR\_UART\_OVERRUN\_FRAME\_PARITY\_NOI↵SE, [708](#)
- ERR\_UART\_PARAM, [708](#)
- ERR\_UART\_RXD\_BUSY, [708](#)
- ERR\_UART\_TXD\_BUSY, [708](#)
- ERR\_UART\_UNDERRUN, [708](#)
- ERR\_USBD\_BAD\_CFG\_DESC, [707](#)
- ERR\_USBD\_BAD\_DESC, [707](#)
- ERR\_USBD\_BAD\_EP\_DESC, [707](#)
- ERR\_USBD\_BAD\_INTF\_DESC, [707](#)
- ERR\_USBD\_BAD\_MEM\_BUF, [707](#)
- ERR\_USBD\_BASE, [707](#)
- ERR\_USBD\_INVALID\_REQ, [707](#)
- ERR\_USBD\_SEND\_DATA, [707](#)
- ERR\_USBD\_SEND\_ZLP, [707](#)
- ERR\_USBD\_STALL, [707](#)
- ERR\_USBD\_TOO\_MANY\_CLASS\_HDLR, [707](#)
- ERR\_USBD\_UNHANDLED, [707](#)
- ErrorCode\_t, [705](#)
- LPC\_OK, [706](#)
- offsetof, [705](#)
- PENDING\_SPIFI\_LITE, [708](#)
- PWR\_ERROR\_CLOCK\_FREQ\_TOO\_HIGH, [709](#)
- PWR\_ERROR\_ILLEGAL\_MODE, [709](#)
- PWR\_ERROR\_INVALID\_CFG, [709](#)
- PWR\_ERROR\_INVALID\_STATE, [709](#)
- PWR\_ERROR\_PVT\_DETECT, [709](#)
- SEC\_AES\_DMA\_BUSY, [707](#)
- SEC\_AES\_NO\_ERROR, [707](#)
- ErrorCode\_t
  - error.h, [705](#)
- eventCB
  - SPIM\_XFER\_T, [640](#)
  - SPIS\_XFER\_T, [642](#)
- ExtClockIn
  - CHIP: LPC5411x Chip driver build time options, [388](#)

- FALL
  - LPC\_PIN\_INT\_T, [592](#)
- FALSE
  - LPC Public Types, [479](#)
- FEED
  - LPC\_WWDT\_T, [625](#)
- FFCR
  - TPI\_Type, [645](#)
- FFSR
  - TPI\_Type, [645](#)
- FIFO0
  - TPI\_Type, [645](#)
- FIFO1
  - TPI\_Type, [645](#)
- FIFO\_CTRL
  - CHIP: LPC5411X DMIC driver, [144](#)
- FIFO\_DATA
  - CHIP: LPC5411X DMIC driver, [144](#)
- FIFO\_STATUS
  - CHIP: LPC5411X DMIC driver, [144](#)
- FIFOCFG
  - LPC\_I2S\_T, [584](#)
  - LPC\_SPI\_T, [605](#)
  - LPC\_USART\_T, [622](#)
- FIFOINTENCLR
  - LPC\_I2S\_T, [584](#)
  - LPC\_SPI\_T, [605](#)
  - LPC\_USART\_T, [622](#)
- FIFOINTENSET
  - LPC\_I2S\_T, [584](#)
  - LPC\_SPI\_T, [605](#)
  - LPC\_USART\_T, [622](#)
- FIFOINTSTAT
  - LPC\_I2S\_T, [585](#)
  - LPC\_SPI\_T, [605](#)
  - LPC\_USART\_T, [622](#)
- FIFORD
  - LPC\_I2S\_T, [585](#)
  - LPC\_SPI\_T, [605](#)
  - LPC\_USART\_T, [622](#)
- FIFORD48HNOPOP
  - LPC\_I2S\_T, [585](#)
- FIFORDNOPOP
  - LPC\_I2S\_T, [585](#)
  - LPC\_SPI\_T, [605](#)
  - LPC\_USART\_T, [622](#)
- FIFOSTAT
  - LPC\_I2S\_T, [585](#)
  - LPC\_SPI\_T, [605](#)
  - LPC\_USART\_T, [622](#)
- FIFOTRIG
  - LPC\_I2S\_T, [585](#)
  - LPC\_SPI\_T, [605](#)
  - LPC\_USART\_T, [622](#)
- FIFOWR
  - LPC\_I2S\_T, [585](#)
  - LPC\_SPI\_T, [606](#)
  - LPC\_USART\_T, [623](#)
- FIFOdepth
  - I2S\_AUDIO\_FORMAT\_T, [562](#)
- FLAGS
  - LPC\_ADC\_T, [571](#)
- FLASHCFG
  - LPC\_SYSCON\_T, [611](#)
- FLASHTIM\_20MHZ\_CPU
  - CHIP: LPC5411X System and Control Driver, [333](#)
- FLEXCOMM0\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- FLEXCOMM1\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- FLEXCOMM2\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- FLEXCOMM3\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- FLEXCOMM4\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- FLEXCOMM5\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- FLEXCOMM6\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- FLEXCOMM7\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- FLEXCOMM\_ID\_I2C
  - CHIP: LPC5411X flexcomm API, [379](#)
- FLEXCOMM\_ID\_I2S
  - CHIP: LPC5411X flexcomm API, [379](#)
- FLEXCOMM\_ID\_SPI
  - CHIP: LPC5411X flexcomm API, [379](#)
- FLEXCOMM\_ID\_USART
  - CHIP: LPC5411X flexcomm API, [379](#)
- FLEXCOMM\_LOCK
  - CHIP: LPC5411X flexcomm API, [379](#)



- FLEXCOMM\_PERIPH\_I2C
  - CHIP: LPC5411X flexcomm API, [380](#)
- FLEXCOMM\_PERIPH\_I2S\_RX
  - CHIP: LPC5411X flexcomm API, [380](#)
- FLEXCOMM\_PERIPH\_I2S\_TX
  - CHIP: LPC5411X flexcomm API, [380](#)
- FLEXCOMM\_PERIPH\_NONE
  - CHIP: LPC5411X flexcomm API, [380](#)
- FLEXCOMM\_PERIPH\_SPI
  - CHIP: LPC5411X flexcomm API, [380](#)
- FLEXCOMM\_PERIPH\_T
  - CHIP: LPC5411X flexcomm API, [380](#)
- FLEXCOMM\_PERIPH\_USART
  - CHIP: LPC5411X flexcomm API, [380](#)
- FLEXCOMM\_PSEL\_OFFSET
  - CHIP: LPC5411X flexcomm API, [379](#)
- FOLDCNT
  - DWT\_Type, [558](#)
- FPCA
  - CONTROL\_Type, [552](#)
- FREQ\_MEAS\_MAIN\_CLK
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FREQMEAS\_REF
  - LPC\_INMUX\_T, [587](#)
- FREQMEAS\_TARGET
  - LPC\_INMUX\_T, [587](#)
- FREQMECTRL
  - LPC\_SYSCON\_T, [611](#)
- FREQMSR\_32KHZOSC
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FREQMSR\_CLKIN
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FREQMSR\_FRO12MHZ
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FREQMSR\_PIO0\_20
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FREQMSR\_PIO0\_24
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FREQMSR\_PIO0\_4
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FREQMSR\_PIO1\_4
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FREQMSR\_SRC\_T
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FREQMSR\_WDOSC
  - CHIP: LPC5411X Input Mux Registers and Driver, [195](#)
- FRGCLKSEL
  - LPC\_SYSCON\_T, [611](#)
- FRGCTRL
  - LPC\_SYSCON\_T, [611](#)
- FROCTRL
  - LPC\_SYSCON\_T, [611](#)
- FSCR
  - TPI\_Type, [646](#)
- FUNCTION0
  - DWT\_Type, [558](#)
- FUNCTION1
  - DWT\_Type, [558](#)
- FUNCTION2
  - DWT\_Type, [558](#)
- FUNCTION3
  - DWT\_Type, [558](#)
- FXCOMCLKSEL
  - LPC\_SYSCON\_T, [611](#)
- fifo\_config\_dma
  - uart\_5411x.c, [805](#)
- fifo\_config\_int
  - uart\_5411x.c, [805](#)
- fifo\_ctrl\_rx
  - i2s\_5411x.c, [782](#)
- fifo\_ctrl\_tx
  - i2s\_5411x.c, [782](#)
- fifo\_err\_rx
  - I2S\_STATISTICS\_T, [564](#)
  - UART\_STATISTICS\_T, [649](#)
- fifo\_err\_tx
  - I2S\_STATISTICS\_T, [564](#)
  - UART\_STATISTICS\_T, [649](#)
- fifo\_ints
  - CHIP: LPC5411X DMIC driver, [144](#)
- fifo\_overrun
  - CHIP: LPC5411X DMIC driver, [144](#)
- fifo\_underrun
  - CHIP: LPC5411X DMIC driver, [144](#)
- FindGreatestCommonDivisor
  - pll\_5411x.c, [795](#)
- findPIIMult
  - pll\_5411x.c, [795](#)
- findPIIPostDiv
  - pll\_5411x.c, [795](#)
- findPIIPreDiv
  - pll\_5411x.c, [795](#)
- FlagStatus
  - LPC Public Types, [480](#)
- flags
  - PLL\_CONFIG\_T, [630](#)
  - PLL\_SETUP\_T, [631](#)
- fpuInit
  - CHIP: FPU initialization, [54](#)
- FrameDelay
  - SPIM\_DELAY\_CONFIG\_T, [639](#)
- FrameWidth
  - I2S\_AUDIO\_FORMAT\_T, [562](#)
- FunctionalState
  - LPC Public Types, [480](#)
- Functions and Instructions Reference, [465](#)



- GAINSHFT
  - CHIP: LPC5411X DMIC driver, [145](#)
- GE
  - APSR\_Type, [550](#)
  - xPSR\_Type, [667](#)
- GINT0\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- GINT1\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- GPIOGR\_COMB
  - CHIP: LPC5411X GPIO group driver, [166](#)
- GPIOGR\_INT
  - CHIP: LPC5411X GPIO group driver, [166](#)
- GPIOGR\_TRIG
  - CHIP: LPC5411X GPIO group driver, [166](#)
- gainshft
  - CHIP: LPC5411X DMIC driver, [145](#)
- get\_tx\_data
  - spim\_5411x.c, [799](#)
  - spis\_5411x.c, [800](#)
- GetDMLY
  - rtc\_ut.c, [798](#)
- H
  - \_WB\_T, [665](#)
  - LPC\_SCT\_T, [601](#)
- HALT\_H
  - LPC\_SCT\_T, [601](#)
- HALT\_L
  - LPC\_SCT\_T, [601](#)
- HFSR
  - SCB\_Type, [635](#)
- HOURSPERDAY
  - rtc\_ut.c, [797](#)
- HWVAD
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
- HWVAD\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- HWVADGAIN
  - CHIP: LPC5411X DMIC driver, [145](#)
- HWVADHPFS
  - CHIP: LPC5411X DMIC driver, [145](#)
- HWVADLOWZ
  - CHIP: LPC5411X DMIC driver, [145](#)
- HWVADRSTT
  - CHIP: LPC5411X DMIC driver, [145](#)
- HWVADST10
  - CHIP: LPC5411X DMIC driver, [145](#)
- HWVADTHGN
  - CHIP: LPC5411X DMIC driver, [145](#)
- HWVADTHGS
  - CHIP: LPC5411X DMIC driver, [145](#)
- HardFault\_Handler
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [807](#)
- HardFault\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- head
  - RINGBUFF\_T, [633](#)
- hostIrqPortPin
  - PINTABLE\_T, [629](#)
- hostMisoPortPin
  - PINTABLE\_T, [629](#)
- hostMosiPortPin
  - PINTABLE\_T, [629](#)
- hostSckPortPin
  - PINTABLE\_T, [629](#)
- hostSselPortPin
  - PINTABLE\_T, [629](#)
- I2C\_CFG\_MASK
  - CHIP: LPC5411x I2C driver, [393](#)
- I2C\_CFG\_MONCLKSTR
  - CHIP: LPC5411x I2C driver, [393](#)
- I2C\_CFG\_MONEN
  - CHIP: LPC5411x I2C driver, [393](#)
- I2C\_CFG\_MSTEN
  - CHIP: LPC5411x I2C driver, [393](#)
- I2C\_CFG\_SLVEN
  - CHIP: LPC5411x I2C driver, [393](#), [394](#)
- I2C\_CFG\_TIMEOUTEN
  - CHIP: LPC5411x I2C driver, [394](#)
- I2C\_INTENCLR\_EVENTTIMEOUT
  - CHIP: LPC5411x I2C driver, [394](#)
- I2C\_INTENCLR\_MONIDLE
  - CHIP: LPC5411x I2C driver, [394](#)
- I2C\_INTENCLR\_MONOV
  - CHIP: LPC5411x I2C driver, [394](#)
- I2C\_INTENCLR\_MONRDY
  - CHIP: LPC5411x I2C driver, [395](#)
- I2C\_INTENCLR\_MSTPENDING
  - CHIP: LPC5411x I2C driver, [395](#)
- I2C\_INTENCLR\_MSTRARBLOSS
  - CHIP: LPC5411x I2C driver, [395](#)
- I2C\_INTENCLR\_MSTSTSTPERR
  - CHIP: LPC5411x I2C driver, [395](#)
- I2C\_INTENCLR\_SCLTIMEOUT
  - CHIP: LPC5411x I2C driver, [395](#), [396](#)
- I2C\_INTENCLR\_SLVDESEL
  - CHIP: LPC5411x I2C driver, [396](#)
- I2C\_INTENCLR\_SLVNOTSTR
  - CHIP: LPC5411x I2C driver, [396](#)
- I2C\_INTENCLR\_SLVPENDING
  - CHIP: LPC5411x I2C driver, [396](#)
- I2C\_INTENSET\_EVENTTIMEOUT
  - CHIP: LPC5411x I2C driver, [396](#)
- I2C\_INTENSET\_MONIDLE

- CHIP: LPC5411x I2C driver, [397](#)
- I2C\_INTENSET\_MONOV
  - CHIP: LPC5411x I2C driver, [397](#)
- I2C\_INTENSET\_MONRDY
  - CHIP: LPC5411x I2C driver, [397](#)
- I2C\_INTENSET\_MSTPENDING
  - CHIP: LPC5411x I2C driver, [397](#)
- I2C\_INTENSET\_MSTRARBLOSS
  - CHIP: LPC5411x I2C driver, [397](#), [398](#)
- I2C\_INTENSET\_MSTSTSTPERR
  - CHIP: LPC5411x I2C driver, [398](#)
- I2C\_INTENSET\_SCLTIMEOUT
  - CHIP: LPC5411x I2C driver, [398](#)
- I2C\_INTENSET\_SLVDESEL
  - CHIP: LPC5411x I2C driver, [398](#)
- I2C\_INTENSET\_SLVNOTSTR
  - CHIP: LPC5411x I2C driver, [398](#)
- I2C\_INTENSET\_SLVPENDING
  - CHIP: LPC5411x I2C driver, [399](#)
- I2C\_INTSTAT\_EVENTTIMEOUT
  - CHIP: LPC5411x I2C driver, [399](#)
- I2C\_INTSTAT\_MONIDLE
  - CHIP: LPC5411x I2C driver, [399](#)
- I2C\_INTSTAT\_MONOV
  - CHIP: LPC5411x I2C driver, [399](#)
- I2C\_INTSTAT\_MONRDY
  - CHIP: LPC5411x I2C driver, [399](#), [400](#)
- I2C\_INTSTAT\_MSTPENDING
  - CHIP: LPC5411x I2C driver, [400](#)
- I2C\_INTSTAT\_MSTRARBLOSS
  - CHIP: LPC5411x I2C driver, [400](#)
- I2C\_INTSTAT\_MSTSTSTPERR
  - CHIP: LPC5411x I2C driver, [400](#)
- I2C\_INTSTAT\_SCLTIMEOUT
  - CHIP: LPC5411x I2C driver, [400](#)
- I2C\_INTSTAT\_SLVDESEL
  - CHIP: LPC5411x I2C driver, [401](#)
- I2C\_INTSTAT\_SLVNOTSTR
  - CHIP: LPC5411x I2C driver, [401](#)
- I2C\_INTSTAT\_SLVPENDING
  - CHIP: LPC5411x I2C driver, [401](#)
- I2C\_MONRXDAT\_DATA
  - CHIP: LPC5411x I2C driver, [401](#)
- I2C\_MONRXDAT\_MONNACK
  - CHIP: LPC5411x I2C driver, [401](#), [402](#)
- I2C\_MONRXDAT\_MONRESTART
  - CHIP: LPC5411x I2C driver, [402](#)
- I2C\_MONRXDAT\_MONSTART
  - CHIP: LPC5411x I2C driver, [402](#)
- I2C\_MSTCTL\_MSTCONTINUE
  - CHIP: LPC5411x I2C driver, [402](#)
- I2C\_MSTCTL\_MSTDMA
  - CHIP: LPC5411x I2C driver, [402](#)
- I2C\_MSTCTL\_MSTSTART
  - CHIP: LPC5411x I2C driver, [403](#)
- I2C\_MSTCTL\_MSTSTOP
  - CHIP: LPC5411x I2C driver, [403](#)
- I2C\_MSTDAT\_DATAMASK
  - CHIP: LPC5411x I2C driver, [403](#)
- I2C\_MSTTIME\_MSTSCLHIGH
  - CHIP: LPC5411x I2C driver, [403](#)
- I2C\_MSTTIME\_MSTSCLLOW
  - CHIP: LPC5411x I2C driver, [403](#), [404](#)
- I2C\_SLVADR\_MASK
  - CHIP: LPC5411x I2C driver, [404](#)
- I2C\_SLVADR\_SADISABLE
  - CHIP: LPC5411x I2C driver, [404](#)
- I2C\_SLVADR\_SLVADR
  - CHIP: LPC5411x I2C driver, [404](#)
- I2C\_SLVCTL\_SLVCONTINUE
  - CHIP: LPC5411x I2C driver, [404](#)
- I2C\_SLVCTL\_SLVDMA
  - CHIP: LPC5411x I2C driver, [405](#)
- I2C\_SLVCTL\_SLVNACK
  - CHIP: LPC5411x I2C driver, [405](#)
- I2C\_SLVDAT\_DATAMASK
  - CHIP: LPC5411x I2C driver, [405](#)
- I2C\_SLVQUAL\_QUALMODE0
  - CHIP: LPC5411x I2C driver, [405](#)
- I2C\_SLVQUAL\_SLVQUAL0
  - CHIP: LPC5411x I2C driver, [405](#), [406](#)
- I2C\_STAT\_EVENTTIMEOUT
  - CHIP: LPC5411x I2C driver, [406](#)
- I2C\_STAT\_MONACTIVE
  - CHIP: LPC5411x I2C driver, [406](#)
- I2C\_STAT\_MONIDLE
  - CHIP: LPC5411x I2C driver, [406](#)
- I2C\_STAT\_MONOV
  - CHIP: LPC5411x I2C driver, [406](#)
- I2C\_STAT\_MONRDY
  - CHIP: LPC5411x I2C driver, [407](#)
- I2C\_STAT\_MSTCODE\_IDLE
  - CHIP: LPC5411x I2C driver, [407](#)
- I2C\_STAT\_MSTCODE\_NACKADR
  - CHIP: LPC5411x I2C driver, [407](#)
- I2C\_STAT\_MSTCODE\_NACKDAT
  - CHIP: LPC5411x I2C driver, [407](#)
- I2C\_STAT\_MSTCODE\_RXREADY
  - CHIP: LPC5411x I2C driver, [407](#), [408](#)
- I2C\_STAT\_MSTCODE\_TXREADY
  - CHIP: LPC5411x I2C driver, [408](#)
- I2C\_STAT\_MSTPENDING
  - CHIP: LPC5411x I2C driver, [408](#)
- I2C\_STAT\_MSTRARBLOSS
  - CHIP: LPC5411x I2C driver, [408](#)
- I2C\_STAT\_MSTSTATE
  - CHIP: LPC5411x I2C driver, [408](#)
- I2C\_STAT\_MSTSTSTPERR
  - CHIP: LPC5411x I2C driver, [409](#)
- I2C\_STAT\_SCLTIMEOUT
  - CHIP: LPC5411x I2C driver, [409](#)
- I2C\_STAT\_SLVCODE\_ADDR
  - CHIP: LPC5411x I2C driver, [409](#)
- I2C\_STAT\_SLVCODE\_RX
  - CHIP: LPC5411x I2C driver, [409](#)
- I2C\_STAT\_SLVCODE\_TX
  - CHIP: LPC5411x I2C driver, [409](#)

- CHIP: LPC5411x I2C driver, [409](#), [410](#)
- I2C\_STAT\_SLVDESEL
  - CHIP: LPC5411x I2C driver, [410](#)
- I2C\_STAT\_SLVIDX
  - CHIP: LPC5411x I2C driver, [410](#)
- I2C\_STAT\_SLVNOTSTR
  - CHIP: LPC5411x I2C driver, [410](#)
- I2C\_STAT\_SLVPENDING
  - CHIP: LPC5411x I2C driver, [410](#)
- I2C\_STAT\_SLVSEL
  - CHIP: LPC5411x I2C driver, [411](#)
- I2C\_STAT\_SLVSTATE
  - CHIP: LPC5411x I2C driver, [411](#)
- I2C\_TIMEOUT\_VAL
  - CHIP: LPC5411x I2C driver, [411](#)
- I2CM\_STATUS\_ARBLOST
  - CHIP: LPC5411X I2C master-only driver, [172](#)
- I2CM\_STATUS\_BUS\_ERROR
  - CHIP: LPC5411X I2C master-only driver, [172](#)
- I2CM\_STATUS\_BUSY
  - CHIP: LPC5411X I2C master-only driver, [172](#)
- I2CM\_STATUS\_ERROR
  - CHIP: LPC5411X I2C master-only driver, [172](#)
- I2CM\_STATUS\_NAK\_ADR
  - CHIP: LPC5411X I2C master-only driver, [172](#)
- I2CM\_STATUS\_NAK\_DAT
  - CHIP: LPC5411X I2C master-only driver, [172](#)
- I2CM\_STATUS\_OK
  - CHIP: LPC5411X I2C master-only driver, [172](#)
- I2CM\_XFER\_T, [559](#)
  - rxBuff, [560](#)
  - rxSz, [560](#)
  - slaveAddr, [560](#)
  - status, [560](#)
  - txBuff, [560](#)
  - txSz, [560](#)
- I2CS\_XFER\_T, [560](#)
  - slaveDone, [561](#)
  - slaveRecv, [561](#)
  - slaveSend, [561](#)
  - slaveStart, [561](#)
- I2CSlaveXferDone
  - CHIP: LPC5411X I2C slave-only driver, [180](#)
- I2CSlaveXferRecv
  - CHIP: LPC5411X I2C slave-only driver, [180](#)
- I2CSlaveXferSend
  - CHIP: LPC5411X I2C slave-only driver, [180](#)
- I2CSlaveXferStart
  - CHIP: LPC5411X I2C slave-only driver, [181](#)
- I2S6\_IRQHandler
  - i2s\_5411x.h, [723](#)
- I2S6\_IRQn
  - i2s\_5411x.h, [723](#)
- I2S7\_IRQHandler
  - i2s\_5411x.h, [723](#)
- I2S7\_IRQn
  - i2s\_5411x.h, [723](#)
- I2S\_AUDIO\_FORMAT\_T, [562](#)
- ChannelNumber, [562](#)
- DataPos, [562](#)
- Direction, [562](#)
- Divider, [562](#)
- FIFOdepth, [562](#)
- FrameWidth, [562](#)
- LeftJust, [563](#)
- MSCfg, [563](#)
- Mode, [563](#)
- PDMDData, [563](#)
- RightLow, [563](#)
- SCKPol, [563](#)
- WSPol, [563](#)
- WordWidth, [563](#)
- I2S\_CFG1\_2NDCOUNT
  - i2s\_5411x.h, [723](#)
- I2S\_CFG1\_DATALEN
  - i2s\_5411x.h, [723](#)
- I2S\_CFG1\_DATAPAUSE
  - i2s\_5411x.h, [724](#)
- I2S\_CFG1\_LEFTJUST
  - i2s\_5411x.h, [724](#)
- I2S\_CFG1\_MAINENABLE
  - i2s\_5411x.h, [724](#)
- I2S\_CFG1\_MODE
  - i2s\_5411x.h, [724](#)
- I2S\_CFG1\_MSTSLVCFG
  - i2s\_5411x.h, [724](#)
- I2S\_CFG1\_ONECHANNEL
  - i2s\_5411x.h, [724](#)
- I2S\_CFG1\_PDMDATA
  - i2s\_5411x.h, [724](#)
- I2S\_CFG1\_RIGHTLOW
  - i2s\_5411x.h, [724](#)
- I2S\_CFG1\_SCK\_POL
  - i2s\_5411x.h, [724](#)
- I2S\_CFG1\_WS\_POL
  - i2s\_5411x.h, [725](#)
- I2S\_CFG2\_FRAMELEN
  - i2s\_5411x.h, [725](#)
- I2S\_CFG2\_POSITION
  - i2s\_5411x.h, [725](#)
- I2S\_CLASSIC
  - i2s\_5411x.h, [730](#)
- I2S\_DIR\_T
  - i2s\_5411x.h, [729](#)
- I2S\_FIFO\_CFG\_DMARX
  - i2s\_5411x.h, [725](#)
- I2S\_FIFO\_CFG\_DMATX
  - i2s\_5411x.h, [725](#)
- I2S\_FIFO\_CFG\_EMPTYRX
  - i2s\_5411x.h, [725](#)
- I2S\_FIFO\_CFG\_EMPTYTX
  - i2s\_5411x.h, [725](#)
- I2S\_FIFO\_CFG\_ENABLERX
  - i2s\_5411x.h, [725](#)
- I2S\_FIFO\_CFG\_ENABLETX
  - i2s\_5411x.h, [725](#)

- I2S\_FIFO\_CFG\_PACK48
  - i2s\_5411x.h, [726](#)
- I2S\_FIFO\_CFG\_POPDBG
  - i2s\_5411x.h, [726](#)
- I2S\_FIFO\_CFG\_SIZE
  - i2s\_5411x.h, [726](#)
- I2S\_FIFO\_CFG\_TXI2SE0
  - i2s\_5411x.h, [726](#)
- I2S\_FIFO\_CFG\_WAKERX
  - i2s\_5411x.h, [726](#)
- I2S\_FIFO\_CFG\_WAKETX
  - i2s\_5411x.h, [726](#)
- I2S\_FIFO\_CLEAR
  - i2s\_5411x.h, [729](#)
- I2S\_FIFO\_CMD\_T
  - i2s\_5411x.h, [729](#)
- I2S\_FIFO\_DISABLE
  - i2s\_5411x.h, [729](#)
- I2S\_FIFO\_DMA\_DISABLE
  - i2s\_5411x.h, [729](#)
- I2S\_FIFO\_DMA\_ENABLE
  - i2s\_5411x.h, [729](#)
- I2S\_FIFO\_ENABLE
  - i2s\_5411x.h, [729](#)
- I2S\_FIFO\_INT\_BITMASK
  - i2s\_5411x.h, [726](#)
- I2S\_FIFO\_INT\_PERINT
  - i2s\_5411x.h, [726](#)
- I2S\_FIFO\_INT\_RXERR
  - i2s\_5411x.h, [726](#)
- I2S\_FIFO\_INT\_RXLVL
  - i2s\_5411x.h, [727](#)
- I2S\_FIFO\_INT\_TXERR
  - i2s\_5411x.h, [727](#)
- I2S\_FIFO\_INT\_TXLVL
  - i2s\_5411x.h, [727](#)
- I2S\_FIFO\_STAT\_PERINT
  - i2s\_5411x.h, [727](#)
- I2S\_FIFO\_STAT\_RXERR
  - i2s\_5411x.h, [727](#)
- I2S\_FIFO\_STAT\_RXFULL
  - i2s\_5411x.h, [727](#)
- I2S\_FIFO\_STAT\_RXLVL
  - i2s\_5411x.h, [727](#)
- I2S\_FIFO\_STAT\_RXNOTEMPTY
  - i2s\_5411x.h, [727](#)
- I2S\_FIFO\_STAT\_TXEMPTY
  - i2s\_5411x.h, [727](#)
- I2S\_FIFO\_STAT\_TXERR
  - i2s\_5411x.h, [728](#)
- I2S\_FIFO\_STAT\_TXLVL
  - i2s\_5411x.h, [728](#)
- I2S\_FIFO\_STAT\_TXNOTFULL
  - i2s\_5411x.h, [728](#)
- I2S\_FIFO\_TRIG\_RXLVL
  - i2s\_5411x.h, [728](#)
- I2S\_FIFO\_TRIG\_RXLVLENA
  - i2s\_5411x.h, [728](#)
- I2S\_FIFO\_TRIG\_TXLVL
  - i2s\_5411x.h, [728](#)
- I2S\_FIFO\_TRIG\_TXLVLENA
  - i2s\_5411x.h, [728](#)
- I2S\_FIFO\_TXZ\_DISABLE
  - i2s\_5411x.h, [730](#)
- I2S\_FIFO\_TXZ\_ENABLE
  - i2s\_5411x.h, [730](#)
- I2S\_MODE\_T
  - i2s\_5411x.h, [730](#)
- I2S\_MSTSLVCFG\_T
  - i2s\_5411x.h, [730](#)
- I2S\_RX
  - i2s\_5411x.h, [729](#)
- I2S\_STAT\_BUSY
  - i2s\_5411x.h, [728](#)
- I2S\_STAT\_LR
  - i2s\_5411x.h, [728](#)
- I2S\_STAT\_PAUSED
  - i2s\_5411x.h, [729](#)
- I2S\_STAT\_SLVFRMERR
  - i2s\_5411x.h, [729](#)
- I2S\_STATISTICS\_T, [564](#)
  - fifo\_err\_rx, [564](#)
  - fifo\_err\_tx, [564](#)
  - i2s\_busy, [564](#)
  - i2s\_data\_paused, [564](#)
  - i2s\_slvfrmerr, [564](#)
  - interrupts, [564](#)
  - lvl\_rx, [565](#)
  - lvl\_tx, [565](#)
- I2S\_TX
  - i2s\_5411x.h, [729](#)
- i2s\_5411x.c
  - Chip\_I2S\_Config, [781](#)
  - Chip\_I2S\_ErrorHandler, [781](#)
  - Chip\_I2S\_FIFO\_Config, [782](#)
  - Chip\_I2S\_FIFO\_Control, [782](#)
  - Chip\_I2S\_Init, [782](#)
  - fifo\_ctrl\_rx, [782](#)
  - fifo\_ctrl\_tx, [782](#)
- i2s\_5411x.h
  - Chip\_I2S\_ClearStatus, [730](#)
  - Chip\_I2S\_ClrFIFOStatus, [730](#)
  - Chip\_I2S\_Config, [731](#)
  - Chip\_I2S\_DeInit, [731](#)
  - Chip\_I2S\_ErrorHandler, [731](#)
  - Chip\_I2S\_FIFO\_ClearStatus, [731](#)
  - Chip\_I2S\_FIFO\_ClrInterrupt, [733](#)
  - Chip\_I2S\_FIFO\_Config, [733](#)
  - Chip\_I2S\_FIFO\_Control, [733](#)
  - Chip\_I2S\_FIFO\_GetPendingInts, [733](#)
  - Chip\_I2S\_FIFO\_SetInterrupt, [734](#)
  - Chip\_I2S\_GetFIFORxLevel, [734](#)
  - Chip\_I2S\_GetFIFOStatus, [734](#)
  - Chip\_I2S\_GetFIFOTrigLevel, [734](#)
  - Chip\_I2S\_GetFIFOTxLevel, [735](#)
  - Chip\_I2S\_GetStatus, [735](#)

- Chip\_I2S\_Init, [735](#)
- Chip\_I2S\_Pause, [735](#)
- Chip\_I2S\_Play, [736](#)
- Chip\_I2S\_RX\_Init, [736](#)
- Chip\_I2S\_Receive, [736](#)
- Chip\_I2S\_Send, [736](#)
- Chip\_I2S\_SetFIFOTrigLevel, [737](#)
- Chip\_I2S\_Start, [737](#)
- Chip\_I2S\_Stop, [737](#)
- Chip\_I2S\_TX\_Init, [738](#)
- DMAREQ\_I2S6\_RX, [723](#)
- DMAREQ\_I2S6\_TX, [723](#)
- DMAREQ\_I2S7\_RX, [723](#)
- DMAREQ\_I2S7\_TX, [723](#)
- DSP\_WS\_50, [730](#)
- DSP\_WS\_LONG, [730](#)
- DSP\_WS\_SHORT, [730](#)
- EXT\_SCLK\_MASTER, [730](#)
- I2S6\_IRQHandler, [723](#)
- I2S6\_IRQn, [723](#)
- I2S7\_IRQHandler, [723](#)
- I2S7\_IRQn, [723](#)
- I2S\_CFG1\_2NDCOUNT, [723](#)
- I2S\_CFG1\_DATALEN, [723](#)
- I2S\_CFG1\_DATAPAUSE, [724](#)
- I2S\_CFG1\_LEFTJUST, [724](#)
- I2S\_CFG1\_MAINENABLE, [724](#)
- I2S\_CFG1\_MODE, [724](#)
- I2S\_CFG1\_MSTSLVCFG, [724](#)
- I2S\_CFG1\_ONECHANNEL, [724](#)
- I2S\_CFG1\_PDMDATA, [724](#)
- I2S\_CFG1\_RIGHTLOW, [724](#)
- I2S\_CFG1\_SCK\_POL, [724](#)
- I2S\_CFG1\_WS\_POL, [725](#)
- I2S\_CFG2\_FRAMELEN, [725](#)
- I2S\_CFG2\_POSITION, [725](#)
- I2S\_CLASSIC, [730](#)
- I2S\_DIR\_T, [729](#)
- I2S\_FIFO\_CFG\_DMARX, [725](#)
- I2S\_FIFO\_CFG\_DMATX, [725](#)
- I2S\_FIFO\_CFG\_EMPTYRX, [725](#)
- I2S\_FIFO\_CFG\_EMPTYTX, [725](#)
- I2S\_FIFO\_CFG\_ENABLERX, [725](#)
- I2S\_FIFO\_CFG\_ENABLETX, [725](#)
- I2S\_FIFO\_CFG\_PACK48, [726](#)
- I2S\_FIFO\_CFG\_POPDBG, [726](#)
- I2S\_FIFO\_CFG\_SIZE, [726](#)
- I2S\_FIFO\_CFG\_TXI2SE0, [726](#)
- I2S\_FIFO\_CFG\_WAKERX, [726](#)
- I2S\_FIFO\_CFG\_WAKETX, [726](#)
- I2S\_FIFO\_CLEAR, [729](#)
- I2S\_FIFO\_CMD\_T, [729](#)
- I2S\_FIFO\_DISABLE, [729](#)
- I2S\_FIFO\_DMA\_DISABLE, [729](#)
- I2S\_FIFO\_DMA\_ENABLE, [729](#)
- I2S\_FIFO\_ENABLE, [729](#)
- I2S\_FIFO\_INT\_BITMASK, [726](#)
- I2S\_FIFO\_INT\_PERINT, [726](#)
- I2S\_FIFO\_INT\_RXERR, [726](#)
- I2S\_FIFO\_INT\_RXLVL, [727](#)
- I2S\_FIFO\_INT\_TXERR, [727](#)
- I2S\_FIFO\_INT\_TXLVL, [727](#)
- I2S\_FIFO\_STAT\_PERINT, [727](#)
- I2S\_FIFO\_STAT\_RXERR, [727](#)
- I2S\_FIFO\_STAT\_RXFULL, [727](#)
- I2S\_FIFO\_STAT\_RXLVL, [727](#)
- I2S\_FIFO\_STAT\_RXNOTEMPTY, [727](#)
- I2S\_FIFO\_STAT\_TXEMPTY, [727](#)
- I2S\_FIFO\_STAT\_TXERR, [728](#)
- I2S\_FIFO\_STAT\_TXLVL, [728](#)
- I2S\_FIFO\_STAT\_TXNOTFULL, [728](#)
- I2S\_FIFO\_TRIG\_RXLVL, [728](#)
- I2S\_FIFO\_TRIG\_RXLVLENA, [728](#)
- I2S\_FIFO\_TRIG\_TXLVL, [728](#)
- I2S\_FIFO\_TRIG\_TXLVLENA, [728](#)
- I2S\_FIFO\_TXZ\_DISABLE, [730](#)
- I2S\_FIFO\_TXZ\_ENABLE, [730](#)
- I2S\_MODE\_T, [730](#)
- I2S\_MSTSLVCFG\_T, [730](#)
- I2S\_RX, [729](#)
- I2S\_STAT\_BUSY, [728](#)
- I2S\_STAT\_LR, [728](#)
- I2S\_STAT\_PAUSED, [729](#)
- I2S\_STAT\_SLVFRMERR, [729](#)
- I2S\_TX, [729](#)
- LPC\_I2S6, [729](#)
- LPC\_I2S6\_BASE, [729](#)
- LPC\_I2S7, [729](#)
- LPC\_I2S7\_BASE, [729](#)
- NORMAL\_MASTER, [730](#)
- NORMAL\_SLAVE, [730](#)
- WS\_SYNC\_MASTER, [730](#)
- i2s\_busy
  - I2S\_STATISTICS\_T, [564](#)
- i2s\_data\_paused
  - I2S\_STATISTICS\_T, [564](#)
- i2s\_slvfrmerr
  - I2S\_STATISTICS\_T, [564](#)
- IABR
  - NVIC\_Type, [626](#)
- IAP\_ADDR\_ERROR
  - CHIP: Common Chip ISP/IAP commands and return codes, [47](#)
- IAP\_ADDR\_NOT\_MAPPED
  - CHIP: Common Chip ISP/IAP commands and return codes, [47](#)
- IAP\_BLANK\_CHECK\_SECTOR\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [47](#)
- IAP\_BUSY
  - CHIP: Common Chip ISP/IAP commands and return codes, [47](#)
- IAP\_CMD\_LOCKED
  - CHIP: Common Chip ISP/IAP commands and return codes, [47](#)
- IAP\_CMD\_SUCCESS

- CHIP: Common Chip ISP/IAP commands and return codes, [47](#)
- IAP\_COMPARE\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [47](#)
- IAP\_COMPARE\_ERROR
  - CHIP: Common Chip ISP/IAP commands and return codes, [47](#)
- IAP\_COUNT\_ERROR
  - CHIP: Common Chip ISP/IAP commands and return codes, [48](#)
- IAP\_CRP\_ENABLED
  - CHIP: Common Chip ISP/IAP commands and return codes, [48](#)
- IAP\_DST\_ADDR\_ERROR
  - CHIP: Common Chip ISP/IAP commands and return codes, [48](#)
- IAP\_DST\_ADDR\_NOT\_MAPPED
  - CHIP: Common Chip ISP/IAP commands and return codes, [48](#)
- IAP\_EEPROM\_READ
  - CHIP: Common Chip ISP/IAP commands and return codes, [48](#)
- IAP\_EEPROM\_WRITE
  - CHIP: Common Chip ISP/IAP commands and return codes, [48](#)
- IAP\_ENTRY\_LOCATION
  - CHIP: LPC5411X ROM API declarations and functions, [257](#)
- IAP\_ENTRY\_T
  - CHIP: Common Chip ISP/IAP commands and return codes, [50](#)
- IAP\_ERASE\_PAGE\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [48](#)
- IAP\_ERSECTOR\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [48](#)
- IAP\_INVALID\_BAUD\_RATE
  - CHIP: Common Chip ISP/IAP commands and return codes, [48](#)
- IAP\_INVALID\_CODE
  - CHIP: Common Chip ISP/IAP commands and return codes, [49](#)
- IAP\_INVALID\_COMMAND
  - CHIP: Common Chip ISP/IAP commands and return codes, [49](#)
- IAP\_INVALID\_SECTOR
  - CHIP: Common Chip ISP/IAP commands and return codes, [49](#)
- IAP\_INVALID\_STOP\_BIT
  - CHIP: Common Chip ISP/IAP commands and return codes, [49](#)
- IAP\_PARAM\_ERROR
  - CHIP: Common Chip ISP/IAP commands and return codes, [49](#)
- IAP\_PREWRITE\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [49](#)
- IAP\_READ\_BOOT\_CODE\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [49](#)
- IAP\_READ\_UID\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [49](#)
- IAP\_REINVOKE\_ISP\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [49](#)
- IAP\_REPID\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [50](#)
- IAP\_SECTOR\_NOT\_BLANK
  - CHIP: Common Chip ISP/IAP commands and return codes, [50](#)
- IAP\_SECTOR\_NOT\_PREPARED
  - CHIP: Common Chip ISP/IAP commands and return codes, [50](#)
- IAP\_SRC\_ADDR\_ERROR
  - CHIP: Common Chip ISP/IAP commands and return codes, [50](#)
- IAP\_SRC\_ADDR\_NOT\_MAPPED
  - CHIP: Common Chip ISP/IAP commands and return codes, [50](#)
- IAP\_WRISECTOR\_CMD
  - CHIP: Common Chip ISP/IAP commands and return codes, [50](#)
- IAR EWARM support in LPCOpen, [466](#)
- ICER
  - NVIC\_Type, [626](#)
- ICPR
  - NVIC\_Type, [626](#)
- ICSR
  - SCB\_Type, [635](#)
- ICTR
  - SCnSCB\_Type, [637](#)
- IConfiguration
  - \_USB\_OTHER\_SPEED\_CONFIGURATION, [663](#)
- iConfiguration
  - \_USB\_CONFIGURATION\_DESCRIPTOR, [652](#)
- ID
  - CHIP: LPC5411X DMIC driver, [145](#)
- IDLE\_CH
  - LPC\_MRT\_T, [591](#)
- IENF
  - LPC\_PIN\_INT\_T, [592](#)
- IENR
  - LPC\_PIN\_INT\_T, [592](#)
- IFSEL\_T
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- iFunction
  - \_USB\_IAD\_DESCRIPTOR, [660](#)
- ilInterface
  - \_USB\_INTERFACE\_DESCRIPTOR, [661](#)
- IMAGE\_BOOT\_BLOCK\_OFF



- CHIP: LPC5411X Enhanced boot block support, [148](#)
- IMAGE\_DUAL\_ENH\_SIG
  - CHIP: LPC5411X Enhanced boot block support, [148](#)
- IMAGE\_ENH\_BLOCK\_MARKER
  - CHIP: LPC5411X Enhanced boot block support, [148](#)
- IMAGE\_ENH\_MARKER\_OFF
  - CHIP: LPC5411X Enhanced boot block support, [148](#)
- IMAGE\_SINGLE\_ENH\_SIG
  - CHIP: LPC5411X Enhanced boot block support, [148](#)
- IMAGE\_T
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- IMCR
  - ITM\_Type, [567](#)
- IMG\_ISP\_WAIT
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- IMG\_JUST\_BOOT
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- IMG\_NO\_CRC
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- IMG\_NO\_WAIT
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- IMG\_NORMAL
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- iManufacturer
  - \_USB\_DEVICE\_DESCRIPTOR, [654](#)
- INLINE
  - LPC Public Types, [477](#)
- INPUT
  - LPC\_SCT\_T, [601](#)
- INSEL
  - LPC\_ADC\_T, [571](#)
- INT\_16
  - LPC Public Types, [478](#)
- INT\_32
  - LPC Public Types, [478](#)
- INT\_64
  - LPC Public Types, [478](#)
- INT\_8
  - LPC Public Types, [478](#)
- INTA
  - LPC\_DMA\_COMMON\_T, [577](#)
- INTB
  - LPC\_DMA\_COMMON\_T, [577](#)
- INTEN
  - LPC\_ADC\_T, [571](#)
- INTENCLR
  - CHIP: LPC5411x I2C driver, [414](#)
- LPC\_DMA\_COMMON\_T, [577](#)
- LPC\_SPI\_T, [606](#)
- LPC\_USART\_T, [623](#)
- INTENSET
  - CHIP: LPC5411x I2C driver, [414](#)
  - LPC\_DMA\_COMMON\_T, [577](#)
  - LPC\_SPI\_T, [606](#)
  - LPC\_USART\_T, [623](#)
- INTSTAT
  - CHIP: LPC5411x I2C driver, [414](#)
  - LPC\_DMA\_T, [579](#)
  - LPC\_SPI\_T, [606](#)
  - LPC\_USART\_T, [623](#)
- INTVAL
  - LPC\_MRT\_CH\_T, [590](#)
- IOCFG
  - CHIP: LPC5411X DMIC driver, [145](#)
- IOCON\_ANALOG\_EN
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_CLKDIV
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_DIGITAL\_EN
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_FASTI2C\_EN
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_FUNC0
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_FUNC1
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_FUNC2
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_FUNC3
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_FUNC4
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_FUNC5
  - CHIP: LPC5411X IOCON register block and driver, [189](#)
- IOCON\_FUNC6
  - CHIP: LPC5411X IOCON register block and driver, [190](#)
- IOCON\_FUNC7
  - CHIP: LPC5411X IOCON register block and driver, [190](#)
- IOCON\_GPIO\_MODE
  - CHIP: LPC5411X IOCON register block and driver, [190](#)
- IOCON\_HYS\_EN

- CHIP: LPC5411X IOCON register block and driver, [190](#)
- IOCON\_I2C\_SLEW
  - CHIP: LPC5411X IOCON register block and driver, [190](#)
- IOCON\_INPFILT\_OFF
  - CHIP: LPC5411X IOCON register block and driver, [190](#)
- IOCON\_INPFILT\_ON
  - CHIP: LPC5411X IOCON register block and driver, [190](#)
- IOCON\_INV\_EN
  - CHIP: LPC5411X IOCON register block and driver, [190](#)
- IOCON\_MODE\_INACT
  - CHIP: LPC5411X IOCON register block and driver, [190](#)
- IOCON\_MODE\_PULLDOWN
  - CHIP: LPC5411X IOCON register block and driver, [191](#)
- IOCON\_MODE\_PULLUP
  - CHIP: LPC5411X IOCON register block and driver, [191](#)
- IOCON\_MODE\_REPEATER
  - CHIP: LPC5411X IOCON register block and driver, [191](#)
- IOCON\_OPENDRAIN\_EN
  - CHIP: LPC5411X IOCON register block and driver, [191](#)
- IOCON\_S\_MODE
  - CHIP: LPC5411X IOCON register block and driver, [191](#)
- IOCON\_S\_MODE\_0CLK
  - CHIP: LPC5411X IOCON register block and driver, [191](#)
- IOCON\_S\_MODE\_1CLK
  - CHIP: LPC5411X IOCON register block and driver, [191](#)
- IOCON\_S\_MODE\_2CLK
  - CHIP: LPC5411X IOCON register block and driver, [191](#)
- IOCON\_S\_MODE\_3CLK
  - CHIP: LPC5411X IOCON register block and driver, [191](#)
- IOCON\_STDI2C\_EN
  - CHIP: LPC5411X IOCON register block and driver, [192](#)
- IP
  - NVIC\_Type, [626](#)
- IPSR\_Type, [565](#)
  - \_reserved0, [565](#)
  - b, [565](#), [566](#)
  - ISR, [566](#)
  - w, [566](#)
- iProduct
  - \_USB\_DEVICE\_DESCRIPTOR, [654](#)
- IR
  - LPC\_TIMER\_T, [620](#)
- IRQ
  - LPC\_MBOXIRQ\_T, [589](#)
- IRQ\_FLAG
  - LPC\_MRT\_T, [591](#)
- IRQCLR
  - LPC\_MBOXIRQ\_T, [589](#)
- IRQSET
  - LPC\_MBOXIRQ\_T, [589](#)
- IRR
  - ITM\_Type, [567](#)
- ISAR
  - SCB\_Type, [635](#)
- ISEL
  - LPC\_PIN\_INT\_T, [592](#)
- ISER
  - NVIC\_Type, [626](#)
- ISPR
  - NVIC\_Type, [626](#)
- ISR
  - IPSR\_Type, [566](#)
  - xPSR\_Type, [667](#)
- IST
  - LPC\_PIN\_INT\_T, [592](#)
- iSerialNumber
  - \_USB\_DEVICE\_DESCRIPTOR, [654](#)
- IT
  - xPSR\_Type, [667](#)
- ITATBCTR0
  - TPI\_Type, [646](#)
- ITATBCTR2
  - TPI\_Type, [646](#)
- ITCTRL
  - TPI\_Type, [646](#)
- ITM
  - Core Definitions, [452](#)
- ITM Functions, [467](#)
  - ITM\_CheckChar, [467](#)
  - ITM\_RXBUFFER\_EMPTY, [467](#)
  - ITM\_ReceiveChar, [467](#)
  - ITM\_RxBuffer, [468](#)
  - ITM\_SendChar, [468](#)
- ITM\_BASE
  - Core Definitions, [452](#)
- ITM\_CheckChar
  - ITM Functions, [467](#)
- ITM\_IMCR\_INTEGRATION\_Msk
  - Instrumentation Trace Macrocell (ITM), [469](#)
- ITM\_IMCR\_INTEGRATION\_Pos
  - Instrumentation Trace Macrocell (ITM), [469](#)
- ITM\_IRR\_ATREADYM\_Msk
  - Instrumentation Trace Macrocell (ITM), [470](#)
- ITM\_IRR\_ATREADYM\_Pos
  - Instrumentation Trace Macrocell (ITM), [470](#)
- ITM\_IWR\_ATVALIDM\_Msk
  - Instrumentation Trace Macrocell (ITM), [470](#)
- ITM\_IWR\_ATVALIDM\_Pos
  - Instrumentation Trace Macrocell (ITM), [470](#)
- ITM\_LSR\_Access\_Msk



- Instrumentation Trace Macrocell (ITM), [470](#)
- ITM\_LSR\_Access\_Pos
  - Instrumentation Trace Macrocell (ITM), [470](#)
- ITM\_LSR\_ByteAcc\_Msk
  - Instrumentation Trace Macrocell (ITM), [470](#)
- ITM\_LSR\_ByteAcc\_Pos
  - Instrumentation Trace Macrocell (ITM), [470](#)
- ITM\_LSR\_Present\_Msk
  - Instrumentation Trace Macrocell (ITM), [470](#)
- ITM\_LSR\_Present\_Pos
  - Instrumentation Trace Macrocell (ITM), [471](#)
- ITM\_RXBUFFER\_EMPTY
  - ITM Functions, [467](#)
- ITM\_ReceiveChar
  - ITM Functions, [467](#)
- ITM\_RxBuffer
  - ITM Functions, [468](#)
- ITM\_SendChar
  - ITM Functions, [468](#)
- ITM\_TCR\_BUSY\_Msk
  - Instrumentation Trace Macrocell (ITM), [471](#)
- ITM\_TCR\_BUSY\_Pos
  - Instrumentation Trace Macrocell (ITM), [471](#)
- ITM\_TCR\_DWTENA\_Msk
  - Instrumentation Trace Macrocell (ITM), [471](#)
- ITM\_TCR\_DWTENA\_Pos
  - Instrumentation Trace Macrocell (ITM), [471](#)
- ITM\_TCR\_GTSFREQ\_Msk
  - Instrumentation Trace Macrocell (ITM), [471](#)
- ITM\_TCR\_GTSFREQ\_Pos
  - Instrumentation Trace Macrocell (ITM), [471](#)
- ITM\_TCR\_ITMENA\_Msk
  - Instrumentation Trace Macrocell (ITM), [471](#)
- ITM\_TCR\_ITMENA\_Pos
  - Instrumentation Trace Macrocell (ITM), [471](#)
- ITM\_TCR\_SWOENA\_Msk
  - Instrumentation Trace Macrocell (ITM), [472](#)
- ITM\_TCR\_SWOENA\_Pos
  - Instrumentation Trace Macrocell (ITM), [472](#)
- ITM\_TCR\_SYNCENA\_Msk
  - Instrumentation Trace Macrocell (ITM), [472](#)
- ITM\_TCR\_SYNCENA\_Pos
  - Instrumentation Trace Macrocell (ITM), [472](#)
- ITM\_TCR\_TSENA\_Msk
  - Instrumentation Trace Macrocell (ITM), [472](#)
- ITM\_TCR\_TSENA\_Pos
  - Instrumentation Trace Macrocell (ITM), [472](#)
- ITM\_TCR\_TSPrescale\_Msk
  - Instrumentation Trace Macrocell (ITM), [472](#)
- ITM\_TCR\_TSPrescale\_Pos
  - Instrumentation Trace Macrocell (ITM), [473](#)
- ITM\_TCR\_TraceBusID\_Msk
  - Instrumentation Trace Macrocell (ITM), [472](#)
- ITM\_TCR\_TraceBusID\_Pos
  - Instrumentation Trace Macrocell (ITM), [472](#)
- ITM\_TPR\_PRIVMASK\_Msk
  - Instrumentation Trace Macrocell (ITM), [473](#)
- ITM\_TPR\_PRIVMASK\_Pos
  - Instrumentation Trace Macrocell (ITM), [473](#)
- ITM\_Type, [566](#)
- CID0, [567](#)
- CID1, [567](#)
- CID2, [567](#)
- CID3, [567](#)
- IMCR, [567](#)
- IRR, [567](#)
- IWR, [567](#)
- LAR, [568](#)
- LSR, [568](#)
- PID0, [568](#)
- PID1, [568](#)
- PID2, [568](#)
- PID3, [568](#)
- PID4, [568](#)
- PID5, [568](#)
- PID6, [568](#)
- PID7, [569](#)
- PORT, [569](#)
- RESERVED0, [569](#)
- RESERVED1, [569](#)
- RESERVED2, [569](#)
- RESERVED3, [569](#)
- RESERVED4, [569](#)
- RESERVED5, [569](#)
- TCR, [569](#)
- TER, [569](#)
- TPR, [569](#)
- u16, [570](#)
- u32, [570](#)
- u8, [570](#)
- IWR
  - ITM\_Type, [567](#)
- iap\_entry
  - CHIP: LPC5411X ROM API declarations and functions, [257](#)
- idProduct
  - \_USB\_DEVICE\_DESCRIPTOR, [654](#)
- idVendor
  - \_USB\_DEVICE\_DESCRIPTOR, [654](#)
- ifSel
  - PINTABLE\_T, [629](#)
- img\_type
  - PINTABLE\_T, [629](#)
- InputRate
  - PLL\_CONFIG\_T, [630](#)
- Instrumentation Trace Macrocell (ITM), [469](#)
- ITM\_IMCR\_INTEGRATION\_Msk, [469](#)
- ITM\_IMCR\_INTEGRATION\_Pos, [469](#)
- ITM\_IRR\_ATREADYM\_Msk, [470](#)
- ITM\_IRR\_ATREADYM\_Pos, [470](#)
- ITM\_IWR\_ATVALIDM\_Msk, [470](#)
- ITM\_IWR\_ATVALIDM\_Pos, [470](#)
- ITM\_LSR\_Access\_Msk, [470](#)
- ITM\_LSR\_Access\_Pos, [470](#)
- ITM\_LSR\_ByteAcc\_Msk, [470](#)
- ITM\_LSR\_ByteAcc\_Pos, [470](#)

- ITM\_LSR\_Present\_Msk, [470](#)
- ITM\_LSR\_Present\_Pos, [471](#)
- ITM\_TCR\_BUSY\_Msk, [471](#)
- ITM\_TCR\_BUSY\_Pos, [471](#)
- ITM\_TCR\_DWTENA\_Msk, [471](#)
- ITM\_TCR\_DWTENA\_Pos, [471](#)
- ITM\_TCR\_GTSFREQ\_Msk, [471](#)
- ITM\_TCR\_GTSFREQ\_Pos, [471](#)
- ITM\_TCR\_ITMENA\_Msk, [471](#)
- ITM\_TCR\_ITMENA\_Pos, [471](#)
- ITM\_TCR\_SWOENA\_Msk, [472](#)
- ITM\_TCR\_SWOENA\_Pos, [472](#)
- ITM\_TCR\_SYNCENA\_Msk, [472](#)
- ITM\_TCR\_SYNCENA\_Pos, [472](#)
- ITM\_TCR\_TSENA\_Msk, [472](#)
- ITM\_TCR\_TSENA\_Pos, [472](#)
- ITM\_TCR\_TSPrescale\_Msk, [472](#)
- ITM\_TCR\_TSPrescale\_Pos, [473](#)
- ITM\_TCR\_TraceBusID\_Msk, [472](#)
- ITM\_TCR\_TraceBusID\_Pos, [472](#)
- ITM\_TPR\_PRIVMASK\_Msk, [473](#)
- ITM\_TPR\_PRIVMASK\_Pos, [473](#)
- IntDefaultHandler
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [807](#)
- IntStatus
  - LPC Public Types, [479](#)
- interrupts
  - I2S\_STATISTICS\_T, [564](#)
  - UART\_STATISTICS\_T, [649](#)
- itemSz
  - RINGBUFF\_T, [633](#)
- JTAGIDCODE
  - LPC\_SYSCON\_T, [611](#)
- Keil uVision support in LPCOpen, [474](#)
- L
  - \_WB\_T, [665](#)
  - LPC\_SCT\_T, [601](#)
- LAR
  - ITM\_Type, [568](#)
- LAST\_TIMER
  - timer\_5411x.c, [803](#)
- LIMIT\_H
  - LPC\_SCT\_T, [601](#)
- LIMIT\_L
  - LPC\_SCT\_T, [601](#)
- LOAD
  - SysTick\_Type, [643](#)
- LPC Public Macros, [475](#)
  - \_BIT, [475](#)
  - \_BITMASK, [475](#)
  - \_SBF, [475](#)
  - EXTERN, [475](#)
  - MAX, [475](#)
  - MIN, [475](#)
  - NELEMENTS, [475](#)
  - NULL, [475](#)
  - STATIC, [475](#)
- LPC Public Types, [477](#)
  - ALIGN, [477](#)
  - BLOCKING, [480](#)
  - BOOL\_16, [478](#)
  - BOOL\_32, [478](#)
  - BOOL\_8, [478](#)
  - Bool, [479](#)
  - CHAR, [478](#)
  - DISABLE, [480](#)
  - ENABLE, [480](#)
  - ERROR, [480](#)
  - FALSE, [479](#)
  - FlagStatus, [480](#)
  - FunctionalState, [480](#)
  - INLINE, [477](#)
  - INT\_16, [478](#)
  - INT\_32, [478](#)
  - INT\_64, [478](#)
  - INT\_8, [478](#)
  - IntStatus, [479](#)
  - NONE\_BLOCKING, [480](#)
  - PARAM\_FUNCTIONALSTATE, [477](#)
  - PARAM\_SETSTATE, [478](#)
  - PFI, [479](#)
  - PFV, [479](#)
  - RESET, [480](#)
  - SET, [480](#)
  - SUCCESS, [480](#)
  - SetState, [479](#)
  - Status, [480](#)
  - TRANSFER\_BLOCK\_T, [480](#)
  - TRUE, [479](#)
  - UNS\_16, [479](#)
  - UNS\_32, [479](#)
  - UNS\_64, [479](#)
  - UNS\_8, [479](#)
  - WEAK, [478](#)
- LPC5411X multi-core use in LPCOpen, [481](#)
- LPC5411X\_IRQn\_Type
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- LPC5411X\_M0\_IRQn\_Type
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
- LPC5411X\_ROMVER\_0
  - CHIP: LPC5411X Power LIBRARY functions, [251](#)
- LPC5411X\_ROMVER\_1
  - CHIP: LPC5411X Power LIBRARY functions, [251](#)
- LPC5411X\_ROMVER\_2
  - CHIP: LPC5411X Power LIBRARY functions, [251](#)
- LPC5411x Chip specific drivers, [483](#)
- LPC\_ADC
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [233](#)
- LPC\_ADC\_BASE

- CHIP: LPC5411X Peripheral addresses and register set declarations, [233](#)
- LPC\_ADC\_T, [570](#)
  - CALIBR, [571](#)
  - CHAN\_THRSEL, [571](#)
  - CTRL, [571](#)
  - DAT, [571](#)
  - FLAGS, [571](#)
  - INSEL, [571](#)
  - INTEN, [571](#)
  - RESERVED0, [571](#)
  - SEQ\_CTRL, [571](#)
  - SEQ\_GDAT, [572](#)
  - SEQA\_CTRL, [572](#)
  - SEQA\_GDAT, [572](#)
  - SEQB\_CTRL, [572](#)
  - SEQB\_GDAT, [572](#)
  - STARTUP, [572](#)
  - THR\_HIGH, [572](#)
  - THR\_LOW, [572](#)
- LPC\_ASSERT
  - lpc\_assert.h, [741](#)
- LPC\_ASYNC\_SYSCON
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [233](#)
- LPC\_ASYNC\_SYSCON\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [233](#)
- LPC\_ASYNC\_SYSCON\_T, [572](#)
  - ASYNCAPBCLKCTRL, [573](#)
  - ASYNCAPBCLKCTRLCLR, [573](#)
  - ASYNCAPBCLKCTRLSET, [573](#)
  - ASYNCAPBCLKSELA, [573](#)
  - ASYNCPRESETCTRLCLR, [573](#)
  - ASYNCPRESETCTRLSET, [573](#)
  - AYSNCPRESETCTRL, [573](#)
  - RESERVED0, [573](#)
  - RESERVED1, [574](#)
- LPC\_CRC
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [233](#)
- LPC\_CRC\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [233](#)
- LPC\_CRC\_T, [574](#)
  - MODE, [574](#)
  - SEED, [574](#)
  - SUM, [574](#)
  - WRDATA16, [574](#)
  - WRDATA32, [575](#)
  - WRDATA8, [575](#)
- LPC\_DMA
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [233](#)
- LPC\_DMA\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [233](#)
- LPC\_DMA\_CHANNEL\_T, [575](#)
- CFG, [575](#)
- CTLSTAT, [575](#)
- RESERVED, [575](#)
- XFERCFG, [575](#)
- LPC\_DMA\_COMMON\_T, [576](#)
  - ABORT, [576](#)
  - ACTIVE, [576](#)
  - BUSY, [577](#)
  - ENABLECLR, [577](#)
  - ENABLESET, [577](#)
  - ERRINT, [577](#)
  - INTA, [577](#)
  - INTB, [577](#)
  - INTENCLR, [577](#)
  - INTENSET, [577](#)
  - RESERVED0, [577](#)
  - RESERVED1, [578](#)
  - RESERVED10, [578](#)
  - RESERVED2, [578](#)
  - RESERVED3, [578](#)
  - RESERVED4, [578](#)
  - RESERVED5, [578](#)
  - RESERVED6, [578](#)
  - RESERVED7, [578](#)
  - RESERVED8, [578](#)
  - RESERVED9, [578](#)
  - SETTRIG, [578](#)
  - SETVALID, [578](#)
- LPC\_DMA\_T, [579](#)
  - CTRL, [579](#)
  - DMACH, [579](#)
  - DMACOMMON, [579](#)
  - INTSTAT, [579](#)
  - RESERVED0, [579](#)
  - RESERVED2, [580](#)
  - SRAMBASE, [580](#)
- LPC\_DMIC
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_DMIC\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_DMIC\_Channel\_Type, [580](#)
- LPC\_DMIC\_T, [580](#)
- LPC\_FLASHMEM\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_FLEXCOMM0\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_FLEXCOMM1\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_FLEXCOMM2\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_FLEXCOMM3\_BASE

- CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_FLEXCOMM4\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_FLEXCOMM5\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_FLEXCOMM6\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_FLEXCOMM7\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_FLEXCOMM\_T
  - CHIP: LPC5411X flexcomm API, [380](#)
- LPC\_FMC\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [234](#)
- LPC\_GINT
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_GPIO
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_GPIO\_GROUPINT0\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_GPIO\_GROUPINT1\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_GPIO\_PORT\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_GPIO\_T, [581](#)
  - B, [581](#)
  - CLR, [581](#)
  - DIR, [582](#)
  - MASK, [582](#)
  - MPIN, [582](#)
  - NOT, [582](#)
  - PIN, [582](#)
  - SET, [582](#)
  - W, [582](#)
- LPC\_GPIOGROUPINT\_T, [582](#)
  - CTRL, [583](#)
  - PORT\_ENA, [583](#)
  - PORT\_POL, [583](#)
  - RESERVED0, [583](#)
  - RESERVED1, [583](#)
- LPC\_I2C\_T
  - CHIP: LPC5411x I2C driver, [414](#)
- LPC\_I2S6
  - i2s\_5411x.h, [729](#)
- LPC\_I2S6\_BASE
  - i2s\_5411x.h, [729](#)
- LPC\_I2S7
  - i2s\_5411x.h, [729](#)
- LPC\_I2S7\_BASE
  - i2s\_5411x.h, [729](#)
- LPC\_I2S\_T, [583](#)
  - CFG1, [584](#)
  - CFG2, [584](#)
  - DIV, [584](#)
  - FIFOCFG, [584](#)
  - FIFOINTENCLR, [584](#)
  - FIFOINTENSET, [584](#)
  - FIFOINTSTAT, [585](#)
  - FIFORD, [585](#)
  - FIFORD48HNOPOP, [585](#)
  - FIFORDNOPOP, [585](#)
  - FIFOSTAT, [585](#)
  - FIFOTRIG, [585](#)
  - FIFOWR, [585](#)
  - PID, [585](#)
  - PSELID, [585](#)
  - RESERVED0, [586](#)
  - RESERVED00, [586](#)
  - RESERVED0A, [586](#)
  - RESERVED5, [586](#)
  - STAT, [586](#)
- LPC\_INMUX
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_INMUX\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_INMUX\_T, [586](#)
  - DMA\_ITRIG\_INMUX, [587](#)
  - DMA\_OTRIG\_INMUX, [587](#)
  - FREQMEAS\_REF, [587](#)
  - FREQMEAS\_TARGET, [587](#)
  - PINTSEL, [587](#)
  - RESERVED1, [587](#)
  - RESERVED2, [587](#)
  - RESERVED3, [587](#)
- LPC\_IOCON
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_IOCON\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_IOCON\_T, [588](#)
  - PIO, [588](#)
- LPC\_ISPAP\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_MBOX
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_MBOX\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [235](#)
- LPC\_MBOX\_T, [588](#)
  - BOX, [588](#)
  - MUTEX, [588](#)

- RESERVED1, [588](#)
- RESERVED2, [589](#)
- LPC\_MBOXIRQ\_T, [589](#)
  - IRQ, [589](#)
  - IRQCLR, [589](#)
  - IRQSET, [589](#)
  - RESERVED, [589](#)
- LPC\_MRT
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_MRT\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_MRT\_CH
  - CHIP: LPC5411X Multi-Rate Timer driver, [215](#)
- LPC\_MRT\_CH0
  - CHIP: LPC5411X Multi-Rate Timer driver, [215](#)
- LPC\_MRT\_CH1
  - CHIP: LPC5411X Multi-Rate Timer driver, [215](#)
- LPC\_MRT\_CH2
  - CHIP: LPC5411X Multi-Rate Timer driver, [215](#)
- LPC\_MRT\_CH3
  - CHIP: LPC5411X Multi-Rate Timer driver, [215](#)
- LPC\_MRT\_CH\_T, [590](#)
  - CTRL, [590](#)
  - INTVAL, [590](#)
  - STAT, [590](#)
  - TIMER, [590](#)
- LPC\_MRT\_T, [590](#)
  - CHANNEL, [591](#)
  - IDLE\_CH, [591](#)
  - IRQ\_FLAG, [591](#)
  - MODCFG, [591](#)
  - unused, [591](#)
- LPC\_OK
  - error.h, [706](#)
- LPC\_PIN\_INT\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_PIN\_INT\_T, [591](#)
  - CIENF, [592](#)
  - CIENR, [592](#)
  - FALL, [592](#)
  - IENF, [592](#)
  - IENR, [592](#)
  - ISEL, [592](#)
  - IST, [592](#)
  - PMCFG, [592](#)
  - PMCTRL, [593](#)
  - PMSRC, [593](#)
  - RISE, [593](#)
  - SIENF, [593](#)
  - SIENR, [593](#)
- LPC\_PININT
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_PMU
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_PMU\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_PMU\_T, [593](#)
  - BODCTRL, [594](#)
  - RESERVED0, [594](#)
- LPC\_ROM\_API
  - CHIP: LPC5411X ROM API declarations and functions, [257](#)
- LPC\_ROM\_API\_BASE\_LOC
  - CHIP: LPC5411X ROM API declarations and functions, [257](#)
- LPC\_ROM\_API\_T, [594](#)
  - reserved\_adcaltd, [594](#)
  - reserved\_clib, [594](#)
  - reserved\_div, [595](#)
  - reserved\_dmaaltd, [595](#)
  - reserved\_flexcomm, [595](#)
  - reserved\_i2cm, [595](#)
  - reserved\_i2cmon, [595](#)
  - reserved\_i2cs, [595](#)
  - reserved\_power, [595](#)
  - reserved\_spim, [595](#)
  - reserved\_spis, [595](#)
  - reserved\_uartalt, [596](#)
  - reserved\_usart, [596](#)
  - usbdApiBase, [596](#)
- LPC\_ROM\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_RTC
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_RTC\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_RTC\_T, [596](#)
  - COUNT, [596](#)
  - CTRL, [596](#)
  - MATCH, [597](#)
  - WAKE, [597](#)
- LPC\_SCT
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_SCT\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_SCT\_T, [597](#)
  - CAP, [599](#)
  - CAPCTRL, [599](#)
  - CLR, [599](#)
  - CONEN, [599](#)
  - CONFIG, [599](#)
  - CONFLAG, [599](#)
  - COUNT\_H, [599](#)
  - COUNT\_L, [599](#)

- COUNT\_U, [600](#)
- CTRL, [600](#)
- CTRL\_H, [600](#)
- CTRL\_L, [600](#)
- CTRL\_U, [600](#)
- DMA0REQUEST, [600](#)
- DMA1REQUEST, [600](#)
- EVEN, [600](#)
- EVENT, [600](#)
- EVFLAG, [601](#)
- H, [601](#)
- HALT\_H, [601](#)
- HALT\_L, [601](#)
- INPUT, [601](#)
- L, [601](#)
- LIMIT\_H, [601](#)
- LIMIT\_L, [601](#)
- MATCH, [601](#)
- MATCHREL, [601](#)
- MODULECONTENT, [602](#)
- OUT, [602](#)
- OUTPUT, [602](#)
- OUTPUTDIRCTRL, [602](#)
- REGMODE\_H, [602](#)
- REGMODE\_L, [602](#)
- RES, [602](#)
- RESERVED1, [602](#)
- RESERVED10, [602](#)
- RESERVED2, [602](#)
- RESERVED3, [602](#)
- RESERVED6, [603](#)
- RESERVED9, [603](#)
- SET, [603](#)
- START\_H, [603](#)
- START\_L, [603](#)
- STATE, [603](#)
- STATE\_H, [603](#)
- STATE\_L, [603](#)
- STOP\_H, [603](#)
- STOP\_L, [603](#)
- U, [604](#)
- LPC\_SPI\_T, [604](#)
  - CFG, [604](#)
  - DIV, [604](#)
  - DLY, [605](#)
  - FIFOCFG, [605](#)
  - FIFOINTENCLR, [605](#)
  - FIFOINTENSET, [605](#)
  - FIFOINTSTAT, [605](#)
  - FIFORD, [605](#)
  - FIFORDNOPOP, [605](#)
  - FIFOSTAT, [605](#)
  - FIFOTRIG, [605](#)
  - FIFOWR, [606](#)
  - INTENCLR, [606](#)
  - INTENSET, [606](#)
  - INTSTAT, [606](#)
  - PID, [606](#)
  - PSELID, [606](#)
  - RESERVED0, [606](#)
  - RESERVED1, [606](#)
  - STAT, [606](#)
- LPC\_SPIFI\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [236](#)
- LPC\_SRAM0\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_SRAM1\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_SRAM2\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_SRAMX\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_SYSCON
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_SYSCON\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_SYSCON\_T, [607](#)
  - ADCCLKDIV, [609](#)
  - ADCCLKSEL, [609](#)
  - AHBCLKCTRL, [609](#)
  - AHBCLKCTRLCLR, [609](#)
  - AHBCLKCTRLSET, [609](#)
  - AHBCLKDIV, [609](#)
  - AHBMATPRIO, [609](#)
  - ASYNCAPBCTRL, [609](#)
  - AUTOCGOR, [610](#)
  - CLKOUTDIV, [610](#)
  - CLKOUTSELA, [610](#)
  - CPBOOT, [610](#)
  - CPCTRL, [610](#)
  - CPSTACK, [610](#)
  - CPSTAT, [610](#)
  - DEVICE\_ID, [610](#)
  - DMICCLKDIV, [610](#)
  - DMICCLKSEL, [611](#)
  - FLASHCFG, [611](#)
  - FREQMECTRL, [611](#)
  - FRGCLKSEL, [611](#)
  - FRGCTRL, [611](#)
  - FROCTRL, [611](#)
  - FXCOMCLKSEL, [611](#)
  - JTAGIDCODE, [611](#)
  - MAINCLKSELA, [611](#)
  - MAINCLKSELB, [612](#)
  - MCLKCLKSEL, [612](#)
  - MCLKDIV, [612](#)
  - MCLKIO, [612](#)
  - NMISRC, [612](#)
  - PDRUNCFG, [612](#)

- PDRUNCFGCLR, [612](#)
- PDRUNCFGSET, [612](#)
- PIOPORCAP, [612](#)
- PIORESCAP, [613](#)
- PRESETCTRL, [613](#)
- PRESETCTRLCLR, [613](#)
- PRESETCTRLSET, [613](#)
- RESERVED0, [613](#)
- RESERVED1, [613](#)
- RESERVED10, [613](#)
- RESERVED11, [613](#)
- RESERVED12, [613](#)
- RESERVED13, [613](#)
- RESERVED14, [613](#)
- RESERVED15, [614](#)
- RESERVED16, [614](#)
- RESERVED16A, [614](#)
- RESERVED17, [614](#)
- RESERVED18, [614](#)
- RESERVED19, [614](#)
- RESERVED2, [614](#)
- RESERVED20, [614](#)
- RESERVED21, [614](#)
- RESERVED22, [614](#)
- RESERVED23, [614](#)
- RESERVED24, [614](#)
- RESERVED25, [615](#)
- RESERVED26, [615](#)
- RESERVED27, [615](#)
- RESERVED28, [615](#)
- RESERVED29, [615](#)
- RESERVED3, [615](#)
- RESERVED30, [615](#)
- RESERVED31, [615](#)
- RESERVED32, [615](#)
- RESERVED33, [615](#)
- RESERVED34, [615](#)
- RESERVED35, [615](#)
- RESERVED36, [616](#)
- RESERVED37, [616](#)
- RESERVED4, [616](#)
- RESERVED5, [616](#)
- RESERVED6, [616](#)
- RESERVED7, [616](#)
- RESERVED8, [616](#)
- RESERVED9, [616](#)
- RTCOSCCTRL, [616](#)
- SPIFCLKDIV, [616](#)
- SPIFCLKSEL, [616](#)
- STARTERP, [617](#)
- STARTERPCLR, [617](#)
- STARTERPSET, [617](#)
- SYSMEMREMAP, [617](#)
- SYSPLLCLKSEL, [617](#)
- SYSPLLCTRL, [617](#)
- SYSPLLNDEC, [617](#)
- SYSPLLDEC, [617](#)
- SYSPLLSSCTRL, [617](#)
- SYSPLLSTAT, [618](#)
- SYSRSTSTAT, [618](#)
- SYSTCKCAL, [618](#)
- SYSTICKCLKDIV, [618](#)
- USBCLKCTRL, [618](#)
- USBCLKDIV, [618](#)
- USBCLKSEL, [618](#)
- USBCLKSTAT, [618](#)
- WDTOSCCTRL, [618](#)
- LPC\_TIMER0
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_TIMER0\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_TIMER1
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_TIMER1\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_TIMER2
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_TIMER2\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [237](#)
- LPC\_TIMER3
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_TIMER32\_1
  - stopwatch\_5411x.c, [801](#)
- LPC\_TIMER3\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_TIMER4
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_TIMER4\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_TIMER\_T, [619](#)
  - CCR, [619](#)
  - CR, [619](#)
  - CTCR, [619](#)
  - EMR, [619](#)
  - IR, [620](#)
  - MCR, [620](#)
  - MR, [620](#)
  - PC, [620](#)
  - PR, [620](#)
  - PWMC, [620](#)
  - RESERVED0, [620](#)
  - TC, [620](#)
  - TCR, [620](#)
- LPC\_USART\_T, [621](#)
  - ADDR, [621](#)
  - BRG, [621](#)



- CFG, [621](#)
- CTL, [622](#)
- FIFOCFG, [622](#)
- FIFOINTENCLR, [622](#)
- FIFOINTENSET, [622](#)
- FIFOINTSTAT, [622](#)
- FIFORD, [622](#)
- FIFORDNOPOP, [622](#)
- FIFOSTAT, [622](#)
- FIFOTRIG, [622](#)
- FIFOWR, [623](#)
- INTENCLR, [623](#)
- INTENSET, [623](#)
- INTSTAT, [623](#)
- OSR, [623](#)
- PID, [623](#)
- PSELID, [623](#)
- STAT, [623](#)
- LPC\_USB
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_USB\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_UTICK
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_UTICK\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_UTICK\_T, [624](#)
  - CTRL, [624](#)
  - STATUS, [624](#)
- LPC\_WWDT
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_WWDT\_BASE
  - CHIP: LPC5411X Peripheral addresses and register set declarations, [238](#)
- LPC\_WWDT\_T, [624](#)
  - FEED, [625](#)
  - MOD, [625](#)
  - RESERVED0, [625](#)
  - TC, [625](#)
  - TV, [625](#)
  - WARNINT, [625](#)
  - WINDOW, [625](#)
- LPCOpen download and installation information, [484](#)
- LPCOpen versioning and release history, [485](#)
- LSR
  - ITM\_Type, [568](#)
- LSUCNT
  - DWT\_Type, [558](#)
- LeftJust
  - I2S\_AUDIO\_FORMAT\_T, [563](#)
- lpc\_assert.h
  - LPC\_ASSERT, [741](#)
- lsbFirst
  - SPI\_CFGSETUP\_T, [638](#)
- lvl\_rx
  - I2S\_STATISTICS\_T, [565](#)
  - UART\_STATISTICS\_T, [649](#)
- lvl\_tx
  - I2S\_STATISTICS\_T, [565](#)
  - UART\_STATISTICS\_T, [649](#)
- MAILBOX\_AVAIL
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [205](#)
- MAILBOX\_CM0PLUS
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [205](#)
- MAILBOX\_CM4
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [205](#)
- MAILBOX\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- MAINCLKSELA
  - LPC\_SYSCON\_T, [611](#)
- MAINCLKSELB
  - LPC\_SYSCON\_T, [612](#)
- MASK
  - LPC\_GPIO\_T, [582](#)
- MASK0
  - DWT\_Type, [558](#)
- MASK1
  - DWT\_Type, [558](#)
- MASK2
  - DWT\_Type, [558](#)
- MASK3
  - DWT\_Type, [559](#)
- MATCH
  - LPC\_RTC\_T, [597](#)
  - LPC\_SCT\_T, [601](#)
- MATCHREL
  - LPC\_SCT\_T, [601](#)
- MAX
  - LPC Public Macros, [475](#)
- MAX\_DMA\_CHANNEL
  - CHIP: LPC5411X DMA Engine driver (legacy), [127](#)
- MBOX\_IDX\_T
  - CHIP: LPC5411X Mailbox M4/M0+ driver, [205](#)
- MCLKCLKSEL
  - LPC\_SYSCON\_T, [612](#)
- MCLKDIV
  - LPC\_SYSCON\_T, [612](#)
- MCLKIO
  - LPC\_SYSCON\_T, [612](#)
- MCR
  - LPC\_TIMER\_T, [620](#)
- MIN
  - LPC Public Macros, [475](#)
- MINSPEURHOUR
  - rtc\_ut.c, [797](#)
- MMFAR
  - SCB\_Type, [636](#)
- MMFR



- SCB\_Type, [636](#)
- MOD
  - LPC\_WWDT\_T, [625](#)
- MODCFG
  - LPC\_MRT\_T, [591](#)
- MODE
  - LPC\_CRC\_T, [574](#)
- MODE\_CFG\_CCITT
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- MODE\_CFG\_CRC16
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- MODE\_CFG\_CRC32
  - CHIP: LPC5411X Cyclic Redundancy Check Engine driver, [99](#)
- MODULECONTENT
  - LPC\_SCT\_T, [602](#)
- MONETHSPERYEAR
  - rtc\_ut.c, [797](#)
- MONRXDAT
  - CHIP: LPC5411x I2C driver, [414](#)
- MPIN
  - LPC\_GPIO\_T, [582](#)
- MR
  - LPC\_TIMER\_T, [620](#)
- MRT0\_INTFLAG
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT1\_INTFLAG
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT2\_INTFLAG
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT3\_INTFLAG
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT\_CHANNELS\_NUM
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT\_CTRL\_INTEN\_MASK
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT\_CTRL\_MODE\_MASK
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT\_INTVAL\_IVALUE
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT\_INTVAL\_LOAD
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- MRT\_MODE\_ONESHOT
  - CHIP: LPC5411X Multi-Rate Timer driver, [217](#)
- MRT\_MODE\_REPEAT
  - CHIP: LPC5411X Multi-Rate Timer driver, [217](#)
- MRT\_MODE\_T
  - CHIP: LPC5411X Multi-Rate Timer driver, [217](#)
- MRT\_NO\_IDLE\_CHANNEL
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRT\_STAT\_INTFLAG
  - CHIP: LPC5411X Multi-Rate Timer driver, [216](#)
- MRTn\_INTFLAG
  - CHIP: LPC5411X Multi-Rate Timer driver, [217](#)
- MSCfg
  - I2S\_AUDIO\_FORMAT\_T, [563](#)
- MSTCTL
  - CHIP: LPC5411x I2C driver, [414](#)
- MSTDAT
  - CHIP: LPC5411x I2C driver, [414](#)
- MSTTIME
  - CHIP: LPC5411x I2C driver, [415](#)
- MUTEX
  - LPC\_MBOX\_T, [588](#)
- MVALMAX
  - pll\_5411x.c, [786](#)
- marker
  - PINTABLE\_T, [629](#)
- master
  - SPI\_CFGSETUP\_T, [638](#)
- mclk\_in\_rate
  - clock\_5411x.c, [777](#)
- MemManage\_Handler
  - cr\_startup\_lpc5411x.c, [807](#)
- MemoryManagement\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- mfDither
  - PLL\_CONFIG\_T, [630](#)
- Mode
  - I2S\_AUDIO\_FORMAT\_T, [563](#)
- mode
  - SPI\_CFGSETUP\_T, [638](#)
- modefunc
  - PINMUX\_GRP\_T, [628](#)
- mul
  - UART\_BAUD\_T, [648](#)
- N
  - APSR\_Type, [550](#)
  - xPSR\_Type, [668](#)
- NELEMENTS
  - LPC Public Macros, [475](#)
- NMI\_Handler
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [807](#)
- NMISRC
  - LPC\_SYSCON\_T, [612](#)
- NONE\_BLOCKING
  - LPC Public Types, [480](#)
- NORMAL\_MASTER
  - i2s\_5411x.h, [730](#)
- NORMAL\_SLAVE
  - i2s\_5411x.h, [730](#)
- NOT
  - LPC\_GPIO\_T, [582](#)
- nPRIV
  - CONTROL\_Type, [552](#)

- NULL
  - LPC Public Macros, [475](#)
- NVALMAX
  - pll\_5411x.c, [786](#)
- NVIC
  - Core Definitions, [452](#)
- NVIC Functions, [486](#)
  - \_BIT\_SHIFT, [486](#)
  - \_IP\_IDX, [486](#)
  - \_SHP\_IDX, [487](#)
  - NVIC\_ClearPendingIRQ, [487](#)
  - NVIC\_DecodePriority, [487](#)
  - NVIC\_DisableIRQ, [487](#)
  - NVIC\_EnableIRQ, [487](#)
  - NVIC\_EncodePriority, [488](#)
  - NVIC\_GetActive, [488](#)
  - NVIC\_GetPendingIRQ, [488](#)
  - NVIC\_GetPriority, [488](#)
  - NVIC\_GetPriorityGrouping, [489](#)
  - NVIC\_SetPendingIRQ, [489](#)
  - NVIC\_SetPriority, [489](#)
  - NVIC\_SetPriorityGrouping, [489](#)
  - NVIC\_SystemReset, [490](#)
- NVIC\_BASE
  - Core Definitions, [452](#)
- NVIC\_ClearPendingIRQ
  - NVIC Functions, [487](#)
- NVIC\_DecodePriority
  - NVIC Functions, [487](#)
- NVIC\_DisableIRQ
  - NVIC Functions, [487](#)
- NVIC\_EnableIRQ
  - NVIC Functions, [487](#)
- NVIC\_EncodePriority
  - NVIC Functions, [488](#)
- NVIC\_GetActive
  - NVIC Functions, [488](#)
- NVIC\_GetPendingIRQ
  - NVIC Functions, [488](#)
- NVIC\_GetPriority
  - NVIC Functions, [488](#)
- NVIC\_GetPriorityGrouping
  - NVIC Functions, [489](#)
- NVIC\_STIR\_INTID\_Msk
  - Nested Vectored Interrupt Controller (NVIC), [492](#)
- NVIC\_STIR\_INTID\_Pos
  - Nested Vectored Interrupt Controller (NVIC), [492](#)
- NVIC\_SetPendingIRQ
  - NVIC Functions, [489](#)
- NVIC\_SetPriority
  - NVIC Functions, [489](#)
- NVIC\_SetPriorityGrouping
  - NVIC Functions, [489](#)
- NVIC\_SystemReset
  - NVIC Functions, [490](#)
- NVIC\_Type, [625](#)
  - IABR, [626](#)
  - ICER, [626](#)
  - ICPR, [626](#)
  - IP, [626](#)
  - ISER, [626](#)
  - ISPR, [626](#)
  - RESERVED0, [627](#)
  - RESERVED2, [627](#)
  - RESERVED3, [627](#)
  - RESERVED4, [627](#)
  - RESERVED5, [627](#)
  - RSERVED1, [627](#)
  - STIR, [627](#)
- NXP LPCXpresso LPC54114 LQFP board, [491](#)
- Nested Vectored Interrupt Controller (NVIC), [492](#)
  - NVIC\_STIR\_INTID\_Msk, [492](#)
  - NVIC\_STIR\_INTID\_Pos, [492](#)
- next
  - DMA\_CHDESC\_T, [554](#)
- NonMaskableInt\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- OP\_MODE\_T
  - CHIP: LPC5411X DMIC driver, [137](#)
- OSR
  - CHIP: LPC5411X DMIC driver, [145](#)
  - LPC\_USART\_T, [623](#)
- OUT
  - LPC\_SCT\_T, [602](#)
- OUTPUT
  - LPC\_SCT\_T, [602](#)
- OUTPUTDIRCTRL
  - LPC\_SCT\_T, [602](#)
- offsetof
  - error.h, [705](#)
- option
  - SPIM\_XFER\_T, [640](#)
- osr
  - CHIP: LPC5411X DMIC driver, [146](#)
- ovr
  - UART\_BAUD\_T, [648](#)
- PARAM\_FUNCTIONALSTATE
  - LPC Public Types, [477](#)
- PARAM\_SETSTATE
  - LPC Public Types, [478](#)
- PC
  - LPC\_TIMER\_T, [620](#)
- PCSR
  - DWT\_Type, [559](#)
- PDM\_DIV\_T
  - CHIP: LPC5411X DMIC driver, [138](#)
- PDMData
  - I2S\_AUDIO\_FORMAT\_T, [563](#)
- PDRUNCFG
  - LPC\_SYSCON\_T, [612](#)
- PDRUNCFGCLR
  - LPC\_SYSCON\_T, [612](#)

- PDRUNCFGSET
  - LPC\_SYSCON\_T, [612](#)
- PENDING\_SPIFI\_LITE
  - error.h, [708](#)
- PFI
  - LPC Public Types, [479](#)
- PFR
  - SCB\_Type, [636](#)
- PFV
  - LPC Public Types, [479](#)
- PHY\_CTRL
  - CHIP: LPC5411X DMIC driver, [146](#)
- PID
  - CHIP: LPC5411x I2C driver, [415](#)
  - LPC\_I2S\_T, [585](#)
  - LPC\_SPI\_T, [606](#)
  - LPC\_USART\_T, [623](#)
- PID0
  - ITM\_Type, [568](#)
- PID1
  - ITM\_Type, [568](#)
- PID2
  - ITM\_Type, [568](#)
- PID3
  - ITM\_Type, [568](#)
- PID4
  - ITM\_Type, [568](#)
- PID5
  - ITM\_Type, [568](#)
- PID6
  - ITM\_Type, [568](#)
- PID7
  - ITM\_Type, [569](#)
- PIN
  - LPC\_GPIO\_T, [582](#)
- PIN\_INT0\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- PIN\_INT1\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- PIN\_INT2\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- PIN\_INT3\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- PIN\_INT4\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- PIN\_INT5\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- PIN\_INT6\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- PIN\_INT7\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- PININT\_ISEL\_PMODE\_MASK
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [240](#)
- PININT\_PATTERNCONST0
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININT\_PATTERNCONST1
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININT\_PATTERNEVENT
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININT\_PATTERNFALLING
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININT\_PATTERNHIGH
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININT\_PATTERNLOW
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININT\_PATTERNRISING
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININT\_PATTERNRISINGORFALLING
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININT\_PMCTRL\_MASK
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [240](#)
- PININT\_PMCTRL\_PMATCH\_SEL
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININT\_PMCTRL\_RXEV\_ENA
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININT\_SRC\_BITCFG\_MASK
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININT\_SRC\_BITCFG\_START
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININT\_SRC\_BITSOURCE\_MASK
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININT\_SRC\_BITSOURCE\_START
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININTBITSlice0

- CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTBITSlice1
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTBITSlice2
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTBITSlice3
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTBITSlice4
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTBITSlice5
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTBITSlice6
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTBITSlice7
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTCH
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININTCH0
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININTCH1
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININTCH2
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININTCH3
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININTCH4
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [241](#)
- PININTCH5
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTCH6
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTCH7
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [242](#)
- PININTSELECT0
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- PININTSELECT1
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- PININTSELECT2
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- PININTSELECT3
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- PININTSELECT4
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- PININTSELECT5
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- PININTSELECT6
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- PININTSELECT7
  - CHIP: LPC5411X Pin Interrupt and Pattern Match driver, [243](#)
- PINMUX\_GRP\_T, [627](#)
  - modefunc, [628](#)
  - pin, [628](#)
  - port, [628](#)
- PINTABLE\_T, [628](#)
  - crc32\_len, [628](#)
  - crc32\_val, [628](#)
  - hostIrqPortPin, [629](#)
  - hostMisoPortPin, [629](#)
  - hostMosiPortPin, [629](#)
  - hostSckPortPin, [629](#)
  - hostSselPortPin, [629](#)
  - ifSel, [629](#)
  - img\_type, [629](#)
  - marker, [629](#)
  - version, [629](#)
  - xorVal, [629](#)
- PINTSEL
  - LPC\_INMUX\_T, [587](#)
- PIO
  - LPC\_IOCON\_T, [588](#)
- PIOPORCAP
  - LPC\_SYSCON\_T, [612](#)
- PIORESCAP
  - LPC\_SYSCON\_T, [613](#)
- PLL0\_SSCG\_DITHER\_VALUE
  - pll\_5411x.c, [786](#)
- PLL0\_SSCG\_MC\_COMP\_VALUE
  - pll\_5411x.c, [786](#)
- PLL0\_SSCG\_MF\_FREQ\_VALUE
  - pll\_5411x.c, [786](#)
- PLL0\_SSCG\_MR\_DEPTH\_VALUE
  - pll\_5411x.c, [786](#)
- PLL\_CONFIG\_T, [630](#)
  - desiredRate, [630](#)
  - flags, [630](#)
  - InputRate, [630](#)
  - mfDither, [630](#)
  - ss\_mc, [630](#)
  - ss\_mf, [630](#)
  - ss\_mr, [631](#)

- PLL\_CONFIGFLAG\_FORCENOFRACT  
CHIP: LPC5411X PLL Driver, [226](#)
- PLL\_CONFIGFLAG\_USEINRATE  
CHIP: LPC5411X PLL Driver, [226](#)
- PLL\_CTRL\_BANDSEL\_SSCGREG\_N  
pll\_5411x.c, [786](#)
- PLL\_CTRL\_BANDSEL\_SSCGREG\_N\_P  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_BYPASS  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_BYPASS\_FBDIV2  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_BYPASS\_FBDIV2\_P  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_BYPASS\_P  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_DIRECTI  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_DIRECTI\_P  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_DIRECTO  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_DIRECTO\_P  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_UPLIMOFF  
pll\_5411x.c, [787](#)
- PLL\_CTRL\_UPLIMOFF\_P  
pll\_5411x.c, [787](#)
- PLL\_ERROR\_INPUT\_TOO\_HIGH  
CHIP: LPC5411X PLL Driver, [227](#)
- PLL\_ERROR\_INPUT\_TOO\_LOW  
CHIP: LPC5411X PLL Driver, [227](#)
- PLL\_ERROR\_OUTPUT\_TOO\_HIGH  
CHIP: LPC5411X PLL Driver, [227](#)
- PLL\_ERROR\_OUTPUT\_TOO\_LOW  
CHIP: LPC5411X PLL Driver, [227](#)
- PLL\_ERROR\_OUTSIDE\_INTLIMIT  
CHIP: LPC5411X PLL Driver, [227](#)
- PLL\_ERROR\_SUCCESS  
CHIP: LPC5411X PLL Driver, [227](#)
- PLL\_ERROR\_T  
CHIP: LPC5411X PLL Driver, [227](#)
- PLL\_LOWER\_IN\_LIMIT  
pll\_5411x.c, [787](#)
- PLL\_MAX\_CCO\_FREQ\_MHZ  
pll\_5411x.c, [788](#)
- PLL\_MAX\_IN\_SSMODE  
pll\_5411x.c, [788](#)
- PLL\_MAX\_N\_DIV  
pll\_5411x.c, [788](#)
- PLL\_MIN\_CCO\_FREQ\_MHZ  
pll\_5411x.c, [788](#)
- PLL\_MIN\_IN\_SSMODE  
pll\_5411x.c, [788](#)
- PLL\_NDEC\_NREQ  
pll\_5411x.c, [788](#)
- PLL\_NDEC\_NREQ\_P  
pll\_5411x.c, [788](#)
- PLL\_NDEC\_VAL\_M  
pll\_5411x.c, [788](#)
- PLL\_NDEC\_VAL\_P  
pll\_5411x.c, [788](#)
- PLL\_NDEC\_VAL\_SET  
pll\_5411x.c, [788](#)
- PLL\_PDEC\_PREQ  
pll\_5411x.c, [788](#)
- PLL\_PDEC\_PREQ\_P  
pll\_5411x.c, [788](#)
- PLL\_PDEC\_VAL\_M  
pll\_5411x.c, [789](#)
- PLL\_PDEC\_VAL\_P  
pll\_5411x.c, [789](#)
- PLL\_PDEC\_VAL\_SET  
pll\_5411x.c, [789](#)
- PLL\_SETUP\_T, [631](#)  
flags, [631](#)  
pllRate, [631](#)  
SYSPLLCTRL, [631](#)  
SYSPLLNDEC, [632](#)  
SYSPLLDEC, [632](#)  
SYSPLLSSCTRL, [632](#)
- PLL\_SETUPFLAG\_ADGVOLT  
CHIP: LPC5411X PLL Driver, [226](#)
- PLL\_SETUPFLAG\_POWERUP  
CHIP: LPC5411X PLL Driver, [226](#)
- PLL\_SETUPFLAG\_WAITLOCK  
CHIP: LPC5411X PLL Driver, [226](#)
- PLL\_SSCG0\_MDEC\_VAL\_M  
pll\_5411x.c, [789](#)
- PLL\_SSCG0\_MDEC\_VAL\_P  
pll\_5411x.c, [789](#)
- PLL\_SSCG0\_MDEC\_VAL\_SET  
pll\_5411x.c, [789](#)
- PLL\_SSCG0\_MREQ  
pll\_5411x.c, [789](#)
- PLL\_SSCG0\_MREQ\_P  
pll\_5411x.c, [789](#)
- PLL\_SSCG0\_SEL\_EXT\_SSCG\_N  
pll\_5411x.c, [789](#)
- PLL\_SSCG0\_SEL\_EXT\_SSCG\_N\_P  
pll\_5411x.c, [790](#)
- PLL\_SSCG1\_DITHER  
pll\_5411x.c, [790](#)
- PLL\_SSCG1\_DITHER\_P  
pll\_5411x.c, [790](#)
- PLL\_SSCG1\_MC\_M  
pll\_5411x.c, [790](#)
- PLL\_SSCG1\_MC\_P  
pll\_5411x.c, [790](#)
- PLL\_SSCG1\_MD\_FRACT\_M  
pll\_5411x.c, [790](#)
- PLL\_SSCG1\_MD\_FRACT\_P  
pll\_5411x.c, [790](#)
- PLL\_SSCG1\_MD\_FRACT\_SET  
pll\_5411x.c, [790](#)
- PLL\_SSCG1\_MD\_INT\_M

- pll\_5411x.c, [790](#)
- PLL\_SSCG1\_MD\_INT\_P
  - pll\_5411x.c, [790](#)
- PLL\_SSCG1\_MD\_INT\_SET
  - pll\_5411x.c, [791](#)
- PLL\_SSCG1\_MD\_REQ
  - pll\_5411x.c, [791](#)
- PLL\_SSCG1\_MD\_REQ\_P
  - pll\_5411x.c, [791](#)
- PLL\_SSCG1\_MF\_M
  - pll\_5411x.c, [791](#)
- PLL\_SSCG1\_MF\_P
  - pll\_5411x.c, [791](#)
- PLL\_SSCG1\_MOD\_PD\_SSCGCLK\_N
  - pll\_5411x.c, [791](#)
- PLL\_SSCG1\_MOD\_PD\_SSCGCLK\_N\_P
  - pll\_5411x.c, [791](#)
- PLL\_SSCG1\_MR\_M
  - pll\_5411x.c, [791](#)
- PLL\_SSCG1\_MR\_P
  - pll\_5411x.c, [791](#)
- PLL\_SSCG\_DITHER\_VALUE
  - pll\_5411x.c, [791](#)
- PLL\_SSCG\_MC\_COMP\_VALUE
  - pll\_5411x.c, [791](#)
- PLL\_SSCG\_MF\_FREQ\_VALUE
  - pll\_5411x.c, [792](#)
- PLL\_SSCG\_MR\_DEPTH\_VALUE
  - pll\_5411x.c, [792](#)
- PMCFG
  - LPC\_PIN\_INT\_T, [592](#)
- PMCTRL
  - LPC\_PIN\_INT\_T, [593](#)
- PMSRC
  - LPC\_PIN\_INT\_T, [593](#)
- PMU\_BOD\_INT
  - CHIP: LPC5411X Power Management declarations and functions, [254](#)
- PMU\_BOD\_RST
  - CHIP: LPC5411X Power Management declarations and functions, [254](#)
- PMU\_BODINTVAL\_2\_05v
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODINTVAL\_2\_45v
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODINTVAL\_2\_75v
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODINTVAL\_3\_05v
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODINTVAL\_LVL0
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODINTVAL\_LVL1
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODINTVAL\_LVL2
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODINTVAL\_LVL3
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODRSTLVL\_0
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODRSTLVL\_1
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODRSTLVL\_1\_50V
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODRSTLVL\_1\_85V
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODRSTLVL\_2
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODRSTLVL\_2\_00V
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODRSTLVL\_2\_30V
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PMU\_BODRSTLVL\_3
  - CHIP: LPC5411X Power Management declarations and functions, [255](#)
- PORT
  - ITM\_Type, [569](#)
- PORT\_ENA
  - LPC\_GPIOGROUPINT\_T, [583](#)
- PORT\_POL
  - LPC\_GPIOGROUPINT\_T, [583](#)
- POWER\_DEEP\_POWER\_DOWN
  - CHIP: LPC5411X Power LIBRARY functions, [251](#)
- POWER\_DEEP\_SLEEP
  - CHIP: LPC5411X Power LIBRARY functions, [251](#)
- POWER\_MODE\_T
  - CHIP: LPC5411X Power LIBRARY functions, [251](#)
- POWER\_SLEEP
  - CHIP: LPC5411X Power LIBRARY functions, [251](#)
- PR
  - LPC\_TIMER\_T, [620](#)
- PREAC2FSCOEF
  - CHIP: LPC5411X DMIC driver, [146](#)
- PREAC4FSCOEF
  - CHIP: LPC5411X DMIC driver, [146](#)
- PRESETCTRL
  - LPC\_SYSCON\_T, [613](#)
- PRESETCTRLCLR
  - LPC\_SYSCON\_T, [613](#)
- PRESETCTRLSET
  - LPC\_SYSCON\_T, [613](#)

- PSELID
  - CHIP: LPC5411x I2C driver, [415](#)
  - LPC\_I2S\_T, [585](#)
  - LPC\_SPI\_T, [606](#)
  - LPC\_USART\_T, [623](#)
- PVALMAX
  - pll\_5411x.c, [792](#)
- PWMC
  - LPC\_TIMER\_T, [620](#)
- PWR\_ERROR\_CLOCK\_FREQ\_TOO\_HIGH
  - error.h, [709](#)
- PWR\_ERROR\_ILLEGAL\_MODE
  - error.h, [709](#)
- PWR\_ERROR\_INVALID\_CFG
  - error.h, [709](#)
- PWR\_ERROR\_INVALID\_STATE
  - error.h, [709](#)
- PWR\_ERROR\_PVT\_DETECT
  - error.h, [709](#)
- packing.h
  - ALIGNED, [745](#)
- pdm\_bypass
  - CHIP: LPC5411X DMIC driver, [137](#)
- pdm\_bypass\_clk0
  - CHIP: LPC5411X DMIC driver, [137](#)
- pdm\_bypass\_clk1
  - CHIP: LPC5411X DMIC driver, [137](#)
- pdm\_dual
  - CHIP: LPC5411X DMIC driver, [137](#)
- pdm\_stereo
  - CHIP: LPC5411X DMIC driver, [137](#)
- PendSV\_Handler
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [807](#)
- PendSV\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- pin
  - PINMUX\_GRP\_T, [628](#)
- pll\_5411x.c
  - Chip\_Clock\_GetPllConfig, [795](#)
  - Chip\_Clock\_GetSystemPLLOutFromSetupUpdate, [795](#)
  - curPllRate, [796](#)
  - FindGreatestCommonDivisor, [795](#)
  - findPllMMult, [795](#)
  - findPllPostDiv, [795](#)
  - findPllPreDiv, [795](#)
  - MVALMAX, [786](#)
  - NVALMAX, [786](#)
  - PLL0\_SSCG\_DITHER\_VALUE, [786](#)
  - PLL0\_SSCG\_MC\_COMP\_VALUE, [786](#)
  - PLL0\_SSCG\_MF\_FREQ\_VALUE, [786](#)
  - PLL0\_SSCG\_MR\_DEPTH\_VALUE, [786](#)
  - PLL\_CTRL\_BANDSEL\_SSCGREG\_N, [786](#)
  - PLL\_CTRL\_BANDSEL\_SSCGREG\_N\_P, [787](#)
  - PLL\_CTRL\_BYPASS, [787](#)
  - PLL\_CTRL\_BYPASS\_FBDIV2, [787](#)
  - PLL\_CTRL\_BYPASS\_FBDIV2\_P, [787](#)
  - PLL\_CTRL\_BYPASS\_P, [787](#)
  - PLL\_CTRL\_DIRECTI, [787](#)
  - PLL\_CTRL\_DIRECTI\_P, [787](#)
  - PLL\_CTRL\_DIRECTO, [787](#)
  - PLL\_CTRL\_DIRECTO\_P, [787](#)
  - PLL\_CTRL\_UPLIMOFF, [787](#)
  - PLL\_CTRL\_UPLIMOFF\_P, [787](#)
  - PLL\_LOWER\_IN\_LIMIT, [787](#)
  - PLL\_MAX\_CCO\_FREQ\_MHZ, [788](#)
  - PLL\_MAX\_IN\_SSMODE, [788](#)
  - PLL\_MAX\_N\_DIV, [788](#)
  - PLL\_MIN\_CCO\_FREQ\_MHZ, [788](#)
  - PLL\_MIN\_IN\_SSMODE, [788](#)
  - PLL\_NDEC\_NREQ, [788](#)
  - PLL\_NDEC\_NREQ\_P, [788](#)
  - PLL\_NDEC\_VAL\_M, [788](#)
  - PLL\_NDEC\_VAL\_P, [788](#)
  - PLL\_NDEC\_VAL\_SET, [788](#)
  - PLL\_PDEC\_PREQ, [788](#)
  - PLL\_PDEC\_PREQ\_P, [788](#)
  - PLL\_PDEC\_VAL\_M, [789](#)
  - PLL\_PDEC\_VAL\_P, [789](#)
  - PLL\_PDEC\_VAL\_SET, [789](#)
  - PLL\_SSCG0\_MDEC\_VAL\_M, [789](#)
  - PLL\_SSCG0\_MDEC\_VAL\_P, [789](#)
  - PLL\_SSCG0\_MDEC\_VAL\_SET, [789](#)
  - PLL\_SSCG0\_MREQ, [789](#)
  - PLL\_SSCG0\_MREQ\_P, [789](#)
  - PLL\_SSCG0\_SEL\_EXT\_SSCG\_N, [789](#)
  - PLL\_SSCG0\_SEL\_EXT\_SSCG\_N\_P, [790](#)
  - PLL\_SSCG1\_DITHER, [790](#)
  - PLL\_SSCG1\_DITHER\_P, [790](#)
  - PLL\_SSCG1\_MC\_M, [790](#)
  - PLL\_SSCG1\_MC\_P, [790](#)
  - PLL\_SSCG1\_MD\_FRACT\_M, [790](#)
  - PLL\_SSCG1\_MD\_FRACT\_P, [790](#)
  - PLL\_SSCG1\_MD\_FRACT\_SET, [790](#)
  - PLL\_SSCG1\_MD\_INT\_M, [790](#)
  - PLL\_SSCG1\_MD\_INT\_P, [790](#)
  - PLL\_SSCG1\_MD\_INT\_SET, [791](#)
  - PLL\_SSCG1\_MD\_REQ, [791](#)
  - PLL\_SSCG1\_MD\_REQ\_P, [791](#)
  - PLL\_SSCG1\_MF\_M, [791](#)
  - PLL\_SSCG1\_MF\_P, [791](#)
  - PLL\_SSCG1\_MOD\_PD\_SSCGCLK\_N, [791](#)
  - PLL\_SSCG1\_MOD\_PD\_SSCGCLK\_N\_P, [791](#)
  - PLL\_SSCG1\_MR\_M, [791](#)
  - PLL\_SSCG1\_MR\_P, [791](#)
  - PLL\_SSCG\_DITHER\_VALUE, [791](#)
  - PLL\_SSCG\_MC\_COMP\_VALUE, [791](#)
  - PLL\_SSCG\_MF\_FREQ\_VALUE, [792](#)
  - PLL\_SSCG\_MR\_DEPTH\_VALUE, [792](#)
  - PVALMAX, [792](#)
  - pllDecodeM, [795](#)
  - pllDecodeN, [795](#)



- pllDecodeP, [795](#)
- pllEncodeM, [795](#)
- pllEncodeN, [795](#)
- pllEncodeP, [795](#)
- pllFindSel, [795](#)
- SYS\_PLL\_BANDSEL, [792](#)
- SYS\_PLL\_BYPASS, [792](#)
- SYS\_PLL\_BYPASSCCODIV2, [792](#)
- SYS\_PLL\_DIRECTI, [792](#)
- SYS\_PLL\_DIRECTO, [792](#)
- SYS\_PLL\_SELI, [792](#)
- SYS\_PLL\_SELP, [792](#)
- SYS\_PLL\_SELRL, [793](#)
- SYS\_PLL\_UPLIMOFF, [793](#)
- SYSCON\_SYSPLLCTRL\_BANDSEL\_SSCGRE←  
G\_N, [793](#)
- SYSCON\_SYSPLLCTRL\_BANDSEL\_SSCGRE←  
G\_N\_P, [793](#)
- SYSCON\_SYSPLLCTRL\_BYPASS, [793](#)
- SYSCON\_SYSPLLCTRL\_BYPASS\_FBDIV2, [793](#)
- SYSCON\_SYSPLLCTRL\_BYPASS\_FBDIV2\_P,  
[793](#)
- SYSCON\_SYSPLLCTRL\_BYPASS\_P, [793](#)
- SYSCON\_SYSPLLCTRL\_DIRECTI, [793](#)
- SYSCON\_SYSPLLCTRL\_DIRECTI\_P, [793](#)
- SYSCON\_SYSPLLCTRL\_DIRECTO, [793](#)
- SYSCON\_SYSPLLCTRL\_DIRECTO\_P, [794](#)
- SYSCON\_SYSPLLCTRL\_SELI\_M, [794](#)
- SYSCON\_SYSPLLCTRL\_SELI\_P, [794](#)
- SYSCON\_SYSPLLCTRL\_SELP\_M, [794](#)
- SYSCON\_SYSPLLCTRL\_SELP\_P, [794](#)
- SYSCON\_SYSPLLCTRL\_SELRL\_M, [794](#)
- SYSCON\_SYSPLLCTRL\_SELRL\_P, [794](#)
- SYSCON\_SYSPLLCTRL\_UPLIMOFF, [794](#)
- SYSCON\_SYSPLLCTRL\_UPLIMOFF\_P, [794](#)
- SYSCON\_SYSPLLSTAT\_LOCK, [794](#)
- SYSCON\_SYSPLLSTAT\_LOCK\_P, [794](#)
- pllDecodeM
  - pll\_5411x.c, [795](#)
- pllDecodeN
  - pll\_5411x.c, [795](#)
- pllDecodeP
  - pll\_5411x.c, [795](#)
- pllEncodeM
  - pll\_5411x.c, [795](#)
- pllEncodeN
  - pll\_5411x.c, [795](#)
- pllEncodeP
  - pll\_5411x.c, [795](#)
- pllFindSel
  - pll\_5411x.c, [795](#)
- pllRate
  - PLL\_SETUP\_T, [631](#)
- port
  - PINMUX\_GRP\_T, [628](#)
- PostDelay
  - SPIM\_DELAY\_CONFIG\_T, [639](#)
- PreDelay
  - SPIM\_DELAY\_CONFIG\_T, [639](#)
- preac2coef
  - CHIP: LPC5411X DMIC driver, [146](#)
- preac4coef
  - CHIP: LPC5411X DMIC driver, [146](#)
- put\_rx\_data
  - spim\_5411x.c, [799](#)
  - spis\_5411x.c, [800](#)
- Q
  - APSR\_Type, [550](#)
  - xPSR\_Type, [668](#)
- RB\_INDH
  - ring\_buffer.c, [796](#)
- RB\_INDТ
  - ring\_buffer.c, [796](#)
- RB\_VHEAD
  - CHIP: Simple ring buffer implementation, [419](#)
- RB\_VTAIL
  - CHIP: Simple ring buffer implementation, [419](#)
- REGMODE\_H
  - LPC\_SCT\_T, [602](#)
- REGMODE\_L
  - LPC\_SCT\_T, [602](#)
- REMAP\_BOOT\_LOADER\_MODE
  - CHIP: LPC5411X System and Control Driver, [330](#)
- REMAP\_USER\_FLASH\_MODE
  - CHIP: LPC5411X System and Control Driver, [331](#)
- REMAP\_USER\_RAM\_MODE
  - CHIP: LPC5411X System and Control Driver, [330](#)
- REQUEST\_CLASS
  - USBD\_Core, [537](#)
- REQUEST\_DEVICE\_TO\_HOST
  - USBD\_Core, [537](#)
- REQUEST\_HOST\_TO\_DEVICE
  - USBD\_Core, [538](#)
- REQUEST\_RESERVED
  - USBD\_Core, [538](#)
- REQUEST\_STANDARD
  - USBD\_Core, [538](#)
- REQUEST\_TO\_DEVICE
  - USBD\_Core, [538](#)
- REQUEST\_TO\_ENDPOINT
  - USBD\_Core, [538](#)
- REQUEST\_TO\_INTERFACE
  - USBD\_Core, [538](#)
- REQUEST\_TO\_OTHER
  - USBD\_Core, [538](#)
- REQUEST\_TYPE, [632](#)
- REQUEST\_VENDOR
  - USBD\_Core, [538](#)
- RES
  - LPC\_SCT\_T, [602](#)
- RESERVED
  - LPC\_DMA\_CHANNEL\_T, [575](#)
  - LPC\_MBOXIRQ\_T, [589](#)
- RESERVED0
  - CHIP: LPC5411x I2C driver, [415](#)



- DWT\_Type, [559](#)
- ITM\_Type, [569](#)
- LPC\_ADC\_T, [571](#)
- LPC\_ASYNC\_SYSCON\_T, [573](#)
- LPC\_DMA\_COMMON\_T, [577](#)
- LPC\_DMA\_T, [579](#)
- LPC\_GPIOGROUPINT\_T, [583](#)
- LPC\_I2S\_T, [586](#)
- LPC\_PMU\_T, [594](#)
- LPC\_SPI\_T, [606](#)
- LPC\_SYSCON\_T, [613](#)
- LPC\_TIMER\_T, [620](#)
- LPC\_WWDT\_T, [625](#)
- NVIC\_Type, [627](#)
- SCB\_Type, [636](#)
- SCnSCB\_Type, [637](#)
- TPI\_Type, [646](#)
- RESERVED00
  - LPC\_I2S\_T, [586](#)
- RESERVED0A
  - LPC\_I2S\_T, [586](#)
- RESERVED1
  - CHIP: LPC5411x I2C driver, [415](#)
  - DWT\_Type, [559](#)
  - ITM\_Type, [569](#)
  - LPC\_ASYNC\_SYSCON\_T, [574](#)
  - LPC\_DMA\_COMMON\_T, [578](#)
  - LPC\_GPIOGROUPINT\_T, [583](#)
  - LPC\_INMUX\_T, [587](#)
  - LPC\_MBOX\_T, [588](#)
  - LPC\_SCT\_T, [602](#)
  - LPC\_SPI\_T, [606](#)
  - LPC\_SYSCON\_T, [613](#)
  - SCB\_Type, [636](#)
  - TPI\_Type, [646](#)
- RESERVED10
  - LPC\_DMA\_COMMON\_T, [578](#)
  - LPC\_SCT\_T, [602](#)
  - LPC\_SYSCON\_T, [613](#)
- RESERVED11
  - LPC\_SYSCON\_T, [613](#)
- RESERVED12
  - LPC\_SYSCON\_T, [613](#)
- RESERVED13
  - LPC\_SYSCON\_T, [613](#)
- RESERVED14
  - LPC\_SYSCON\_T, [613](#)
- RESERVED15
  - LPC\_SYSCON\_T, [614](#)
- RESERVED16
  - LPC\_SYSCON\_T, [614](#)
- RESERVED16A
  - LPC\_SYSCON\_T, [614](#)
- RESERVED17
  - LPC\_SYSCON\_T, [614](#)
- RESERVED18
  - LPC\_SYSCON\_T, [614](#)
- RESERVED19
  - LPC\_SYSCON\_T, [614](#)
- RESERVED2
  - CHIP: LPC5411x I2C driver, [415](#)
  - DWT\_Type, [559](#)
  - ITM\_Type, [569](#)
  - LPC\_DMA\_COMMON\_T, [578](#)
  - LPC\_DMA\_T, [580](#)
  - LPC\_INMUX\_T, [587](#)
  - LPC\_MBOX\_T, [589](#)
  - LPC\_SCT\_T, [602](#)
  - LPC\_SYSCON\_T, [614](#)
  - NVIC\_Type, [627](#)
  - TPI\_Type, [646](#)
- RESERVED20
  - LPC\_SYSCON\_T, [614](#)
- RESERVED21
  - LPC\_SYSCON\_T, [614](#)
- RESERVED22
  - LPC\_SYSCON\_T, [614](#)
- RESERVED23
  - LPC\_SYSCON\_T, [614](#)
- RESERVED24
  - LPC\_SYSCON\_T, [614](#)
- RESERVED25
  - LPC\_SYSCON\_T, [615](#)
- RESERVED26
  - LPC\_SYSCON\_T, [615](#)
- RESERVED27
  - LPC\_SYSCON\_T, [615](#)
- RESERVED28
  - LPC\_SYSCON\_T, [615](#)
- RESERVED29
  - LPC\_SYSCON\_T, [615](#)
- RESERVED3
  - ITM\_Type, [569](#)
  - LPC\_DMA\_COMMON\_T, [578](#)
  - LPC\_INMUX\_T, [587](#)
  - LPC\_SCT\_T, [602](#)
  - LPC\_SYSCON\_T, [615](#)
  - NVIC\_Type, [627](#)
  - TPI\_Type, [646](#)
- RESERVED30
  - LPC\_SYSCON\_T, [615](#)
- RESERVED31
  - LPC\_SYSCON\_T, [615](#)
- RESERVED32
  - LPC\_SYSCON\_T, [615](#)
- RESERVED33
  - LPC\_SYSCON\_T, [615](#)
- RESERVED34
  - LPC\_SYSCON\_T, [615](#)
- RESERVED35
  - LPC\_SYSCON\_T, [615](#)
- RESERVED36
  - LPC\_SYSCON\_T, [616](#)
- RESERVED37
  - LPC\_SYSCON\_T, [616](#)
- RESERVED4

- ITM\_Type, [569](#)
- LPC\_DMA\_COMMON\_T, [578](#)
- LPC\_SYSCON\_T, [616](#)
- NVIC\_Type, [627](#)
- TPI\_Type, [646](#)
- RESERVED5
  - ITM\_Type, [569](#)
  - LPC\_DMA\_COMMON\_T, [578](#)
  - LPC\_I2S\_T, [586](#)
  - LPC\_SYSCON\_T, [616](#)
  - NVIC\_Type, [627](#)
  - TPI\_Type, [646](#)
- RESERVED6
  - LPC\_DMA\_COMMON\_T, [578](#)
  - LPC\_SCT\_T, [603](#)
  - LPC\_SYSCON\_T, [616](#)
- RESERVED7
  - LPC\_DMA\_COMMON\_T, [578](#)
  - LPC\_SYSCON\_T, [616](#)
  - TPI\_Type, [646](#)
- RESERVED8
  - LPC\_DMA\_COMMON\_T, [578](#)
  - LPC\_SYSCON\_T, [616](#)
- RESERVED9
  - LPC\_DMA\_COMMON\_T, [578](#)
  - LPC\_SCT\_T, [603](#)
  - LPC\_SYSCON\_T, [616](#)
- RESET
  - LPC Public Types, [480](#)
- RESET\_ADC
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_ADC0
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_CRC
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_DMA
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_DMIC
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FLASH
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FLEXCOMM0
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FLEXCOMM1
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FLEXCOMM2
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FLEXCOMM3
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FLEXCOMM4
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FLEXCOMM5
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FLEXCOMM6
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FLEXCOMM7
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_FMC
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_GINT
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_GPIO0
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_GPIO1
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_IOCON
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_MRT
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_MUX
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_PINT
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_SCT
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_SCT0
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_SPIFI
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_TIMER0
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_TIMER1
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_TIMER2
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_TIMER3
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_TIMER4
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_USB
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_UTICK
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RESET\_WWDT
  - CHIP: LPC5411X System and Control Driver, [331](#)
- RINGBUFF\_T, [633](#)
  - copy, [633](#)
  - count, [633](#)
  - data, [633](#)
  - head, [633](#)
  - itemSz, [633](#)
  - tail, [633](#)
- RISE
  - LPC\_PIN\_INT\_T, [593](#)
- ROM\_SPI\_CLOCK\_CPHA0\_CPOL0
  - CHIP: LPC5411X SPI driver, [280](#)
- ROM\_SPI\_CLOCK\_CPHA0\_CPOL1
  - CHIP: LPC5411X SPI driver, [280](#)
- ROM\_SPI\_CLOCK\_CPHA1\_CPOL0
  - CHIP: LPC5411X SPI driver, [280](#)
- ROM\_SPI\_CLOCK\_CPHA1\_CPOL1
  - CHIP: LPC5411X SPI driver, [280](#)
- ROM\_SPI\_CLOCK\_MODE0
  - CHIP: LPC5411X SPI driver, [280](#)
- ROM\_SPI\_CLOCK\_MODE1
  - CHIP: LPC5411X SPI driver, [280](#)

- ROM\_SPI\_CLOCK\_MODE2
  - CHIP: LPC5411X SPI driver, [280](#)
- ROM\_SPI\_CLOCK\_MODE3
  - CHIP: LPC5411X SPI driver, [280](#)
- ROM\_SPI\_CLOCK\_MODE\_T
  - CHIP: LPC5411X SPI driver, [280](#)
- RSERVED1
  - NVIC\_Type, [627](#)
- RTC\_CTRL\_ALARM1HZ
  - CHIP: LPC5411X Real Time clock, [259](#)
- RTC\_CTRL\_ALARMDPD\_EN
  - CHIP: LPC5411X Real Time clock, [259](#)
- RTC\_CTRL\_MASK
  - CHIP: LPC5411X Real Time clock, [259](#)
- RTC\_CTRL\_RTC1KHZ\_EN
  - CHIP: LPC5411X Real Time clock, [259](#)
- RTC\_CTRL\_RTC\_EN
  - CHIP: LPC5411X Real Time clock, [259](#)
- RTC\_CTRL\_RTC\_OSC\_BYPASS
  - CHIP: LPC5411X Real Time clock, [259](#)
- RTC\_CTRL\_RTC\_OSC\_PD
  - CHIP: LPC5411X Real Time clock, [259](#)
- RTC\_CTRL\_SWRESET
  - CHIP: LPC5411X Real Time clock, [260](#)
- RTC\_CTRL\_WAKE1KHZ
  - CHIP: LPC5411X Real Time clock, [260](#)
- RTC\_CTRL\_WAKEDPD\_EN
  - CHIP: LPC5411X Real Time clock, [260](#)
- RTC\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- RTCOSCCTRL
  - LPC\_SYSCON\_T, [616](#)
- RTOS support code, [493](#)
- RTOS: FreeRTOS support code, [494](#)
- Recipient
  - \_BM\_T, [551](#)
- register\_location
  - CHIP: LPC5411X DMA Service driver, [132](#)
- reserved
  - CHIP: LPC5411X DMIC driver, [146](#)
  - SPI\_CFGSETUP\_T, [638](#)
- reserved0
  - CHIP: LPC5411X DMIC driver, [146](#)
- reserved1
  - CHIP: LPC5411X DMIC driver, [146](#)
- Reserved1\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- Reserved\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- reserved\_adcaltd
  - LPC\_ROM\_API\_T, [594](#)
- reserved\_clib
  - LPC\_ROM\_API\_T, [594](#)
- reserved\_div
  - LPC\_ROM\_API\_T, [595](#)
- reserved\_dmaaltd
  - LPC\_ROM\_API\_T, [595](#)
- reserved\_flexcomm
  - LPC\_ROM\_API\_T, [595](#)
- reserved\_i2cm
  - LPC\_ROM\_API\_T, [595](#)
- reserved\_i2cmon
  - LPC\_ROM\_API\_T, [595](#)
- reserved\_i2cs
  - LPC\_ROM\_API\_T, [595](#)
- reserved\_power
  - LPC\_ROM\_API\_T, [595](#)
- reserved\_spim
  - LPC\_ROM\_API\_T, [595](#)
- reserved\_spis
  - LPC\_ROM\_API\_T, [595](#)
- reserved\_uartalt
  - LPC\_ROM\_API\_T, [596](#)
- reserved\_usart
  - LPC\_ROM\_API\_T, [596](#)
- Reset\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- ResetISR
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [808](#)
- RightLow
  - I2S\_AUDIO\_FORMAT\_T, [563](#)
- ring\_buffer.c
  - RB\_INDH, [796](#)
  - RB\_INDT, [796](#)
- RingBuffer\_Flush
  - CHIP: Simple ring buffer implementation, [420](#)
- RingBuffer\_GetCount
  - CHIP: Simple ring buffer implementation, [420](#)
- RingBuffer\_GetFree
  - CHIP: Simple ring buffer implementation, [420](#)
- RingBuffer\_GetSize
  - CHIP: Simple ring buffer implementation, [420](#)
- RingBuffer\_Init
  - CHIP: Simple ring buffer implementation, [420](#)
- RingBuffer\_Insert
  - CHIP: Simple ring buffer implementation, [422](#)
- RingBuffer\_InsertMult
  - CHIP: Simple ring buffer implementation, [422](#)
- RingBuffer\_IsEmpty
  - CHIP: Simple ring buffer implementation, [422](#)
- RingBuffer\_IsFull
  - CHIP: Simple ring buffer implementation, [423](#)
- RingBuffer\_Pop
  - CHIP: Simple ring buffer implementation, [423](#)
- RingBuffer\_PopMult

- CHIP: Simple ring buffer implementation, [423](#)
- rtc\_ut.c
  - DAYSERLEAPYEAR, [797](#)
  - DAYSERWEEK, [797](#)
  - DAYSERYEAR, [797](#)
  - daysPerMonth, [798](#)
  - GetDMLY, [798](#)
  - HOURSERDAY, [797](#)
  - MINSERHOUR, [797](#)
  - MONETHSERYEAR, [797](#)
  - SECSERDAY, [797](#)
  - SECSERHOUR, [798](#)
  - SECSERMIN, [798](#)
- rxBuff
  - I2CM\_XFER\_T, [560](#)
- rxCount
  - SPIM\_XFER\_T, [640](#)
  - SPIS\_XFER\_T, [642](#)
- rxData
  - SPIM\_XFER\_T, [640](#)
  - SPIS\_XFER\_T, [642](#)
- rxIndex
  - SPIM\_XFER\_T, [640](#)
  - SPIS\_XFER\_T, [642](#)
- rxSz
  - I2CM\_XFER\_T, [560](#)
- SCB
  - Core Definitions, [452](#)
- SCB\_AIRCR\_ENDIANESS\_Msk
  - System Control Block (SCB), [501](#)
- SCB\_AIRCR\_ENDIANESS\_Pos
  - System Control Block (SCB), [501](#)
- SCB\_AIRCR\_PRIGROUP\_Msk
  - System Control Block (SCB), [501](#)
- SCB\_AIRCR\_PRIGROUP\_Pos
  - System Control Block (SCB), [501](#)
- SCB\_AIRCR\_SYSRESETREQ\_Msk
  - System Control Block (SCB), [501](#), [502](#)
- SCB\_AIRCR\_SYSRESETREQ\_Pos
  - System Control Block (SCB), [502](#)
- SCB\_AIRCR\_VECTCLRACTIVE\_Msk
  - System Control Block (SCB), [502](#)
- SCB\_AIRCR\_VECTCLRACTIVE\_Pos
  - System Control Block (SCB), [502](#)
- SCB\_AIRCR\_VECTKEY\_Msk
  - System Control Block (SCB), [502](#)
- SCB\_AIRCR\_VECTKEY\_Pos
  - System Control Block (SCB), [503](#)
- SCB\_AIRCR\_VECTKEYSTAT\_Msk
  - System Control Block (SCB), [503](#)
- SCB\_AIRCR\_VECTKEYSTAT\_Pos
  - System Control Block (SCB), [503](#)
- SCB\_AIRCR\_VECTRESET\_Msk
  - System Control Block (SCB), [503](#)
- SCB\_AIRCR\_VECTRESET\_Pos
  - System Control Block (SCB), [503](#)
- SCB\_BASE
  - Core Definitions, [452](#), [453](#)
- SCB\_CCR\_BFHFNMIGN\_Msk
  - System Control Block (SCB), [503](#)
- SCB\_CCR\_BFHFNMIGN\_Pos
  - System Control Block (SCB), [504](#)
- SCB\_CCR\_DIV\_0\_TRP\_Msk
  - System Control Block (SCB), [504](#)
- SCB\_CCR\_DIV\_0\_TRP\_Pos
  - System Control Block (SCB), [504](#)
- SCB\_CCR\_NONBASETHRDENA\_Msk
  - System Control Block (SCB), [504](#)
- SCB\_CCR\_NONBASETHRDENA\_Pos
  - System Control Block (SCB), [504](#)
- SCB\_CCR\_STKALIGN\_Msk
  - System Control Block (SCB), [504](#)
- SCB\_CCR\_STKALIGN\_Pos
  - System Control Block (SCB), [504](#)
- SCB\_CCR\_UNALIGN\_TRP\_Msk
  - System Control Block (SCB), [505](#)
- SCB\_CCR\_UNALIGN\_TRP\_Pos
  - System Control Block (SCB), [505](#)
- SCB\_CCR\_USERSETMPEND\_Msk
  - System Control Block (SCB), [505](#)
- SCB\_CCR\_USERSETMPEND\_Pos
  - System Control Block (SCB), [505](#)
- SCB\_CFSR\_BUSFAULTSR\_Msk
  - System Control Block (SCB), [505](#)
- SCB\_CFSR\_BUSFAULTSR\_Pos
  - System Control Block (SCB), [505](#)
- SCB\_CFSR\_MEMFAULTSR\_Msk
  - System Control Block (SCB), [505](#)
- SCB\_CFSR\_MEMFAULTSR\_Pos
  - System Control Block (SCB), [506](#)
- SCB\_CFSR\_USGFAULTSR\_Msk
  - System Control Block (SCB), [506](#)
- SCB\_CFSR\_USGFAULTSR\_Pos
  - System Control Block (SCB), [506](#)
- SCB\_CPUID\_ARCHITECTURE\_Msk
  - System Control Block (SCB), [506](#)
- SCB\_CPUID\_ARCHITECTURE\_Pos
  - System Control Block (SCB), [506](#)
- SCB\_CPUID\_IMPLEMENTER\_Msk
  - System Control Block (SCB), [506](#)
- SCB\_CPUID\_IMPLEMENTER\_Pos
  - System Control Block (SCB), [507](#)
- SCB\_CPUID\_PARTNO\_Msk
  - System Control Block (SCB), [507](#)
- SCB\_CPUID\_PARTNO\_Pos
  - System Control Block (SCB), [507](#)
- SCB\_CPUID\_REVISION\_Msk
  - System Control Block (SCB), [507](#)
- SCB\_CPUID\_REVISION\_Pos
  - System Control Block (SCB), [507](#), [508](#)
- SCB\_CPUID\_VARIANT\_Msk
  - System Control Block (SCB), [508](#)
- SCB\_CPUID\_VARIANT\_Pos
  - System Control Block (SCB), [508](#)
- SCB\_DFSR\_BKPT\_Msk
  - System Control Block (SCB), [508](#)

- SCB\_DFSR\_BKPT\_Pos
  - System Control Block (SCB), [508](#)
- SCB\_DFSR\_DWTTRAP\_Msk
  - System Control Block (SCB), [508](#)
- SCB\_DFSR\_DWTTRAP\_Pos
  - System Control Block (SCB), [508](#)
- SCB\_DFSR\_EXTERNAL\_Msk
  - System Control Block (SCB), [509](#)
- SCB\_DFSR\_EXTERNAL\_Pos
  - System Control Block (SCB), [509](#)
- SCB\_DFSR\_HALTED\_Msk
  - System Control Block (SCB), [509](#)
- SCB\_DFSR\_HALTED\_Pos
  - System Control Block (SCB), [509](#)
- SCB\_DFSR\_VCATCH\_Msk
  - System Control Block (SCB), [509](#)
- SCB\_DFSR\_VCATCH\_Pos
  - System Control Block (SCB), [509](#)
- SCB\_HFSR\_DEBUGEVT\_Msk
  - System Control Block (SCB), [509](#)
- SCB\_HFSR\_DEBUGEVT\_Pos
  - System Control Block (SCB), [509](#)
- SCB\_HFSR\_FORCED\_Msk
  - System Control Block (SCB), [509](#)
- SCB\_HFSR\_FORCED\_Pos
  - System Control Block (SCB), [510](#)
- SCB\_HFSR\_VECTTBL\_Msk
  - System Control Block (SCB), [510](#)
- SCB\_HFSR\_VECTTBL\_Pos
  - System Control Block (SCB), [510](#)
- SCB\_ICSR\_ISRPENDING\_Msk
  - System Control Block (SCB), [510](#)
- SCB\_ICSR\_ISRPENDING\_Pos
  - System Control Block (SCB), [510](#)
- SCB\_ICSR\_ISRPREEMPT\_Msk
  - System Control Block (SCB), [510](#)
- SCB\_ICSR\_ISRPREEMPT\_Pos
  - System Control Block (SCB), [511](#)
- SCB\_ICSR\_NMIPENDSET\_Msk
  - System Control Block (SCB), [511](#)
- SCB\_ICSR\_NMIPENDSET\_Pos
  - System Control Block (SCB), [511](#)
- SCB\_ICSR\_PENDSTCLR\_Msk
  - System Control Block (SCB), [511](#)
- SCB\_ICSR\_PENDSTCLR\_Pos
  - System Control Block (SCB), [511](#), [512](#)
- SCB\_ICSR\_PENDSTSET\_Msk
  - System Control Block (SCB), [512](#)
- SCB\_ICSR\_PENDSTSET\_Pos
  - System Control Block (SCB), [512](#)
- SCB\_ICSR\_PENDSVCLR\_Msk
  - System Control Block (SCB), [512](#)
- SCB\_ICSR\_PENDSVCLR\_Pos
  - System Control Block (SCB), [512](#)
- SCB\_ICSR\_PENDSVSET\_Msk
  - System Control Block (SCB), [513](#)
- SCB\_ICSR\_PENDSVSET\_Pos
  - System Control Block (SCB), [513](#)
- SCB\_ICSR\_RETTOBASE\_Msk
  - System Control Block (SCB), [513](#)
- SCB\_ICSR\_RETTOBASE\_Pos
  - System Control Block (SCB), [513](#)
- SCB\_ICSR\_VECTACTIVE\_Msk
  - System Control Block (SCB), [513](#)
- SCB\_ICSR\_VECTACTIVE\_Pos
  - System Control Block (SCB), [513](#), [514](#)
- SCB\_ICSR\_VECTPENDING\_Msk
  - System Control Block (SCB), [514](#)
- SCB\_ICSR\_VECTPENDING\_Pos
  - System Control Block (SCB), [514](#)
- SCB\_SCR\_SEVONPEND\_Msk
  - System Control Block (SCB), [514](#)
- SCB\_SCR\_SEVONPEND\_Pos
  - System Control Block (SCB), [514](#)
- SCB\_SCR\_SLEEPDEEP\_Msk
  - System Control Block (SCB), [515](#)
- SCB\_SCR\_SLEEPDEEP\_Pos
  - System Control Block (SCB), [515](#)
- SCB\_SCR\_SLEEPONEXIT\_Msk
  - System Control Block (SCB), [515](#)
- SCB\_SCR\_SLEEPONEXIT\_Pos
  - System Control Block (SCB), [515](#)
- SCB\_SHCSR\_BUSFAULTACT\_Msk
  - System Control Block (SCB), [515](#)
- SCB\_SHCSR\_BUSFAULTACT\_Pos
  - System Control Block (SCB), [516](#)
- SCB\_SHCSR\_BUSFAULTENA\_Msk
  - System Control Block (SCB), [516](#)
- SCB\_SHCSR\_BUSFAULTENA\_Pos
  - System Control Block (SCB), [516](#)
- SCB\_SHCSR\_BUSFAULTPENDEDED\_Msk
  - System Control Block (SCB), [516](#)
- SCB\_SHCSR\_BUSFAULTPENDEDED\_Pos
  - System Control Block (SCB), [516](#)
- SCB\_SHCSR\_MEMFAULTACT\_Msk
  - System Control Block (SCB), [516](#)
- SCB\_SHCSR\_MEMFAULTACT\_Pos
  - System Control Block (SCB), [516](#)
- SCB\_SHCSR\_MEMFAULTENA\_Msk
  - System Control Block (SCB), [516](#)
- SCB\_SHCSR\_MEMFAULTENA\_Pos
  - System Control Block (SCB), [516](#)
- SCB\_SHCSR\_MEMFAULTPENDEDED\_Msk
  - System Control Block (SCB), [517](#)
- SCB\_SHCSR\_MEMFAULTPENDEDED\_Pos
  - System Control Block (SCB), [517](#)
- SCB\_SHCSR\_MONITORACT\_Msk
  - System Control Block (SCB), [517](#)
- SCB\_SHCSR\_MONITORACT\_Pos
  - System Control Block (SCB), [517](#)
- SCB\_SHCSR\_PENDSVACT\_Msk
  - System Control Block (SCB), [517](#)
- SCB\_SHCSR\_PENDSVACT\_Pos
  - System Control Block (SCB), [517](#)
- SCB\_SHCSR\_SVCALLACT\_Msk
  - System Control Block (SCB), [517](#)

- SCB\_SHCSR\_SVCALLACT\_Pos
  - System Control Block (SCB), [517](#)
- SCB\_SHCSR\_SVCALLPENDEd\_Msk
  - System Control Block (SCB), [517](#), [518](#)
- SCB\_SHCSR\_SVCALLPENDEd\_Pos
  - System Control Block (SCB), [518](#)
- SCB\_SHCSR\_SYSTICKACT\_Msk
  - System Control Block (SCB), [518](#)
- SCB\_SHCSR\_SYSTICKACT\_Pos
  - System Control Block (SCB), [518](#)
- SCB\_SHCSR\_USGFAULTACT\_Msk
  - System Control Block (SCB), [518](#)
- SCB\_SHCSR\_USGFAULTACT\_Pos
  - System Control Block (SCB), [518](#)
- SCB\_SHCSR\_USGFAULTENA\_Msk
  - System Control Block (SCB), [518](#)
- SCB\_SHCSR\_USGFAULTENA\_Pos
  - System Control Block (SCB), [518](#)
- SCB\_SHCSR\_USGFAULTPENDEd\_Msk
  - System Control Block (SCB), [519](#)
- SCB\_SHCSR\_USGFAULTPENDEd\_Pos
  - System Control Block (SCB), [519](#)
- SCB\_Type, [634](#)
  - ADR, [634](#)
  - AFSR, [634](#)
  - AIRCR, [634](#)
  - BFAR, [634](#)
  - CCR, [635](#)
  - CFSR, [635](#)
  - CPACR, [635](#)
  - CPUID, [635](#)
  - DFR, [635](#)
  - DFSR, [635](#)
  - HFSR, [635](#)
  - ICSR, [635](#)
  - ISAR, [635](#)
  - MMFAR, [636](#)
  - MMFR, [636](#)
  - PFR, [636](#)
  - RESERVED0, [636](#)
  - RESERVED1, [636](#)
  - SCR, [636](#)
  - SHCSR, [636](#)
  - SHP, [636](#)
  - VTOR, [636](#)
- SCB\_VTOR\_TBLOFF\_Msk
  - System Control Block (SCB), [519](#)
- SCB\_VTOR\_TBLOFF\_Pos
  - System Control Block (SCB), [519](#)
- SCKPol
  - I2S\_AUDIO\_FORMAT\_T, [563](#)
- SCR
  - SCB\_Type, [636](#)
- SCS\_BASE
  - Core Definitions, [453](#)
- SCT0\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
- CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- SCT\_CONFIG\_16BIT\_COUNTER
  - CHIP: LPC5411X State Configurable Timer driver, [310](#)
- SCT\_CONFIG\_32BIT\_COUNTER
  - CHIP: LPC5411X State Configurable Timer driver, [310](#)
- SCT\_CONFIG\_AUTOLIMIT\_H
  - CHIP: LPC5411X State Configurable Timer driver, [310](#)
- SCT\_CONFIG\_AUTOLIMIT\_L
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_AUTOLIMIT\_U
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_0
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_1
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_2
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_3
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_4
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_5
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_6
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_FALLING\_IN\_7
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_0
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_1
  - CHIP: LPC5411X State Configurable Timer driver, [311](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_2
  - CHIP: LPC5411X State Configurable Timer driver, [312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_3
  - CHIP: LPC5411X State Configurable Timer driver, [312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_4
  - CHIP: LPC5411X State Configurable Timer driver, [312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_5



- CHIP: LPC5411X State Configurable Timer driver,  
[312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_6  
CHIP: LPC5411X State Configurable Timer driver,  
[312](#)
- SCT\_CONFIG\_CKSEL\_RISING\_IN\_7  
CHIP: LPC5411X State Configurable Timer driver,  
[312](#)
- SCT\_CONFIG\_CLKMODE\_BUSCLK  
CHIP: LPC5411X State Configurable Timer driver,  
[312](#)
- SCT\_CONFIG\_CLKMODE\_INCLK  
CHIP: LPC5411X State Configurable Timer driver,  
[312](#)
- SCT\_CONFIG\_CLKMODE\_INEDGECLK  
CHIP: LPC5411X State Configurable Timer driver,  
[312](#)
- SCT\_CONFIG\_CLKMODE\_PRESCALED\_SCT\_INP↔  
UT  
CHIP: LPC5411X State Configurable Timer driver,  
[312](#)
- SCT\_CONFIG\_CLKMODE\_PRESCALED\_SYSCLK  
CHIP: LPC5411X State Configurable Timer driver,  
[312](#)
- SCT\_CONFIG\_CLKMODE\_SCT\_INPUT  
CHIP: LPC5411X State Configurable Timer driver,  
[313](#)
- SCT\_CONFIG\_CLKMODE\_SCTCLK  
CHIP: LPC5411X State Configurable Timer driver,  
[313](#)
- SCT\_CONFIG\_CLKMODE\_SYSCLK  
CHIP: LPC5411X State Configurable Timer driver,  
[313](#)
- SCT\_CONFIG\_NORELOADH  
CHIP: LPC5411X State Configurable Timer driver,  
[313](#)
- SCT\_CONFIG\_NORELOADL\_U  
CHIP: LPC5411X State Configurable Timer driver,  
[313](#)
- SCT\_CTRL\_BIDIR\_H  
CHIP: LPC5411X State Configurable Timer driver,  
[313](#)
- SCT\_CTRL\_BIDIR\_L  
CHIP: LPC5411X State Configurable Timer driver,  
[313](#)
- SCT\_CTRL\_CLRCTR\_H  
CHIP: LPC5411X State Configurable Timer driver,  
[313](#)
- SCT\_CTRL\_CLRCTR\_L  
CHIP: LPC5411X State Configurable Timer driver,  
[313](#)
- SCT\_CTRL\_HALT\_H  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_CTRL\_HALT\_L  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_CTRL\_PRE\_H  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_CTRL\_PRE\_L  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_CTRL\_STOP\_H  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_CTRL\_STOP\_L  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_EV\_CTRL\_COMBMODE\_AND  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_EV\_CTRL\_COMBMODE\_IO  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_EV\_CTRL\_COMBMODE\_MATCH  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_EV\_CTRL\_COMBMODE\_OR  
CHIP: LPC5411X State Configurable Timer driver,  
[314](#)
- SCT\_EV\_CTRL\_DIRECTION\_DOWN  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_DIRECTION\_INDEPENDENT  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_DIRECTION\_UP  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_HEVENT\_H  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_HEVENT\_L  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_IOCOND\_FALL  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_IOCOND\_HIGH  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_IOCOND\_LOW  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_IOCOND\_RISE  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_IOSEL  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_MATCHMEM  
CHIP: LPC5411X State Configurable Timer driver,  
[315](#)
- SCT\_EV\_CTRL\_MATCHSEL

- CHIP: LPC5411X State Configurable Timer driver, [315](#)
- SCT\_EV\_CTRL\_OUTSEL\_INPUT
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EV\_CTRL\_OUTSEL\_OUTPUT
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EV\_CTRL\_STATELD
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EV\_CTRL\_STATEV
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EVT\_0
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EVT\_1
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EVT\_10
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_EVT\_11
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_EVT\_12
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_EVT\_13
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_EVT\_14
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_EVT\_15
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_EVT\_2
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EVT\_3
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EVT\_4
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EVT\_5
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_EVT\_6
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_EVT\_7
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_EVT\_8
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_EVT\_9
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_IRQHandler
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- SCT\_IRQn
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- SCT\_MATCH\_0
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_1
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_10
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_11
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_12
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_13
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_14
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_15
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_2
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_3
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_4
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_5
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_6
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_7
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_8
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)



- CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_MATCH\_9
  - CHIP: LPC5411X State Configurable Timer driver, [317](#)
- SCT\_RES\_CLEAR\_OUTPUT
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_RES\_NOCHANGE
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_RES\_SET\_OUTPUT
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCT\_RES\_TOGGLE\_OUTPUT
  - CHIP: LPC5411X State Configurable Timer driver, [316](#)
- SCnSCB
  - Core Definitions, [453](#)
- SCnSCB\_ACTLR\_DISDEFWBUF\_Msk
  - System Controls not in SCB (SCnSCB), [520](#)
- SCnSCB\_ACTLR\_DISDEFWBUF\_Pos
  - System Controls not in SCB (SCnSCB), [520](#)
- SCnSCB\_ACTLR\_DISFOLD\_Msk
  - System Controls not in SCB (SCnSCB), [520](#)
- SCnSCB\_ACTLR\_DISFOLD\_Pos
  - System Controls not in SCB (SCnSCB), [520](#)
- SCnSCB\_ACTLR\_DISFPCA\_Msk
  - System Controls not in SCB (SCnSCB), [520](#)
- SCnSCB\_ACTLR\_DISFPCA\_Pos
  - System Controls not in SCB (SCnSCB), [521](#)
- SCnSCB\_ACTLR\_DISMCYCINT\_Msk
  - System Controls not in SCB (SCnSCB), [521](#)
- SCnSCB\_ACTLR\_DISMCYCINT\_Pos
  - System Controls not in SCB (SCnSCB), [521](#)
- SCnSCB\_ACTLR\_DISOOF\_P\_Msk
  - System Controls not in SCB (SCnSCB), [521](#)
- SCnSCB\_ACTLR\_DISOOF\_Pos
  - System Controls not in SCB (SCnSCB), [521](#)
- SCnSCB\_ICTR\_INTLINESNUM\_Msk
  - System Controls not in SCB (SCnSCB), [521](#)
- SCnSCB\_ICTR\_INTLINESNUM\_Pos
  - System Controls not in SCB (SCnSCB), [521](#)
- SCnSCB\_Type, [637](#)
  - ACTLR, [637](#)
  - ICTR, [637](#)
  - RESERVED0, [637](#)
- SEC\_AES\_DMA\_BUSY
  - error.h, [707](#)
- SEC\_AES\_NO\_ERROR
  - error.h, [707](#)
- SECSPERDAY
  - rtc\_ut.c, [797](#)
- SECSPERHOUR
  - rtc\_ut.c, [798](#)
- SECSPERMIN
  - rtc\_ut.c, [798](#)
- SEED
  - LPC\_CRC\_T, [574](#)
- SEQ\_CTRL
  - LPC\_ADC\_T, [571](#)
- SEQ\_GDAT
  - LPC\_ADC\_T, [572](#)
- SEQA\_CTRL
  - LPC\_ADC\_T, [572](#)
- SEQA\_GDAT
  - LPC\_ADC\_T, [572](#)
- SEQB\_CTRL
  - LPC\_ADC\_T, [572](#)
- SEQB\_GDAT
  - LPC\_ADC\_T, [572](#)
- SET
  - LPC Public Types, [480](#)
  - LPC\_GPIO\_T, [582](#)
  - LPC\_SCT\_T, [603](#)
- SETPORTPIN
  - CHIP: LPC5411X Enhanced boot block support, [148](#)
- SETTRIG
  - LPC\_DMA\_COMMON\_T, [578](#)
- SETVALID
  - LPC\_DMA\_COMMON\_T, [578](#)
- SHCSR
  - SCB\_Type, [636](#)
- SHP
  - SCB\_Type, [636](#)
- SIENF
  - LPC\_PIN\_INT\_T, [593](#)
- SIENR
  - LPC\_PIN\_INT\_T, [593](#)
- SL\_AUTO
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- SL\_I2C0
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- SL\_I2C1
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- SL\_I2C2
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- SL\_SPI0
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- SL\_SPI1
  - CHIP: LPC5411X Enhanced boot block support, [149](#)
- SLEEP\_CNT
  - DWT\_Type, [559](#)
- SLVADR
  - CHIP: LPC5411x I2C driver, [415](#)
- SLVCTL
  - CHIP: LPC5411x I2C driver, [415](#)
- SLVDAT
  - CHIP: LPC5411x I2C driver, [415](#)

- SLVQUAL0
  - CHIP: LPC5411x I2C driver, [415](#)
- SPI\_CFG\_BITMASK
  - CHIP: LPC5411X SPI driver, [272](#)
- SPI\_CFG\_CPHA\_FIRST
  - CHIP: LPC5411X SPI driver, [272](#)
- SPI\_CFG\_CPHA\_SECOND
  - CHIP: LPC5411X SPI driver, [272](#)
- SPI\_CFG\_CPOL\_HI
  - CHIP: LPC5411X SPI driver, [272](#)
- SPI\_CFG\_CPOL\_LO
  - CHIP: LPC5411X SPI driver, [272](#)
- SPI\_CFG\_LBM\_EN
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_CFG\_LSB\_FIRST\_EN
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_CFG\_MASTER\_EN
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_CFG\_MSB\_FIRST\_EN
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_CFG\_SLAVE\_EN
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_CFG\_SPI\_EN
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_CFG\_SPOL\_HI
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_CFG\_SPOL\_LO
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_CFG\_SPOLNUM\_HI
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_CFGSETUP\_T, [637](#)
  - lsbFirst, [638](#)
  - master, [638](#)
  - mode, [638](#)
  - reserved, [638](#)
- SPI\_CLOCK\_CPHA0\_CPOL0
  - CHIP: LPC5411X SPI driver, [280](#)
- SPI\_CLOCK\_CPHA0\_CPOL1
  - CHIP: LPC5411X SPI driver, [280](#)
- SPI\_CLOCK\_CPHA1\_CPOL0
  - CHIP: LPC5411X SPI driver, [280](#)
- SPI\_CLOCK\_CPHA1\_CPOL1
  - CHIP: LPC5411X SPI driver, [280](#)
- SPI\_CLOCK\_MODE0
  - CHIP: LPC5411X SPI driver, [280](#)
- SPI\_CLOCK\_MODE1
  - CHIP: LPC5411X SPI driver, [280](#)
- SPI\_CLOCK\_MODE2
  - CHIP: LPC5411X SPI driver, [280](#)
- SPI\_CLOCK\_MODE3
  - CHIP: LPC5411X SPI driver, [280](#)
- SPI\_CLOCK\_MODE\_T
  - CHIP: LPC5411X SPI driver, [280](#)
- SPI\_DIV\_VAL
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_DLY\_BITMASK
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_DLY\_FRAME\_DELAY
  - CHIP: LPC5411X SPI driver, [273](#)
- SPI\_DLY\_POST\_DELAY
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_DLY\_PRE\_DELAY
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_DLY\_TRANSFER\_DELAY
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_FIFO\_DEPTH
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_FIFOCFG\_DMARX
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_FIFOCFG\_DMATX
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_FIFOCFG\_EMPTYRX
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_FIFOCFG\_EMPTYTX
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_FIFOCFG\_ENABLERX
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_FIFOCFG\_ENABLETX
  - CHIP: LPC5411X SPI driver, [274](#)
- SPI\_FIFOCFG\_WAKERX
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOCFG\_WAKETX
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOINT\_BITMASK
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOINT\_PERINT
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOINT\_RXERR
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOINT\_RXLVL
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOINT\_TXERR
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOINT\_TXLVL
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOSTAT\_BITMASK
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOSTAT\_PERINT
  - CHIP: LPC5411X SPI driver, [275](#)
- SPI\_FIFOSTAT\_RXERR
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOSTAT\_RXFULL
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOSTAT\_RXLVL
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOSTAT\_RXNOTEMPTY
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOSTAT\_TXEMPTY
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOSTAT\_TXERR
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOSTAT\_TXLVL
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOSTAT\_TXNOTFULL
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOTRIG\_BITMASK

- CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOTRIG\_RXLVL
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOTRIG\_RXLVL\_DEFAULT
  - CHIP: LPC5411X SPI driver, [276](#)
- SPI\_FIFOTRIG\_RXLVLENA
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_FIFOTRIG\_TXLVL
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_FIFOTRIG\_TXLVL\_DEFAULT
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_FIFOTRIG\_TXLVLENA
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_INT\_BITMASK
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_INT\_MSTIDLE
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_INT\_SSAEN
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_INT\_SSDEN
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_RXDAT\_BITMASK
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_RXDAT\_DATA
  - CHIP: LPC5411X SPI driver, [277](#)
- SPI\_RXDAT\_RXSSELN
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_RXDAT\_RXSSELN\_ACTIVE
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_RXDAT\_SOT
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_STAT\_BITMASK
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_STAT\_EOT
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_STAT\_MSTIDLE
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_STAT\_SSA
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_STAT\_SSD
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_STAT\_STALLED
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_TXDAT\_ASSERT\_SSEL
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_TXDAT\_ASSERTNUM\_SSEL
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_TXDAT\_BITMASK
  - CHIP: LPC5411X SPI driver, [278](#)
- SPI\_TXDAT\_CTRLMASK
  - CHIP: LPC5411X SPI driver, [279](#)
- SPI\_TXDAT\_DATA
  - CHIP: LPC5411X SPI driver, [279](#)
- SPI\_TXDAT\_DEASSERT\_ALL
  - CHIP: LPC5411X SPI driver, [279](#)
- SPI\_TXDAT\_DEASSERT\_SSEL
  - CHIP: LPC5411X SPI driver, [279](#)
- SPI\_TXDAT\_DEASSERTNUM\_SSEL
  - CHIP: LPC5411X SPI driver, [279](#)
- CHIP: LPC5411X SPI driver, [279](#)
- SPI\_TXDAT\_EOF
  - CHIP: LPC5411X SPI driver, [279](#)
- SPI\_TXDAT\_EOT
  - CHIP: LPC5411X SPI driver, [279](#)
- SPI\_TXDAT\_FLEN
  - CHIP: LPC5411X SPI driver, [279](#)
- SPI\_TXDAT\_FLENMASK
  - CHIP: LPC5411X SPI driver, [279](#)
- SPI\_TXDAT\_RXIGNORE
  - CHIP: LPC5411X SPI driver, [279](#)
- SPI\_XFER\_STATE\_BUSY
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPI\_XFER\_STATE\_DONE
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPI\_XFER\_STATE\_ERROR
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPI\_XFER\_STATE\_IDLE
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPI\_XFER\_STATE\_STALL
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIFI\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- SPIFICKLDIV
  - LPC\_SYSCON\_T, [616](#)
- SPIFICKLSEL
  - LPC\_SYSCON\_T, [616](#)
- SPIM\_DELAY\_CONFIG\_T, [638](#)
  - FrameDelay, [639](#)
  - PostDelay, [639](#)
  - PreDelay, [639](#)
  - TransferDelay, [639](#)
- SPIM\_EVENT\_DONE
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_EVENT\_ERROR
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_EVENT\_ERRORRX
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_EVENT\_ERRORTX
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_EVENT\_T
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_EVENT\_WAIT
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_XFER\_OPT\_DMA
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_XFER\_OPT\_FRAME\_ASSERT
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_XFER\_OPT\_FRAME\_DLY
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_XFER\_STATE\_T
  - CHIP: LPC5411X SPI master driver, [295](#)
- SPIM\_XFER\_T, [639](#)
  - dataWidth, [640](#)
  - eventCB, [640](#)
  - option, [640](#)
  - rxCount, [640](#)

- rxData, [640](#)
- rxIndex, [640](#)
- sselNum, [640](#)
- state, [640](#)
- txCount, [640](#)
- txData, [641](#)
- txIndex, [641](#)
- usrData, [641](#)
- SPIS\_EVENT\_DONE
  - CHIP: LPC5411X SPI slave driver, [300](#)
- SPIS\_EVENT\_ERRORRX
  - CHIP: LPC5411X SPI slave driver, [300](#)
- SPIS\_EVENT\_ERRORTX
  - CHIP: LPC5411X SPI slave driver, [300](#)
- SPIS\_EVENT\_SASSERT
  - CHIP: LPC5411X SPI slave driver, [300](#)
- SPIS\_EVENT\_SDEASSERT
  - CHIP: LPC5411X SPI slave driver, [300](#)
- SPIS\_EVENT\_T
  - CHIP: LPC5411X SPI slave driver, [300](#)
- SPIS\_EVENT\_THRESHOLD
  - CHIP: LPC5411X SPI slave driver, [300](#)
- SPIS\_XFER\_T, [641](#)
  - dataWidth, [642](#)
  - eventCB, [642](#)
  - rxCount, [642](#)
  - rxData, [642](#)
  - rxIndex, [642](#)
  - ss\_count, [642](#)
  - ss\_state, [642](#)
  - sselNum, [642](#)
  - thresCount, [642](#)
  - txCount, [642](#)
  - txData, [643](#)
  - txIndex, [643](#)
- SPPR
  - TPI\_Type, [647](#)
- SPSEL
  - CONTROL\_Type, [552](#)
- SRAMBASE
  - LPC\_DMA\_T, [580](#)
- SS\_MC\_MAXC
  - CHIP: LPC5411X PLL Driver, [227](#)
- SS\_MC\_NOC
  - CHIP: LPC5411X PLL Driver, [227](#)
- SS\_MC\_RECC
  - CHIP: LPC5411X PLL Driver, [227](#)
- SS\_MF\_128
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MF\_16
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MF\_24
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MF\_256
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MF\_32
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MF\_384
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MF\_512
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MF\_64
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MODWVCTRL\_T
  - CHIP: LPC5411X PLL Driver, [227](#)
- SS\_MR\_K0
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MR\_K1
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MR\_K1\_5
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MR\_K2
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MR\_K3
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MR\_K4
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MR\_K6
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_MR\_K8
  - CHIP: LPC5411X PLL Driver, [228](#)
- SS\_PROGMODDP\_T
  - CHIP: LPC5411X PLL Driver, [227](#)
- SS\_PROGMODFM\_T
  - CHIP: LPC5411X PLL Driver, [228](#)
- SSPSR
  - TPI\_Type, [647](#)
- START\_H
  - LPC\_SCT\_T, [603](#)
- START\_L
  - LPC\_SCT\_T, [603](#)
- STARTERP
  - LPC\_SYSCON\_T, [617](#)
- STARTERPCLR
  - LPC\_SYSCON\_T, [617](#)
- STARTERPSET
  - LPC\_SYSCON\_T, [617](#)
- STARTUP
  - LPC\_ADC\_T, [572](#)
- STAT
  - CHIP: LPC5411x I2C driver, [416](#)
  - LPC\_I2S\_T, [586](#)
  - LPC\_MRT\_CH\_T, [590](#)
  - LPC\_SPI\_T, [606](#)
  - LPC\_USART\_T, [623](#)
- STATE
  - LPC\_SCT\_T, [603](#)
- STATE\_H
  - LPC\_SCT\_T, [603](#)
- STATE\_L
  - LPC\_SCT\_T, [603](#)
- STATIC
  - LPC Public Macros, [475](#)
- STATUS
  - LPC\_UTICK\_T, [624](#)
- STEREO\_SIDE\_T

- CHIP: LPC5411X DMIC driver, [138](#)
- STIR
  - NVIC\_Type, [627](#)
- STOP\_H
  - LPC\_SCT\_T, [603](#)
- STOP\_L
  - LPC\_SCT\_T, [603](#)
- SUCCESS
  - LPC Public Types, [480](#)
- SUM
  - LPC\_CRC\_T, [574](#)
- SVC\_Handler
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [808](#)
- SVCall\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- SYS\_PLL\_BANDSEL
  - pll\_5411x.c, [792](#)
- SYS\_PLL\_BYPASS
  - pll\_5411x.c, [792](#)
- SYS\_PLL\_BYPASSCCODIV2
  - pll\_5411x.c, [792](#)
- SYS\_PLL\_DIRECTI
  - pll\_5411x.c, [792](#)
- SYS\_PLL\_DIRECTO
  - pll\_5411x.c, [792](#)
- SYS\_PLL\_SEL1
  - pll\_5411x.c, [792](#)
- SYS\_PLL\_SEL2
  - pll\_5411x.c, [792](#)
- SYS\_PLL\_SEL3
  - pll\_5411x.c, [793](#)
- SYS\_PLL\_UPLIMOFF
  - pll\_5411x.c, [793](#)
- SYSCON\_ADCCLKSEL SRC\_FROHF
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_ADCCLKSEL SRC\_MAINCLK
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_ADCCLKSEL SRC\_SYSPLLOUT
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_ASYNC\_FRO12MHZ
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_ASYNC\_MAINCLK
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_AUTO CGOR\_MASK
  - CHIP: LPC5411X System and Control Driver, [326](#)
- SYSCON\_AUTO CGOR\_RAM0X
  - CHIP: LPC5411X System and Control Driver, [326](#)
- SYSCON\_AUTO CGOR\_RAM1
  - CHIP: LPC5411X System and Control Driver, [327](#)
- SYSCON\_AUTO CGOR\_RAM2
  - CHIP: LPC5411X System and Control Driver, [327](#)
- SYSCON\_CLKOUTSRC\_CLKIN
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_CLKOUTSRC\_DISABLED
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_CLKOUTSRC\_FRO12MHZ
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_CLKOUTSRC\_FROHF
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_CLKOUTSRC\_MAINCLK
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_CLKOUTSRC\_PLL
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_CLKOUTSRC\_RTC
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_CLKOUTSRC\_WDTOSC
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSCON\_CLOCK\_ADC0
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_CRC
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_DMA
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_DMIC
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FLASH
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FLEXCOMM0
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FLEXCOMM1
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FLEXCOMM2
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FLEXCOMM3
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FLEXCOMM4
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FLEXCOMM5
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FLEXCOMM6
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FLEXCOMM7
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_FMC
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_GINT
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_GPIO0
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_GPIO1
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_INPUTMUX
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_IOCON
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_MAILBOX
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_MRT
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_PINT
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_ROM

- CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_RTC
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_SCT0
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_SPIFI
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_SRAM1
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_SRAM2
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_SRAMX
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_TIMER0
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_TIMER1
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_TIMER2
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_TIMER3
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_TIMER4
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_USB
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_UTICK
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_CLOCK\_WWDT
  - CHIP: LPC5411X Clock Driver, [79](#)
- SYSCON\_FLASH\_1CYCLE
  - CHIP: LPC5411X System and Control Driver, [333](#)
- SYSCON\_FLASH\_2CYCLE
  - CHIP: LPC5411X System and Control Driver, [333](#)
- SYSCON\_FLASH\_3CYCLE
  - CHIP: LPC5411X System and Control Driver, [333](#)
- SYSCON\_FLASH\_4CYCLE
  - CHIP: LPC5411X System and Control Driver, [333](#)
- SYSCON\_FLASH\_5CYCLE
  - CHIP: LPC5411X System and Control Driver, [333](#)
- SYSCON\_FLASH\_6CYCLE
  - CHIP: LPC5411X System and Control Driver, [333](#)
- SYSCON\_FLASH\_7CYCLE
  - CHIP: LPC5411X System and Control Driver, [333](#)
- SYSCON\_FLASH\_8CYCLE
  - CHIP: LPC5411X System and Control Driver, [333](#)
- SYSCON\_FLASHTIM\_T
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_FLEXCOMMCLKSELSRC\_FRG
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FLEXCOMMCLKSELSRC\_FRO12MHZ
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FLEXCOMMCLKSELSRC\_FROHF
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FLEXCOMMCLKSELSRC\_MCLK
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FLEXCOMMCLKSELSRC\_NONE
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FLEXCOMMCLKSELSRC\_PLL
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FRGCLKSRC\_FRO12MHZ
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FRGCLKSRC\_FROHF
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FRGCLKSRC\_MAINCLK
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FRGCLKSRC\_NONE
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FRGCLKSRC\_PLL
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_FRO12MHZ\_FREQ
  - CHIP: LPC5411X Clock Driver, [77](#)
- SYSCON\_FRO48MHZ\_FREQ
  - CHIP: LPC5411X Clock Driver, [77](#)
- SYSCON\_FRO96MHZ\_FREQ
  - CHIP: LPC5411X Clock Driver, [77](#)
- SYSCON\_FROCTRL\_HSPDCLK
  - CHIP: LPC5411X System and Control Driver, [327](#)
- SYSCON\_FROCTRL\_MASK
  - CHIP: LPC5411X System and Control Driver, [327](#)
- SYSCON\_FROCTRL\_SEL96MHZ
  - CHIP: LPC5411X System and Control Driver, [327](#)
- SYSCON\_FROCTRL\_USBCLKADJ
  - CHIP: LPC5411X System and Control Driver, [327](#)
- SYSCON\_FROCTRL\_USBMODCHG
  - CHIP: LPC5411X System and Control Driver, [327](#)
- SYSCON\_FROCTRL\_WRTRIM
  - CHIP: LPC5411X System and Control Driver, [327](#)
- SYSCON\_MAIN\_A\_CLKSRC\_FRO12MHZ
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_MAIN\_A\_CLKSRCA\_CLKIN
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_MAIN\_A\_CLKSRCA\_FROHF
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_MAIN\_A\_CLKSRCA\_WDTOSC
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_MAIN\_B\_CLKSRC\_MAINCLKSELA
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_MAIN\_B\_CLKSRC\_PLL
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_MAIN\_B\_CLKSRC\_RTC
  - CHIP: LPC5411X Clock Driver, [80](#)
- SYSCON\_MAINCLKSRC\_CLKIN
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_MAINCLKSRC\_FRO12MHZ
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_MAINCLKSRC\_FROHF
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_MAINCLKSRC\_PLLOUT
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_MAINCLKSRC\_RTC
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_MAINCLKSRC\_WDTOSC
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_MCLKSRC\_DISABLED
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_MCLKSRC\_FROHF
  - CHIP: LPC5411X Clock Driver, [81](#)



- CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_MCLKSRC\_MCLKIN
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_MCLKSRC\_PLL
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_NMISRC\_M0\_ENABLE
  - CHIP: LPC5411X System and Control Driver, [327](#)
- SYSCON\_NMISRC\_M4\_ENABLE
  - CHIP: LPC5411X System and Control Driver, [328](#)
- SYSCON\_PDRUNCFG\_LP\_VDDFLASH
  - CHIP: LPC5411X System and Control Driver, [328](#)
- SYSCON\_PDRUNCFG\_PD\_ADC0
  - CHIP: LPC5411X System and Control Driver, [328](#)
- SYSCON\_PDRUNCFG\_PD\_BOD\_INTR
  - CHIP: LPC5411X System and Control Driver, [328](#)
- SYSCON\_PDRUNCFG\_PD\_BOD\_RST
  - CHIP: LPC5411X System and Control Driver, [328](#)
- SYSCON\_PDRUNCFG\_PD\_FLASH
  - CHIP: LPC5411X System and Control Driver, [328](#)
- SYSCON\_PDRUNCFG\_PD\_FRO
  - CHIP: LPC5411X System and Control Driver, [328](#)
- SYSCON\_PDRUNCFG\_PD\_ROM
  - CHIP: LPC5411X System and Control Driver, [328](#)
- SYSCON\_PDRUNCFG\_PD\_SRAM0
  - CHIP: LPC5411X System and Control Driver, [328](#)
- SYSCON\_PDRUNCFG\_PD\_SRAM1
  - CHIP: LPC5411X System and Control Driver, [329](#)
- SYSCON\_PDRUNCFG\_PD\_SRAM2
  - CHIP: LPC5411X System and Control Driver, [329](#)
- SYSCON\_PDRUNCFG\_PD\_SRAMX
  - CHIP: LPC5411X System and Control Driver, [329](#)
- SYSCON\_PDRUNCFG\_PD\_SYS\_PLL
  - CHIP: LPC5411X System and Control Driver, [329](#)
- SYSCON\_PDRUNCFG\_PD\_TS
  - CHIP: LPC5411X System and Control Driver, [329](#)
- SYSCON\_PDRUNCFG\_PD\_USB\_PHY
  - CHIP: LPC5411X System and Control Driver, [329](#)
- SYSCON\_PDRUNCFG\_PD\_VDDA\_ENA
  - CHIP: LPC5411X System and Control Driver, [329](#)
- SYSCON\_PDRUNCFG\_PD\_VDDFLASH
  - CHIP: LPC5411X System and Control Driver, [329](#)
- SYSCON\_PDRUNCFG\_PD\_VDDHV\_ENA
  - CHIP: LPC5411X System and Control Driver, [329](#)
- SYSCON\_PDRUNCFG\_PD\_VREFP
  - CHIP: LPC5411X System and Control Driver, [330](#)
- SYSCON\_PDRUNCFG\_PD\_WDT\_OSC
  - CHIP: LPC5411X System and Control Driver, [330](#)
- SYSCON\_PLLCLKSRC\_CLKIN
  - CHIP: LPC5411X PLL Driver, [227](#)
- SYSCON\_PLLCLKSRC\_DISABLED
  - CHIP: LPC5411X PLL Driver, [227](#)
- SYSCON\_PLLCLKSRC\_FRO12MHZ
  - CHIP: LPC5411X PLL Driver, [227](#)
- SYSCON\_PLLCLKSRC\_RTC
  - CHIP: LPC5411X PLL Driver, [227](#)
- SYSCON\_PLLCLKSRC\_WDT
  - CHIP: LPC5411X PLL Driver, [227](#)
- SYSCON\_RST\_BOD
  - CHIP: LPC5411X System and Control Driver, [330](#)
- SYSCON\_RST\_EXTRST
  - CHIP: LPC5411X System and Control Driver, [330](#)
- SYSCON\_RST\_POR
  - CHIP: LPC5411X System and Control Driver, [330](#)
- SYSCON\_RST\_SYSRST
  - CHIP: LPC5411X System and Control Driver, [330](#)
- SYSCON\_RST\_WDT
  - CHIP: LPC5411X System and Control Driver, [330](#)
- SYSCON\_RTC\_FREQ
  - CHIP: LPC5411X Clock Driver, [77](#)
- SYSCON\_STARTER\_ADC0\_SEQA
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_ADC0\_SEQB
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_ADC0\_THCMP
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_DMA
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_DMIC
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_FLEXCOMM0
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_FLEXCOMM1
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_FLEXCOMM2
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_FLEXCOMM3
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_FLEXCOMM4
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_FLEXCOMM5
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_FLEXCOMM6
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_FLEXCOMM7
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_GINT0
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_GINT1
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_HWVAD
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_MAILBOX
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_MRT
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_PINT0
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_PINT1
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_PINT2
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_PINT3
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_PINT4
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_PINT5
  - CHIP: LPC5411X System and Control Driver, [332](#)

- CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_PINT6
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_PINT7
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_RESERVED0
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_RTC
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_SCT0
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_TIMER0
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_TIMER1
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_TIMER2
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_TIMER3
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_TIMER4
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_USB
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_USBNEEDCLK
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_UTICK
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_STARTER\_WWDT\_BOD
  - CHIP: LPC5411X System and Control Driver, [332](#)
- SYSCON\_SYSPLLCTRL\_BANDSEL\_SSCGREG\_N
  - pll\_5411x.c, [793](#)
- SYSCON\_SYSPLLCTRL\_BANDSEL\_SSCGREG\_N←\_P
  - pll\_5411x.c, [793](#)
- SYSCON\_SYSPLLCTRL\_BYPASS
  - pll\_5411x.c, [793](#)
- SYSCON\_SYSPLLCTRL\_BYPASS\_FBDIV2
  - pll\_5411x.c, [793](#)
- SYSCON\_SYSPLLCTRL\_BYPASS\_FBDIV2\_P
  - pll\_5411x.c, [793](#)
- SYSCON\_SYSPLLCTRL\_BYPASS\_P
  - pll\_5411x.c, [793](#)
- SYSCON\_SYSPLLCTRL\_DIRECTI
  - pll\_5411x.c, [793](#)
- SYSCON\_SYSPLLCTRL\_DIRECTI\_P
  - pll\_5411x.c, [793](#)
- SYSCON\_SYSPLLCTRL\_DIRECTO
  - pll\_5411x.c, [793](#)
- SYSCON\_SYSPLLCTRL\_DIRECTO\_P
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLCTRL\_SEL\_I\_M
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLCTRL\_SEL\_I\_P
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLCTRL\_SEL\_P\_M
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLCTRL\_SEL\_P\_P
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLCTRL\_SEL\_R\_M
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLCTRL\_SEL\_R\_P
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLCTRL\_UPLIMOFF
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLCTRL\_UPLIMOFF\_P
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLSTAT\_LOCK
  - pll\_5411x.c, [794](#)
- SYSCON\_SYSPLLSTAT\_LOCK\_P
  - pll\_5411x.c, [794](#)
- SYSCON\_USBCLKSRC\_DISABLED
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_USBCLKSRC\_FROHF
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_USBCLKSRC\_PLL
  - CHIP: LPC5411X Clock Driver, [81](#)
- SYSCON\_WDTOSC\_FREQ
  - CHIP: LPC5411X Clock Driver, [78](#)
- SYSMEMREMAP
  - LPC\_SYSCON\_T, [617](#)
- SYSPLLCLKSEL
  - LPC\_SYSCON\_T, [617](#)
- SYSPLLCTRL
  - LPC\_SYSCON\_T, [617](#)
  - PLL\_SETUP\_T, [631](#)
- SYSPLLNDEC
  - LPC\_SYSCON\_T, [617](#)
  - PLL\_SETUP\_T, [632](#)
- SYSPLLPDEC
  - LPC\_SYSCON\_T, [617](#)
  - PLL\_SETUP\_T, [632](#)
- SYSPLLSSCTRL
  - LPC\_SYSCON\_T, [617](#)
  - PLL\_SETUP\_T, [632](#)
- SYSPLLSTAT
  - LPC\_SYSCON\_T, [618](#)
- SYSRSTSTAT
  - LPC\_SYSCON\_T, [618](#)
- SYSTCKCAL
  - LPC\_SYSCON\_T, [618](#)
- SYSTICKCLKDIV
  - LPC\_SYSCON\_T, [618](#)
- SetState
  - LPC Public Types, [479](#)
- setupFlashClocks
  - sysinit\_5411x.c, [802](#)
- side
  - CHIP: LPC5411X DMIC driver, [146](#)
- slaveAddr
  - I2CM\_XFER\_T, [560](#)
- slaveDone
  - I2CS\_XFER\_T, [561](#)
- slaveRecv
  - I2CS\_XFER\_T, [561](#)
- slaveSend
  - I2CS\_XFER\_T, [561](#)



- slaveStart
  - I2CS\_XFER\_T, [561](#)
- source
  - DMA\_CHDESC\_T, [554](#)
- spim\_5411x.c
  - get\_tx\_data, [799](#)
  - put\_rx\_data, [799](#)
  - spim\_do\_tsr, [800](#)
  - spim\_get\_xfercfg, [800](#)
- spim\_do\_tsr
  - spim\_5411x.c, [800](#)
- spim\_get\_xfercfg
  - spim\_5411x.c, [800](#)
- spis\_5411x.c
  - get\_tx\_data, [800](#)
  - put\_rx\_data, [800](#)
  - spis\_do\_tsr, [800](#)
- spis\_do\_tsr
  - spis\_5411x.c, [800](#)
- src\_increment
  - CHIP: LPC5411X DMA Service driver, [132](#)
- ss\_count
  - SPIS\_XFER\_T, [642](#)
- ss\_mc
  - PLL\_CONFIG\_T, [630](#)
- ss\_mf
  - PLL\_CONFIG\_T, [630](#)
- ss\_mr
  - PLL\_CONFIG\_T, [631](#)
- ss\_state
  - SPIS\_XFER\_T, [642](#)
- sselNum
  - SPIM\_XFER\_T, [640](#)
  - SPIS\_XFER\_T, [642](#)
- state
  - SPIM\_XFER\_T, [640](#)
- Status
  - LPC Public Types, [480](#)
- status
  - I2CM\_XFER\_T, [560](#)
- Status and Control Registers, [495](#)
- StopWatch\_DelayMs
  - CHIP: Stopwatch primitives., [424](#)
- StopWatch\_DelayTicks
  - CHIP: Stopwatch primitives., [424](#)
- StopWatch\_DelayUs
  - CHIP: Stopwatch primitives., [425](#)
- StopWatch\_Elapsed
  - CHIP: Stopwatch primitives., [425](#)
- StopWatch\_Init
  - CHIP: Stopwatch primitives., [425](#)
- StopWatch\_MsToTicks
  - CHIP: Stopwatch primitives., [425](#)
- StopWatch\_Start
  - CHIP: Stopwatch primitives., [426](#)
- StopWatch\_TicksPerSecond
  - CHIP: Stopwatch primitives., [426](#)
- StopWatch\_TicksToMs
  - CHIP: Stopwatch primitives., [426](#)
- StopWatch\_TicksToUs
  - CHIP: Stopwatch primitives., [426](#)
- StopWatch\_UsToTicks
  - CHIP: Stopwatch primitives., [426](#)
- stopwatch\_5411x.c
  - LPC\_TIMER32\_1, [801](#)
  - ticksPerMs, [801](#)
  - ticksPerSecond, [801](#)
  - ticksPerUs, [801](#)
- Supported toolchains in LPCOpen, [496](#)
- SysTick
  - Core Definitions, [453](#)
- SysTick Functions, [497](#)
  - SysTick\_Config, [497](#)
- SysTick\_BASE
  - Core Definitions, [453](#)
- SysTick\_CALIB\_NOREF\_Msk
  - System Tick Timer (SysTick), [523](#)
- SysTick\_CALIB\_NOREF\_Pos
  - System Tick Timer (SysTick), [523](#)
- SysTick\_CALIB\_SKEW\_Msk
  - System Tick Timer (SysTick), [523](#)
- SysTick\_CALIB\_SKEW\_Pos
  - System Tick Timer (SysTick), [523](#)
- SysTick\_CALIB\_TENMS\_Msk
  - System Tick Timer (SysTick), [523](#)
- SysTick\_CALIB\_TENMS\_Pos
  - System Tick Timer (SysTick), [524](#)
- SysTick\_CTRL\_CLKSOURCE\_Msk
  - System Tick Timer (SysTick), [524](#)
- SysTick\_CTRL\_CLKSOURCE\_Pos
  - System Tick Timer (SysTick), [524](#)
- SysTick\_CTRL\_COUNTFLAG\_Msk
  - System Tick Timer (SysTick), [524](#)
- SysTick\_CTRL\_COUNTFLAG\_Pos
  - System Tick Timer (SysTick), [524](#), [525](#)
- SysTick\_CTRL\_ENABLE\_Msk
  - System Tick Timer (SysTick), [525](#)
- SysTick\_CTRL\_ENABLE\_Pos
  - System Tick Timer (SysTick), [525](#)
- SysTick\_CTRL\_TICKINT\_Msk
  - System Tick Timer (SysTick), [525](#)
- SysTick\_CTRL\_TICKINT\_Pos
  - System Tick Timer (SysTick), [525](#)
- SysTick\_Config
  - SysTick Functions, [497](#)
- SysTick\_Handler
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [808](#)
- SysTick\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- SysTick\_LOAD\_RELOAD\_Msk
  - System Tick Timer (SysTick), [526](#)
- SysTick\_LOAD\_RELOAD\_Pos

- System Tick Timer (SysTick), [526](#)
- SysTick\_Type, [643](#)
  - CALIB, [643](#)
  - CTRL, [643](#)
  - LOAD, [643](#)
  - VAL, [644](#)
- SysTick\_VAL\_CURRENT\_Msk
  - System Tick Timer (SysTick), [526](#)
- SysTick\_VAL\_CURRENT\_Pos
  - System Tick Timer (SysTick), [526](#)
- sysinit.c
  - SystemInit, [808](#)
- sysinit\_5411x.c
  - setupFlashClocks, [802](#)
- System Control Block (SCB), [498](#)
  - SCB\_AIRCR\_ENDIANESS\_Msk, [501](#)
  - SCB\_AIRCR\_ENDIANESS\_Pos, [501](#)
  - SCB\_AIRCR\_PRIGROUP\_Msk, [501](#)
  - SCB\_AIRCR\_PRIGROUP\_Pos, [501](#)
  - SCB\_AIRCR\_SYSRESETREQ\_Msk, [501](#), [502](#)
  - SCB\_AIRCR\_SYSRESETREQ\_Pos, [502](#)
  - SCB\_AIRCR\_VECTCLRACTIVE\_Msk, [502](#)
  - SCB\_AIRCR\_VECTCLRACTIVE\_Pos, [502](#)
  - SCB\_AIRCR\_VECTKEY\_Msk, [502](#)
  - SCB\_AIRCR\_VECTKEY\_Pos, [503](#)
  - SCB\_AIRCR\_VECTKEYSTAT\_Msk, [503](#)
  - SCB\_AIRCR\_VECTKEYSTAT\_Pos, [503](#)
  - SCB\_AIRCR\_VECTRESET\_Msk, [503](#)
  - SCB\_AIRCR\_VECTRESET\_Pos, [503](#)
  - SCB\_CCR\_BFHFNMIGN\_Msk, [503](#)
  - SCB\_CCR\_BFHFNMIGN\_Pos, [504](#)
  - SCB\_CCR\_DIV\_0\_TRP\_Msk, [504](#)
  - SCB\_CCR\_DIV\_0\_TRP\_Pos, [504](#)
  - SCB\_CCR\_NONBASETHRDENA\_Msk, [504](#)
  - SCB\_CCR\_NONBASETHRDENA\_Pos, [504](#)
  - SCB\_CCR\_STKALIGN\_Msk, [504](#)
  - SCB\_CCR\_STKALIGN\_Pos, [504](#)
  - SCB\_CCR\_UNALIGN\_TRP\_Msk, [505](#)
  - SCB\_CCR\_UNALIGN\_TRP\_Pos, [505](#)
  - SCB\_CCR\_USERSETMPEND\_Msk, [505](#)
  - SCB\_CCR\_USERSETMPEND\_Pos, [505](#)
  - SCB\_CFSR\_BUSFAULTSR\_Msk, [505](#)
  - SCB\_CFSR\_BUSFAULTSR\_Pos, [505](#)
  - SCB\_CFSR\_MEMFAULTSR\_Msk, [505](#)
  - SCB\_CFSR\_MEMFAULTSR\_Pos, [506](#)
  - SCB\_CFSR\_USGFAULTSR\_Msk, [506](#)
  - SCB\_CFSR\_USGFAULTSR\_Pos, [506](#)
  - SCB\_CPUID\_ARCHITECTURE\_Msk, [506](#)
  - SCB\_CPUID\_ARCHITECTURE\_Pos, [506](#)
  - SCB\_CPUID\_IMPLEMENTER\_Msk, [506](#)
  - SCB\_CPUID\_IMPLEMENTER\_Pos, [507](#)
  - SCB\_CPUID\_PARTNO\_Msk, [507](#)
  - SCB\_CPUID\_PARTNO\_Pos, [507](#)
  - SCB\_CPUID\_REVISION\_Msk, [507](#)
  - SCB\_CPUID\_REVISION\_Pos, [507](#), [508](#)
  - SCB\_CPUID\_VARIANT\_Msk, [508](#)
  - SCB\_CPUID\_VARIANT\_Pos, [508](#)
  - SCB\_DFSR\_BKPT\_Msk, [508](#)
  - SCB\_DFSR\_BKPT\_Pos, [508](#)
  - SCB\_DFSR\_DWTTRAP\_Msk, [508](#)
  - SCB\_DFSR\_DWTTRAP\_Pos, [508](#)
  - SCB\_DFSR\_EXTERNAL\_Msk, [509](#)
  - SCB\_DFSR\_EXTERNAL\_Pos, [509](#)
  - SCB\_DFSR\_HALTED\_Msk, [509](#)
  - SCB\_DFSR\_HALTED\_Pos, [509](#)
  - SCB\_DFSR\_VCATCH\_Msk, [509](#)
  - SCB\_DFSR\_VCATCH\_Pos, [509](#)
  - SCB\_HFSR\_DEBUGEVT\_Msk, [509](#)
  - SCB\_HFSR\_DEBUGEVT\_Pos, [509](#)
  - SCB\_HFSR\_FORCED\_Msk, [509](#)
  - SCB\_HFSR\_FORCED\_Pos, [510](#)
  - SCB\_HFSR\_VECTTBL\_Msk, [510](#)
  - SCB\_HFSR\_VECTTBL\_Pos, [510](#)
  - SCB\_ICSR\_ISRPENDING\_Msk, [510](#)
  - SCB\_ICSR\_ISRPENDING\_Pos, [510](#)
  - SCB\_ICSR\_ISRPREEMPT\_Msk, [510](#)
  - SCB\_ICSR\_ISRPREEMPT\_Pos, [511](#)
  - SCB\_ICSR\_NMIPENDSET\_Msk, [511](#)
  - SCB\_ICSR\_NMIPENDSET\_Pos, [511](#)
  - SCB\_ICSR\_PENDSTCLR\_Msk, [511](#)
  - SCB\_ICSR\_PENDSTCLR\_Pos, [511](#), [512](#)
  - SCB\_ICSR\_PENDSTSET\_Msk, [512](#)
  - SCB\_ICSR\_PENDSTSET\_Pos, [512](#)
  - SCB\_ICSR\_PENDSVCLR\_Msk, [512](#)
  - SCB\_ICSR\_PENDSVCLR\_Pos, [512](#)
  - SCB\_ICSR\_PENDSVSET\_Msk, [513](#)
  - SCB\_ICSR\_PENDSVSET\_Pos, [513](#)
  - SCB\_ICSR\_RETTOBASE\_Msk, [513](#)
  - SCB\_ICSR\_RETTOBASE\_Pos, [513](#)
  - SCB\_ICSR\_VECTACTIVE\_Msk, [513](#)
  - SCB\_ICSR\_VECTACTIVE\_Pos, [513](#), [514](#)
  - SCB\_ICSR\_VECTPENDING\_Msk, [514](#)
  - SCB\_ICSR\_VECTPENDING\_Pos, [514](#)
  - SCB\_SCR\_SEVONPEND\_Msk, [514](#)
  - SCB\_SCR\_SEVONPEND\_Pos, [514](#)
  - SCB\_SCR\_SLEEPDEEP\_Msk, [515](#)
  - SCB\_SCR\_SLEEPDEEP\_Pos, [515](#)
  - SCB\_SCR\_SLEEPONEXIT\_Msk, [515](#)
  - SCB\_SCR\_SLEEPONEXIT\_Pos, [515](#)
  - SCB\_SHCSR\_BUSFAULTACT\_Msk, [515](#)
  - SCB\_SHCSR\_BUSFAULTACT\_Pos, [516](#)
  - SCB\_SHCSR\_BUSFAULTENA\_Msk, [516](#)
  - SCB\_SHCSR\_BUSFAULTENA\_Pos, [516](#)
  - SCB\_SHCSR\_BUSFAULTPENDE\_Msk, [516](#)
  - SCB\_SHCSR\_BUSFAULTPENDE\_Pos, [516](#)
  - SCB\_SHCSR\_MEMFAULTACT\_Msk, [516](#)
  - SCB\_SHCSR\_MEMFAULTACT\_Pos, [516](#)
  - SCB\_SHCSR\_MEMFAULTENA\_Msk, [516](#)
  - SCB\_SHCSR\_MEMFAULTENA\_Pos, [516](#)
  - SCB\_SHCSR\_MEMFAULTPENDE\_Msk, [517](#)
  - SCB\_SHCSR\_MEMFAULTPENDE\_Pos, [517](#)
  - SCB\_SHCSR\_MONITORACT\_Msk, [517](#)
  - SCB\_SHCSR\_MONITORACT\_Pos, [517](#)
  - SCB\_SHCSR\_PENDSVACT\_Msk, [517](#)
  - SCB\_SHCSR\_PENDSVACT\_Pos, [517](#)
  - SCB\_SHCSR\_SVCALLACT\_Msk, [517](#)

- SCB\_SHCSR\_SVCALLACT\_Pos, [517](#)
- SCB\_SHCSR\_SVCALLPENDEDED\_Msk, [517](#), [518](#)
- SCB\_SHCSR\_SVCALLPENDEDED\_Pos, [518](#)
- SCB\_SHCSR\_SYSTICKACT\_Msk, [518](#)
- SCB\_SHCSR\_SYSTICKACT\_Pos, [518](#)
- SCB\_SHCSR\_USGFAULTACT\_Msk, [518](#)
- SCB\_SHCSR\_USGFAULTACT\_Pos, [518](#)
- SCB\_SHCSR\_USGFAULTENA\_Msk, [518](#)
- SCB\_SHCSR\_USGFAULTENA\_Pos, [518](#)
- SCB\_SHCSR\_USGFAULTPENDEDED\_Msk, [519](#)
- SCB\_SHCSR\_USGFAULTPENDEDED\_Pos, [519](#)
- SCB\_VTOR\_TBLOFF\_Msk, [519](#)
- SCB\_VTOR\_TBLOFF\_Pos, [519](#)
- System Controls not in SCB (SCnSCB), [520](#)
  - SCnSCB\_ACTLR\_DISDEFWBUF\_Msk, [520](#)
  - SCnSCB\_ACTLR\_DISDEFWBUF\_Pos, [520](#)
  - SCnSCB\_ACTLR\_DISFOLD\_Msk, [520](#)
  - SCnSCB\_ACTLR\_DISFOLD\_Pos, [520](#)
  - SCnSCB\_ACTLR\_DISFPCA\_Msk, [520](#)
  - SCnSCB\_ACTLR\_DISFPCA\_Pos, [521](#)
  - SCnSCB\_ACTLR\_DISMCYCINT\_Msk, [521](#)
  - SCnSCB\_ACTLR\_DISMCYCINT\_Pos, [521](#)
  - SCnSCB\_ACTLR\_DISOFP\_Msk, [521](#)
  - SCnSCB\_ACTLR\_DISOFP\_Pos, [521](#)
  - SCnSCB\_ICTR\_INTLINESNUM\_Msk, [521](#)
  - SCnSCB\_ICTR\_INTLINESNUM\_Pos, [521](#)
- System Tick Timer (SysTick), [522](#)
  - SysTick\_CALIB\_NOREF\_Msk, [523](#)
  - SysTick\_CALIB\_NOREF\_Pos, [523](#)
  - SysTick\_CALIB\_SKEW\_Msk, [523](#)
  - SysTick\_CALIB\_SKEW\_Pos, [523](#)
  - SysTick\_CALIB\_TENMS\_Msk, [523](#)
  - SysTick\_CALIB\_TENMS\_Pos, [524](#)
  - SysTick\_CTRL\_CLKSOURCE\_Msk, [524](#)
  - SysTick\_CTRL\_CLKSOURCE\_Pos, [524](#)
  - SysTick\_CTRL\_COUNTFLAG\_Msk, [524](#)
  - SysTick\_CTRL\_COUNTFLAG\_Pos, [524](#), [525](#)
  - SysTick\_CTRL\_ENABLE\_Msk, [525](#)
  - SysTick\_CTRL\_ENABLE\_Pos, [525](#)
  - SysTick\_CTRL\_TICKINT\_Msk, [525](#)
  - SysTick\_CTRL\_TICKINT\_Pos, [525](#)
  - SysTick\_LOAD\_RELOAD\_Msk, [526](#)
  - SysTick\_LOAD\_RELOAD\_Pos, [526](#)
  - SysTick\_VAL\_CURRENT\_Msk, [526](#)
  - SysTick\_VAL\_CURRENT\_Pos, [526](#)
- SystemCoreClock
  - CHIP: LPC5411X support functions, [387](#)
- SystemCoreClockUpdate
  - CHIP: LPC5411X support functions, [386](#)
- SystemInit
  - sysinit.c, [808](#)
- T
  - xPSR\_Type, [668](#)
- TC
  - LPC\_TIMER\_T, [620](#)
  - LPC\_WWDT\_T, [625](#)
- TCR
  - ITM\_Type, [569](#)
  - LPC\_TIMER\_T, [620](#)
- TDM19EN
  - CHIP: LPC5411X DMIC driver, [147](#)
- TDM96EN
  - CHIP: LPC5411X DMIC driver, [147](#)
- TER
  - ITM\_Type, [569](#)
- THR\_HIGH
  - LPC\_ADC\_T, [572](#)
- THR\_LOW
  - LPC\_ADC\_T, [572](#)
- TIMEOUT
  - CHIP: LPC5411x I2C driver, [416](#)
- TIMER
  - LPC\_MRT\_CH\_T, [590](#)
- TIMER0\_IRQHandler
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- TIMER0\_IRQn
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [199](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- TIMER1\_IRQHandler
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [200](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [203](#)
- TIMER1\_IRQn
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [200](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [204](#)
- TIMER2\_IRQHandler
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [200](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [204](#)
- TIMER2\_IRQn
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [200](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [204](#)
- TIMER3\_IRQHandler
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [200](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [204](#)
- TIMER3\_IRQn
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [200](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [204](#)
- TIMER4\_IRQHandler
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [200](#)

- CHIP: LPC5411X M4 core Cortex CMSIS definitions, [204](#)
- TIMER4\_IRQn
  - CHIP: LPC5411X M0 core Cortex CMSIS definitions, [200](#)
  - CHIP: LPC5411X M4 core Cortex CMSIS definitions, [204](#)
- TIMER\_CAP\_FALLING
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_CAP\_INT
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_CAP\_RISING
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_CAP\_SRC\_STATE\_T
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_CAPSRC\_BOTH\_CAPN
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_CAPSRC\_FALLING\_CAPN
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_CAPSRC\_RISING\_CAPN
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_CAPSRC\_RISING\_PCLK
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_CCR\_MASK
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_CTCR\_MASK
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_CTRL\_MASK
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_EMR\_MASK
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_ENABLE
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_EXTMATCH\_CLEAR
  - CHIP: LPC5411X 32-bit Timer driver, [61](#)
- TIMER\_EXTMATCH\_DO\_NOTHING
  - CHIP: LPC5411X 32-bit Timer driver, [61](#)
- TIMER\_EXTMATCH\_SET
  - CHIP: LPC5411X 32-bit Timer driver, [61](#)
- TIMER\_EXTMATCH\_TOGGLE
  - CHIP: LPC5411X 32-bit Timer driver, [61](#)
- TIMER\_INT\_ON\_CAP
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_INT\_ON\_MATCH
  - CHIP: LPC5411X 32-bit Timer driver, [59](#)
- TIMER\_IR\_CLR
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_MATCH\_INT
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_MCR\_MASK
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_PIN\_MATCH\_STATE\_T
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_RESET
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_RESET\_ON\_MATCH
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TIMER\_STOP\_ON\_MATCH
  - CHIP: LPC5411X 32-bit Timer driver, [60](#)
- TM\_DAYOFWEEK
  - CHIP: RTC tick to (a more) Universal Time conversion functions, [417](#)
- TM\_YEAR\_BASE
  - CHIP: RTC tick to (a more) Universal Time conversion functions, [417](#)
- TPI
  - Core Definitions, [453](#)
- TPI\_ACPR\_PRESCALER\_Msk
  - Trace Port Interface (TPI), [528](#)
- TPI\_ACPR\_PRESCALER\_Pos
  - Trace Port Interface (TPI), [528](#)
- TPI\_BASE
  - Core Definitions, [454](#)
- TPI\_DEVID\_AsynClkIn\_Msk
  - Trace Port Interface (TPI), [528](#)
- TPI\_DEVID\_AsynClkIn\_Pos
  - Trace Port Interface (TPI), [528](#)
- TPI\_DEVID\_MANCVALID\_Msk
  - Trace Port Interface (TPI), [528](#)
- TPI\_DEVID\_MANCVALID\_Pos
  - Trace Port Interface (TPI), [529](#)
- TPI\_DEVID\_MinBufSz\_Msk
  - Trace Port Interface (TPI), [529](#)
- TPI\_DEVID\_MinBufSz\_Pos
  - Trace Port Interface (TPI), [529](#)
- TPI\_DEVID\_NRZVALID\_Msk
  - Trace Port Interface (TPI), [529](#)
- TPI\_DEVID\_NRZVALID\_Pos
  - Trace Port Interface (TPI), [529](#)
- TPI\_DEVID\_NrTraceInput\_Msk
  - Trace Port Interface (TPI), [529](#)
- TPI\_DEVID\_NrTraceInput\_Pos
  - Trace Port Interface (TPI), [529](#)
- TPI\_DEVID\_PTINVALID\_Msk
  - Trace Port Interface (TPI), [529](#)
- TPI\_DEVID\_PTINVALID\_Pos
  - Trace Port Interface (TPI), [529](#)
- TPI\_DEVTTYPE\_MajorType\_Msk
  - Trace Port Interface (TPI), [530](#)
- TPI\_DEVTTYPE\_MajorType\_Pos
  - Trace Port Interface (TPI), [530](#)
- TPI\_DEVTTYPE\_SubType\_Msk
  - Trace Port Interface (TPI), [530](#)
- TPI\_DEVTTYPE\_SubType\_Pos
  - Trace Port Interface (TPI), [530](#)
- TPI\_FFCR\_EnFCont\_Msk
  - Trace Port Interface (TPI), [530](#)
- TPI\_FFCR\_EnFCont\_Pos
  - Trace Port Interface (TPI), [530](#)
- TPI\_FFCR\_TrigIn\_Msk
  - Trace Port Interface (TPI), [530](#)
- TPI\_FFCR\_TrigIn\_Pos
  - Trace Port Interface (TPI), [530](#)
- TPI\_FFSR\_FlInProg\_Msk
  - Trace Port Interface (TPI), [530](#)
- TPI\_FFSR\_FlInProg\_Pos
  - Trace Port Interface (TPI), [530](#)

- Trace Port Interface (TPI), [531](#)
- TPI\_FFSR\_FtNonStop\_Msk
  - Trace Port Interface (TPI), [531](#)
- TPI\_FFSR\_FtNonStop\_Pos
  - Trace Port Interface (TPI), [531](#)
- TPI\_FFSR\_FtStopped\_Msk
  - Trace Port Interface (TPI), [531](#)
- TPI\_FFSR\_FtStopped\_Pos
  - Trace Port Interface (TPI), [531](#)
- TPI\_FFSR\_TCPresent\_Msk
  - Trace Port Interface (TPI), [531](#)
- TPI\_FFSR\_TCPresent\_Pos
  - Trace Port Interface (TPI), [531](#)
- TPI\_FIFO0\_ETM0\_Msk
  - Trace Port Interface (TPI), [531](#)
- TPI\_FIFO0\_ETM0\_Pos
  - Trace Port Interface (TPI), [531](#)
- TPI\_FIFO0\_ETM1\_Msk
  - Trace Port Interface (TPI), [532](#)
- TPI\_FIFO0\_ETM1\_Pos
  - Trace Port Interface (TPI), [532](#)
- TPI\_FIFO0\_ETM2\_Msk
  - Trace Port Interface (TPI), [532](#)
- TPI\_FIFO0\_ETM2\_Pos
  - Trace Port Interface (TPI), [532](#)
- TPI\_FIFO0\_ETM\_ATVALID\_Msk
  - Trace Port Interface (TPI), [532](#)
- TPI\_FIFO0\_ETM\_ATVALID\_Pos
  - Trace Port Interface (TPI), [532](#)
- TPI\_FIFO0\_ETM\_bytecount\_Msk
  - Trace Port Interface (TPI), [532](#)
- TPI\_FIFO0\_ETM\_bytecount\_Pos
  - Trace Port Interface (TPI), [532](#)
- TPI\_FIFO0\_ITM\_ATVALID\_Msk
  - Trace Port Interface (TPI), [532](#)
- TPI\_FIFO0\_ITM\_ATVALID\_Pos
  - Trace Port Interface (TPI), [533](#)
- TPI\_FIFO0\_ITM\_bytecount\_Msk
  - Trace Port Interface (TPI), [533](#)
- TPI\_FIFO0\_ITM\_bytecount\_Pos
  - Trace Port Interface (TPI), [533](#)
- TPI\_FIFO1\_ETM\_ATVALID\_Msk
  - Trace Port Interface (TPI), [533](#)
- TPI\_FIFO1\_ETM\_ATVALID\_Pos
  - Trace Port Interface (TPI), [533](#)
- TPI\_FIFO1\_ETM\_bytecount\_Msk
  - Trace Port Interface (TPI), [533](#)
- TPI\_FIFO1\_ETM\_bytecount\_Pos
  - Trace Port Interface (TPI), [533](#)
- TPI\_FIFO1\_ITM0\_Msk
  - Trace Port Interface (TPI), [533](#)
- TPI\_FIFO1\_ITM0\_Pos
  - Trace Port Interface (TPI), [533](#)
- TPI\_FIFO1\_ITM1\_Msk
  - Trace Port Interface (TPI), [534](#)
- TPI\_FIFO1\_ITM1\_Pos
  - Trace Port Interface (TPI), [534](#)
- TPI\_FIFO1\_ITM2\_Msk
  - Trace Port Interface (TPI), [534](#)
- Trace Port Interface (TPI), [534](#)
- TPI\_FIFO1\_ITM2\_Pos
  - Trace Port Interface (TPI), [534](#)
- TPI\_FIFO1\_ITM\_ATVALID\_Msk
  - Trace Port Interface (TPI), [534](#)
- TPI\_FIFO1\_ITM\_ATVALID\_Pos
  - Trace Port Interface (TPI), [534](#)
- TPI\_FIFO1\_ITM\_bytecount\_Msk
  - Trace Port Interface (TPI), [534](#)
- TPI\_FIFO1\_ITM\_bytecount\_Pos
  - Trace Port Interface (TPI), [534](#)
- TPI\_ITATBCTR0\_ATREADY\_Msk
  - Trace Port Interface (TPI), [534](#)
- TPI\_ITATBCTR0\_ATREADY\_Pos
  - Trace Port Interface (TPI), [535](#)
- TPI\_ITATBCTR2\_ATREADY\_Msk
  - Trace Port Interface (TPI), [535](#)
- TPI\_ITATBCTR2\_ATREADY\_Pos
  - Trace Port Interface (TPI), [535](#)
- TPI\_ITCTRL\_Mode\_Msk
  - Trace Port Interface (TPI), [535](#)
- TPI\_ITCTRL\_Mode\_Pos
  - Trace Port Interface (TPI), [535](#)
- TPI\_SPPR\_TXMODE\_Msk
  - Trace Port Interface (TPI), [535](#)
- TPI\_SPPR\_TXMODE\_Pos
  - Trace Port Interface (TPI), [535](#)
- TPI\_TRIGGER\_TRIGGER\_Msk
  - Trace Port Interface (TPI), [535](#)
- TPI\_TRIGGER\_TRIGGER\_Pos
  - Trace Port Interface (TPI), [535](#)
- TPI\_Type, [644](#)
  - ACPR, [645](#)
  - CLAIMCLR, [645](#)
  - CLAIMSET, [645](#)
  - CSPSR, [645](#)
  - DEVID, [645](#)
  - DEVTYPE, [645](#)
  - FFCR, [645](#)
  - FFSR, [645](#)
  - FIFO0, [645](#)
  - FIFO1, [645](#)
  - FSCR, [646](#)
  - ITATBCTR0, [646](#)
  - ITATBCTR2, [646](#)
  - ITCTRL, [646](#)
  - RESERVED0, [646](#)
  - RESERVED1, [646](#)
  - RESERVED2, [646](#)
  - RESERVED3, [646](#)
  - RESERVED4, [646](#)
  - RESERVED5, [646](#)
  - RESERVED7, [646](#)
  - SPPR, [647](#)
  - SSPSR, [647](#)
  - TRIGGER, [647](#)
- TPR
  - ITM\_Type, [569](#)

- TRANSFER\_BLOCK\_T
  - LPC Public Types, [480](#)
- TRIGGER
  - TPI\_Type, [647](#)
- TRUE
  - LPC Public Types, [479](#)
- TV
  - LPC\_WWDT\_T, [625](#)
- tail
  - RINGBUFF\_T, [633](#)
- thresCount
  - SPIS\_XFER\_T, [642](#)
- ticksPerMs
  - stopwatch\_5411x.c, [801](#)
- ticksPerSecond
  - stopwatch\_5411x.c, [801](#)
- ticksPerUs
  - stopwatch\_5411x.c, [801](#)
- timer\_5411x.c
  - LAST\_TIMER, [803](#)
- Trace Port Interface (TPI), [527](#)
  - TPI\_ACPR\_PRESCALER\_Msk, [528](#)
  - TPI\_ACPR\_PRESCALER\_Pos, [528](#)
  - TPI\_DEVID\_AsynClkIn\_Msk, [528](#)
  - TPI\_DEVID\_AsynClkIn\_Pos, [528](#)
  - TPI\_DEVID\_MANCVALID\_Msk, [528](#)
  - TPI\_DEVID\_MANCVALID\_Pos, [529](#)
  - TPI\_DEVID\_MinBufSz\_Msk, [529](#)
  - TPI\_DEVID\_MinBufSz\_Pos, [529](#)
  - TPI\_DEVID\_NRZVALID\_Msk, [529](#)
  - TPI\_DEVID\_NRZVALID\_Pos, [529](#)
  - TPI\_DEVID\_NrTraceInput\_Msk, [529](#)
  - TPI\_DEVID\_NrTraceInput\_Pos, [529](#)
  - TPI\_DEVID\_PTINVALID\_Msk, [529](#)
  - TPI\_DEVID\_PTINVALID\_Pos, [529](#)
  - TPI\_DEVTYPE\_MajorType\_Msk, [530](#)
  - TPI\_DEVTYPE\_MajorType\_Pos, [530](#)
  - TPI\_DEVTYPE\_SubType\_Msk, [530](#)
  - TPI\_DEVTYPE\_SubType\_Pos, [530](#)
  - TPI\_FFCR\_EnFCont\_Msk, [530](#)
  - TPI\_FFCR\_EnFCont\_Pos, [530](#)
  - TPI\_FFCR\_TrigIn\_Msk, [530](#)
  - TPI\_FFCR\_TrigIn\_Pos, [530](#)
  - TPI\_FFSR\_FlInProg\_Msk, [530](#)
  - TPI\_FFSR\_FlInProg\_Pos, [531](#)
  - TPI\_FFSR\_FtNonStop\_Msk, [531](#)
  - TPI\_FFSR\_FtNonStop\_Pos, [531](#)
  - TPI\_FFSR\_FtStopped\_Msk, [531](#)
  - TPI\_FFSR\_FtStopped\_Pos, [531](#)
  - TPI\_FFSR\_TCPresent\_Msk, [531](#)
  - TPI\_FFSR\_TCPresent\_Pos, [531](#)
  - TPI\_FIFO0\_ETM0\_Msk, [531](#)
  - TPI\_FIFO0\_ETM0\_Pos, [531](#)
  - TPI\_FIFO0\_ETM1\_Msk, [532](#)
  - TPI\_FIFO0\_ETM1\_Pos, [532](#)
  - TPI\_FIFO0\_ETM2\_Msk, [532](#)
  - TPI\_FIFO0\_ETM2\_Pos, [532](#)
  - TPI\_FIFO0\_ETM\_ATVALID\_Msk, [532](#)
  - TPI\_FIFO0\_ETM\_ATVALID\_Pos, [532](#)
  - TPI\_FIFO0\_ETM\_bytecount\_Msk, [532](#)
  - TPI\_FIFO0\_ETM\_bytecount\_Pos, [532](#)
  - TPI\_FIFO0\_ITM\_ATVALID\_Msk, [532](#)
  - TPI\_FIFO0\_ITM\_ATVALID\_Pos, [533](#)
  - TPI\_FIFO0\_ITM\_bytecount\_Msk, [533](#)
  - TPI\_FIFO0\_ITM\_bytecount\_Pos, [533](#)
  - TPI\_FIFO1\_ETM\_ATVALID\_Msk, [533](#)
  - TPI\_FIFO1\_ETM\_ATVALID\_Pos, [533](#)
  - TPI\_FIFO1\_ETM\_bytecount\_Msk, [533](#)
  - TPI\_FIFO1\_ETM\_bytecount\_Pos, [533](#)
  - TPI\_FIFO1\_ITM0\_Msk, [533](#)
  - TPI\_FIFO1\_ITM0\_Pos, [533](#)
  - TPI\_FIFO1\_ITM1\_Msk, [534](#)
  - TPI\_FIFO1\_ITM1\_Pos, [534](#)
  - TPI\_FIFO1\_ITM2\_Msk, [534](#)
  - TPI\_FIFO1\_ITM2\_Pos, [534](#)
  - TPI\_FIFO1\_ITM\_ATVALID\_Msk, [534](#)
  - TPI\_FIFO1\_ITM\_ATVALID\_Pos, [534](#)
  - TPI\_FIFO1\_ITM\_bytecount\_Msk, [534](#)
  - TPI\_FIFO1\_ITM\_bytecount\_Pos, [534](#)
  - TPI\_ITATBCTR0\_ATREADY\_Msk, [534](#)
  - TPI\_ITATBCTR0\_ATREADY\_Pos, [535](#)
  - TPI\_ITATBCTR2\_ATREADY\_Msk, [535](#)
  - TPI\_ITATBCTR2\_ATREADY\_Pos, [535](#)
  - TPI\_ITCTRL\_Mode\_Msk, [535](#)
  - TPI\_ITCTRL\_Mode\_Pos, [535](#)
  - TPI\_SPPR\_TXMODE\_Msk, [535](#)
  - TPI\_SPPR\_TXMODE\_Pos, [535](#)
  - TPI\_TRIGGER\_TRIGGER\_Msk, [535](#)
  - TPI\_TRIGGER\_TRIGGER\_Pos, [535](#)
- TransferDelay
  - SPIM\_DELAY\_CONFIG\_T, [639](#)
- txBuff
  - I2CM\_XFER\_T, [560](#)
- txCount
  - SPIM\_XFER\_T, [640](#)
  - SPIS\_XFER\_T, [642](#)
- txData
  - SPIM\_XFER\_T, [641](#)
  - SPIS\_XFER\_T, [643](#)
- txIndex
  - SPIM\_XFER\_T, [641](#)
  - SPIS\_XFER\_T, [643](#)
- txSz
  - I2CM\_XFER\_T, [560](#)
- Type
  - \_BM\_T, [551](#)
- U
  - LPC\_SCT\_T, [604](#)
- u16
  - ITM\_Type, [570](#)
- u32
  - ITM\_Type, [570](#)
- u8
  - ITM\_Type, [570](#)
- UART\_BAUD\_T, [647](#)
  - baud, [648](#)



- clk, [648](#)
- div, [648](#)
- mul, [648](#)
- ovr, [648](#)
- UART\_CFG\_AUTOADDR
  - CHIP: LPC5411X UART Driver, [344](#)
- UART\_CFG\_BITMASK
  - CHIP: LPC5411X UART Driver, [344](#)
- UART\_CFG\_CLKPOL
  - CHIP: LPC5411X UART Driver, [345](#)
- UART\_CFG\_CTSEN
  - CHIP: LPC5411X UART Driver, [345](#)
- UART\_CFG\_DATALEN\_7
  - CHIP: LPC5411X UART Driver, [345](#)
- UART\_CFG\_DATALEN\_8
  - CHIP: LPC5411X UART Driver, [345](#)
- UART\_CFG\_DATALEN\_9
  - CHIP: LPC5411X UART Driver, [345](#)
- UART\_CFG\_ENABLE
  - CHIP: LPC5411X UART Driver, [345](#)
- UART\_CFG\_IOMODE
  - CHIP: LPC5411X UART Driver, [345](#)
- UART\_CFG\_LINMODE
  - CHIP: LPC5411X UART Driver, [345](#)
- UART\_CFG\_LOOP
  - CHIP: LPC5411X UART Driver, [345](#)
- UART\_CFG\_MODE32K
  - CHIP: LPC5411X UART Driver, [346](#)
- UART\_CFG\_OEPOL
  - CHIP: LPC5411X UART Driver, [346](#)
- UART\_CFG\_OESEL
  - CHIP: LPC5411X UART Driver, [346](#)
- UART\_CFG\_OETA
  - CHIP: LPC5411X UART Driver, [346](#)
- UART\_CFG\_PARITY\_EVEN
  - CHIP: LPC5411X UART Driver, [346](#)
- UART\_CFG\_PARITY\_NONE
  - CHIP: LPC5411X UART Driver, [346](#)
- UART\_CFG\_PARITY\_ODD
  - CHIP: LPC5411X UART Driver, [346](#)
- UART\_CFG\_RXPOL
  - CHIP: LPC5411X UART Driver, [346](#)
- UART\_CFG\_STOPLEN\_1
  - CHIP: LPC5411X UART Driver, [346](#)
- UART\_CFG\_STOPLEN\_2
  - CHIP: LPC5411X UART Driver, [347](#)
- UART\_CFG\_SYNCEN
  - CHIP: LPC5411X UART Driver, [347](#)
- UART\_CFG\_SYNCMST
  - CHIP: LPC5411X UART Driver, [347](#)
- UART\_CFG\_TXPOL
  - CHIP: LPC5411X UART Driver, [347](#)
- UART\_CTRL\_ADDRDET
  - CHIP: LPC5411X UART Driver, [347](#)
- UART\_CTRL\_AUTOBAUD
  - CHIP: LPC5411X UART Driver, [347](#)
- UART\_CTRL\_CC
  - CHIP: LPC5411X UART Driver, [347](#)
- UART\_CTRL\_CLRCCONRX
  - CHIP: LPC5411X UART Driver, [347](#)
- UART\_CTRL\_TXBRKEN
  - CHIP: LPC5411X UART Driver, [347](#)
- UART\_CTRL\_TXDIS
  - CHIP: LPC5411X UART Driver, [348](#)
- UART\_FIFO\_DEPTH
  - CHIP: LPC5411X UART Driver, [348](#)
- UART\_FIFOCFG\_BITMASK
  - CHIP: LPC5411X UART Driver, [348](#)
- UART\_FIFOCFG\_DMARX
  - CHIP: LPC5411X UART Driver, [348](#)
- UART\_FIFOCFG\_DMATX
  - CHIP: LPC5411X UART Driver, [348](#)
- UART\_FIFOCFG\_EMPTYRX
  - CHIP: LPC5411X UART Driver, [348](#)
- UART\_FIFOCFG\_EMPTYTX
  - CHIP: LPC5411X UART Driver, [348](#)
- UART\_FIFOCFG\_ENABLERX
  - CHIP: LPC5411X UART Driver, [348](#)
- UART\_FIFOCFG\_ENABLETX
  - CHIP: LPC5411X UART Driver, [348](#)
- UART\_FIFOCFG\_WAKERX
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOCFG\_WAKETX
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOINT\_BITMASK
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOINT\_PERINT
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOINT\_RXERR
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOINT\_RXLVL
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOINT\_TXERR
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOINT\_TXLVL
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOSTAT\_BITMASK
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOSTAT\_PERIPH
  - CHIP: LPC5411X UART Driver, [349](#)
- UART\_FIFOSTAT\_RXERR
  - CHIP: LPC5411X UART Driver, [350](#)
- UART\_FIFOSTAT\_RXFULL
  - CHIP: LPC5411X UART Driver, [350](#)
- UART\_FIFOSTAT\_RXLVL
  - CHIP: LPC5411X UART Driver, [350](#)
- UART\_FIFOSTAT\_RXNOTEMPTY
  - CHIP: LPC5411X UART Driver, [350](#)
- UART\_FIFOSTAT\_TXEMPTY
  - CHIP: LPC5411X UART Driver, [350](#)
- UART\_FIFOSTAT\_TXERR
  - CHIP: LPC5411X UART Driver, [350](#)
- UART\_FIFOSTAT\_TXLVL
  - CHIP: LPC5411X UART Driver, [350](#)
- UART\_FIFOSTAT\_TXNOTFULL
  - CHIP: LPC5411X UART Driver, [350](#)

- UART\_FIFOTRIG\_BITMASK
  - CHIP: LPC5411X UART Driver, [350](#)
- UART\_FIFOTRIG\_RXLVL
  - CHIP: LPC5411X UART Driver, [351](#)
- UART\_FIFOTRIG\_RXLVLENA
  - CHIP: LPC5411X UART Driver, [351](#)
- UART\_FIFOTRIG\_TXLVL
  - CHIP: LPC5411X UART Driver, [351](#)
- UART\_FIFOTRIG\_TXLVLENA
  - CHIP: LPC5411X UART Driver, [351](#)
- UART\_INT\_ABERR
  - CHIP: LPC5411X UART Driver, [351](#)
- UART\_INT\_DELTACTS
  - CHIP: LPC5411X UART Driver, [351](#)
- UART\_INT\_DELTARXBRK
  - CHIP: LPC5411X UART Driver, [351](#)
- UART\_INT\_FRAMERR
  - CHIP: LPC5411X UART Driver, [351](#)
- UART\_INT\_PARITYERR
  - CHIP: LPC5411X UART Driver, [351](#)
- UART\_INT\_RXNOISE
  - CHIP: LPC5411X UART Driver, [352](#)
- UART\_INT\_START
  - CHIP: LPC5411X UART Driver, [352](#)
- UART\_INT\_TXDIS
  - CHIP: LPC5411X UART Driver, [352](#)
- UART\_INT\_TXIDLE
  - CHIP: LPC5411X UART Driver, [352](#)
- UART\_STAT\_ABERR
  - CHIP: LPC5411X UART Driver, [352](#)
- UART\_STAT\_CTS
  - CHIP: LPC5411X UART Driver, [352](#)
- UART\_STAT\_DELTACTS
  - CHIP: LPC5411X UART Driver, [352](#)
- UART\_STAT\_DELTARXBRK
  - CHIP: LPC5411X UART Driver, [352](#)
- UART\_STAT\_FRM\_ERRINT
  - CHIP: LPC5411X UART Driver, [352](#)
- UART\_STAT\_PAR\_ERRINT
  - CHIP: LPC5411X UART Driver, [353](#)
- UART\_STAT\_RXBRK
  - CHIP: LPC5411X UART Driver, [353](#)
- UART\_STAT\_RXIDLE
  - CHIP: LPC5411X UART Driver, [353](#)
- UART\_STAT\_RXNOISEINT
  - CHIP: LPC5411X UART Driver, [353](#)
- UART\_STAT\_START
  - CHIP: LPC5411X UART Driver, [353](#)
- UART\_STAT\_TXDISINT
  - CHIP: LPC5411X UART Driver, [353](#)
- UART\_STAT\_TXIDLE
  - CHIP: LPC5411X UART Driver, [353](#)
- UART\_STATISTICS\_T, [648](#)
  - fifo\_err\_rx, [649](#)
  - fifo\_err\_tx, [649](#)
  - interrupts, [649](#)
  - lvl\_rx, [649](#)
  - lvl\_tx, [649](#)
  - uart\_break, [649](#)
  - uart\_cts, [649](#)
  - uart\_err\_auto\_baud, [649](#)
  - uart\_err\_frame, [650](#)
  - uart\_err\_parity, [650](#)
  - uart\_err\_rx\_noise, [650](#)
  - uart\_start, [650](#)
- UNS\_16
  - LPC Public Types, [479](#)
- UNS\_32
  - LPC Public Types, [479](#)
- UNS\_64
  - LPC Public Types, [479](#)
- UNS\_8
  - LPC Public Types, [479](#)
- USB\_COMMON\_DESCRIPTOR, [650](#)
- USB\_CONFIG\_BUS\_POWERED
  - USBD\_Core, [538](#)
- USB\_CONFIG\_POWER\_MA
  - USBD\_Core, [539](#)
- USB\_CONFIG\_POWERED\_MASK
  - USBD\_Core, [539](#)
- USB\_CONFIG\_REMOTE\_WAKEUP
  - USBD\_Core, [539](#)
- USB\_CONFIG\_SELF\_POWERED
  - USBD\_Core, [539](#)
- USB\_CONFIGURATION\_DESC\_SIZE
  - USBD\_Core, [539](#)
- USB\_CONFIGURATION\_DESCRIPTOR, [651](#)
- USB\_CONFIGURATION\_DESCRIPTOR\_TYPE
  - USBD\_Core, [539](#)
- USB\_DEBUG\_DESCRIPTOR\_TYPE
  - USBD\_Core, [539](#)
- USB\_DEVICE\_CLASS\_APP
  - USBD\_Core, [539](#)
- USB\_DEVICE\_CLASS\_AUDIO
  - USBD\_Core, [539](#)
- USB\_DEVICE\_CLASS\_COMMUNICATIONS
  - USBD\_Core, [540](#)
- USB\_DEVICE\_CLASS\_HUB
  - USBD\_Core, [540](#)
- USB\_DEVICE\_CLASS\_HUMAN\_INTERFACE
  - USBD\_Core, [540](#)
- USB\_DEVICE\_CLASS\_MISCELLANEOUS
  - USBD\_Core, [540](#)
- USB\_DEVICE\_CLASS\_MONITOR
  - USBD\_Core, [540](#)
- USB\_DEVICE\_CLASS\_PHYSICAL\_INTERFACE
  - USBD\_Core, [540](#)
- USB\_DEVICE\_CLASS\_POWER
  - USBD\_Core, [540](#)
- USB\_DEVICE\_CLASS\_PRINTER
  - USBD\_Core, [540](#)
- USB\_DEVICE\_CLASS\_RESERVED
  - USBD\_Core, [540](#)
- USB\_DEVICE\_CLASS\_STORAGE
  - USBD\_Core, [541](#)
- USB\_DEVICE\_CLASS\_VENDOR\_SPECIFIC



- USBD\_Core, [541](#)
- USB\_DEVICE\_DESC\_SIZE
  - USBD\_Core, [541](#)
- USB\_DEVICE\_DESCRIPTOR, [652](#)
- USB\_DEVICE\_DESCRIPTOR\_TYPE
  - USBD\_Core, [541](#)
- USB\_DEVICE\_QUALI\_SIZE
  - USBD\_Core, [541](#)
- USB\_DEVICE\_QUALIFIER\_DESCRIPTOR, [655](#)
- USB\_DEVICE\_QUALIFIER\_DESCRIPTOR\_TYPE
  - USBD\_Core, [541](#)
- USB\_ENDPOINT\_0\_HS\_MAXP
  - USBD\_Core, [541](#)
- USB\_ENDPOINT\_0\_LS\_MAXP
  - USBD\_Core, [541](#)
- USB\_ENDPOINT\_BULK\_HS\_MAXP
  - USBD\_Core, [541](#)
- USB\_ENDPOINT\_DESC\_SIZE
  - USBD\_Core, [541](#)
- USB\_ENDPOINT\_DESCRIPTOR, [656](#)
- USB\_ENDPOINT\_DESCRIPTOR\_TYPE
  - USBD\_Core, [542](#)
- USB\_ENDPOINT\_DIRECTION\_MASK
  - USBD\_Core, [542](#)
- USB\_ENDPOINT\_IN
  - USBD\_Core, [542](#)
- USB\_ENDPOINT\_OUT
  - USBD\_Core, [542](#)
- USB\_ENDPOINT\_SYNC\_ADAPTIVE
  - USBD\_Core, [542](#)
- USB\_ENDPOINT\_SYNC\_ASYNCHRONOUS
  - USBD\_Core, [542](#)
- USB\_ENDPOINT\_SYNC\_MASK
  - USBD\_Core, [542](#)
- USB\_ENDPOINT\_SYNC\_NO\_SYNCHRONIZATION
  - USBD\_Core, [542](#)
- USB\_ENDPOINT\_SYNC\_SYNCHRONOUS
  - USBD\_Core, [542](#)
- USB\_ENDPOINT\_TYPE\_BULK
  - USBD\_Core, [543](#)
- USB\_ENDPOINT\_TYPE\_CONTROL
  - USBD\_Core, [543](#)
- USB\_ENDPOINT\_TYPE\_INTERRUPT
  - USBD\_Core, [543](#)
- USB\_ENDPOINT\_TYPE\_ISOCHRONOUS
  - USBD\_Core, [543](#)
- USB\_ENDPOINT\_TYPE\_MASK
  - USBD\_Core, [543](#)
- USB\_ENDPOINT\_USAGE\_DATA
  - USBD\_Core, [543](#)
- USB\_ENDPOINT\_USAGE\_FEEDBACK
  - USBD\_Core, [543](#)
- USB\_ENDPOINT\_USAGE\_IMPLICIT\_FEEDBACK
  - USBD\_Core, [543](#)
- USB\_ENDPOINT\_USAGE\_MASK
  - USBD\_Core, [543](#)
- USB\_ENDPOINT\_USAGE\_RESERVED
  - USBD\_Core, [544](#)
- USB\_FEATURE\_ENDPOINT\_STALL
  - USBD\_Core, [544](#)
- USB\_FEATURE\_REMOTE\_WAKEUP
  - USBD\_Core, [544](#)
- USB\_FEATURE\_TEST\_MODE
  - USBD\_Core, [544](#)
- USB\_GETSTATUS\_ENDPOINT\_STALL
  - USBD\_Core, [544](#)
- USB\_GETSTATUS\_REMOTE\_WAKEUP
  - USBD\_Core, [544](#)
- USB\_GETSTATUS\_SELF\_POWERED
  - USBD\_Core, [544](#)
- USB\_IAD\_DESCRIPTOR, [659](#)
- USB\_INTERFACE\_ASSOC\_DESC\_SIZE
  - USBD\_Core, [544](#)
- USB\_INTERFACE\_ASSOCIATION\_DESCRIPTOR\_↔
  - TYPE
    - USBD\_Core, [544](#)
- USB\_INTERFACE\_DESC\_SIZE
  - USBD\_Core, [545](#)
- USB\_INTERFACE\_DESCRIPTOR, [660](#)
- USB\_INTERFACE\_DESCRIPTOR\_TYPE
  - USBD\_Core, [545](#)
- USB\_INTERFACE\_POWER\_DESCRIPTOR\_TYPE
  - USBD\_Core, [545](#)
- USB\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- USB\_OTG\_DESCRIPTOR\_TYPE
  - USBD\_Core, [545](#)
- USB\_OTHER\_SPEED\_CONF\_SIZE
  - USBD\_Core, [545](#)
- USB\_OTHER\_SPEED\_CONFIG\_DESCRIPTOR\_TY↔
  - PE
    - USBD\_Core, [545](#)
- USB\_OTHER\_SPEED\_CONFIGURATION, [662](#)
- USB\_REQUEST\_CLEAR\_FEATURE
  - USBD\_Core, [545](#)
- USB\_REQUEST\_GET\_CONFIGURATION
  - USBD\_Core, [545](#)
- USB\_REQUEST\_GET\_DESCRIPTOR
  - USBD\_Core, [545](#)
- USB\_REQUEST\_GET\_INTERFACE
  - USBD\_Core, [545](#)
- USB\_REQUEST\_GET\_STATUS
  - USBD\_Core, [546](#)
- USB\_REQUEST\_SET\_ADDRESS
  - USBD\_Core, [546](#)
- USB\_REQUEST\_SET\_CONFIGURATION
  - USBD\_Core, [546](#)
- USB\_REQUEST\_SET\_DESCRIPTOR
  - USBD\_Core, [546](#)
- USB\_REQUEST\_SET\_FEATURE
  - USBD\_Core, [546](#)
- USB\_REQUEST\_SET\_INTERFACE
  - USBD\_Core, [546](#)

- USB\_REQUEST\_SYNC\_FRAME
  - USBD\_Core, [546](#)
- USB\_SETUP\_PACKET, [663](#)
- USB\_STRING\_DESCRIPTOR, [664](#)
- USB\_STRING\_DESCRIPTOR\_TYPE
  - USBD\_Core, [546](#)
- USBACT\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [429](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [431](#)
- USBCLKCTRL
  - LPC\_SYSCON\_T, [618](#)
- USBCLKDIV
  - LPC\_SYSCON\_T, [618](#)
- USBCLKSEL
  - LPC\_SYSCON\_T, [618](#)
- USBCLKSTAT
  - LPC\_SYSCON\_T, [618](#)
- USBD\_Core, [537](#)
  - B3VAL, [537](#)
  - REQUEST\_CLASS, [537](#)
  - REQUEST\_DEVICE\_TO\_HOST, [537](#)
  - REQUEST\_HOST\_TO\_DEVICE, [538](#)
  - REQUEST\_RESERVED, [538](#)
  - REQUEST\_STANDARD, [538](#)
  - REQUEST\_TO\_DEVICE, [538](#)
  - REQUEST\_TO\_ENDPOINT, [538](#)
  - REQUEST\_TO\_INTERFACE, [538](#)
  - REQUEST\_TO\_OTHER, [538](#)
  - REQUEST\_VENDOR, [538](#)
  - USB\_CONFIG\_BUS\_POWERED, [538](#)
  - USB\_CONFIG\_POWER\_MA, [539](#)
  - USB\_CONFIG\_POWERED\_MASK, [539](#)
  - USB\_CONFIG\_REMOTE\_WAKEUP, [539](#)
  - USB\_CONFIG\_SELF\_POWERED, [539](#)
  - USB\_CONFIGURATION\_DESC\_SIZE, [539](#)
  - USB\_CONFIGURATION\_DESCRIPTOR\_TYPE, [539](#)
  - USB\_DEBUG\_DESCRIPTOR\_TYPE, [539](#)
  - USB\_DEVICE\_CLASS\_APP, [539](#)
  - USB\_DEVICE\_CLASS\_AUDIO, [539](#)
  - USB\_DEVICE\_CLASS\_COMMUNICATIONS, [540](#)
  - USB\_DEVICE\_CLASS\_HUB, [540](#)
  - USB\_DEVICE\_CLASS\_HUMAN\_INTERFACE, [540](#)
  - USB\_DEVICE\_CLASS\_MISCELLANEOUS, [540](#)
  - USB\_DEVICE\_CLASS\_MONITOR, [540](#)
  - USB\_DEVICE\_CLASS\_PHYSICAL\_INTERFACE, [540](#)
  - USB\_DEVICE\_CLASS\_POWER, [540](#)
  - USB\_DEVICE\_CLASS\_PRINTER, [540](#)
  - USB\_DEVICE\_CLASS\_RESERVED, [540](#)
  - USB\_DEVICE\_CLASS\_STORAGE, [541](#)
  - USB\_DEVICE\_CLASS\_VENDOR\_SPECIFIC, [541](#)
  - USB\_DEVICE\_DESC\_SIZE, [541](#)
  - USB\_DEVICE\_DESCRIPTOR\_TYPE, [541](#)
  - USB\_DEVICE\_QUALI\_SIZE, [541](#)
- USB\_DEVICE\_QUALIFIER\_DESCRIPTOR\_TYPE, [541](#)
- USB\_ENDPOINT\_0\_HS\_MAXP, [541](#)
- USB\_ENDPOINT\_0\_LS\_MAXP, [541](#)
- USB\_ENDPOINT\_BULK\_HS\_MAXP, [541](#)
- USB\_ENDPOINT\_DESC\_SIZE, [541](#)
- USB\_ENDPOINT\_DESCRIPTOR\_TYPE, [542](#)
- USB\_ENDPOINT\_DIRECTION\_MASK, [542](#)
- USB\_ENDPOINT\_IN, [542](#)
- USB\_ENDPOINT\_OUT, [542](#)
- USB\_ENDPOINT\_SYNC\_ADAPTIVE, [542](#)
- USB\_ENDPOINT\_SYNC\_ASYNCHRONOUS, [542](#)
- USB\_ENDPOINT\_SYNC\_MASK, [542](#)
- USB\_ENDPOINT\_SYNC\_NO\_SYNCHRONIZATION, [542](#)
- USB\_ENDPOINT\_SYNC\_SYNCHRONOUS, [542](#)
- USB\_ENDPOINT\_TYPE\_BULK, [543](#)
- USB\_ENDPOINT\_TYPE\_CONTROL, [543](#)
- USB\_ENDPOINT\_TYPE\_INTERRUPT, [543](#)
- USB\_ENDPOINT\_TYPE\_ISOCHRONOUS, [543](#)
- USB\_ENDPOINT\_TYPE\_MASK, [543](#)
- USB\_ENDPOINT\_USAGE\_DATA, [543](#)
- USB\_ENDPOINT\_USAGE\_FEEDBACK, [543](#)
- USB\_ENDPOINT\_USAGE\_IMPLICIT\_FEEDBACK, [543](#)
- USB\_ENDPOINT\_USAGE\_MASK, [543](#)
- USB\_ENDPOINT\_USAGE\_RESERVED, [544](#)
- USB\_FEATURE\_ENDPOINT\_STALL, [544](#)
- USB\_FEATURE\_REMOTE\_WAKEUP, [544](#)
- USB\_FEATURE\_TEST\_MODE, [544](#)
- USB\_GETSTATUS\_ENDPOINT\_STALL, [544](#)
- USB\_GETSTATUS\_REMOTE\_WAKEUP, [544](#)
- USB\_GETSTATUS\_SELF\_POWERED, [544](#)
- USB\_INTERFACE\_ASSOC\_DESC\_SIZE, [544](#)
- USB\_INTERFACE\_ASSOCIATION\_DESCRIPTOR\_TYPE, [544](#)
- USB\_INTERFACE\_DESC\_SIZE, [545](#)
- USB\_INTERFACE\_DESCRIPTOR\_TYPE, [545](#)
- USB\_INTERFACE\_POWER\_DESCRIPTOR\_TYPE, [545](#)
- USB\_OTG\_DESCRIPTOR\_TYPE, [545](#)
- USB\_OTHER\_SPEED\_CONF\_SIZE, [545](#)
- USB\_OTHER\_SPEED\_CONFIG\_DESCRIPTOR\_TYPE, [545](#)
- USB\_REQUEST\_CLEAR\_FEATURE, [545](#)
- USB\_REQUEST\_GET\_CONFIGURATION, [545](#)
- USB\_REQUEST\_GET\_DESCRIPTOR, [545](#)
- USB\_REQUEST\_GET\_INTERFACE, [545](#)
- USB\_REQUEST\_GET\_STATUS, [546](#)
- USB\_REQUEST\_SET\_ADDRESS, [546](#)
- USB\_REQUEST\_SET\_CONFIGURATION, [546](#)
- USB\_REQUEST\_SET\_DESCRIPTOR, [546](#)
- USB\_REQUEST\_SET\_FEATURE, [546](#)
- USB\_REQUEST\_SET\_INTERFACE, [546](#)
- USB\_REQUEST\_SYNC\_FRAME, [546](#)
- USB\_STRING\_DESCRIPTOR\_TYPE, [546](#)
- USBD\_HANDLE\_T, [547](#)
- WBVAL, [546](#)

- USBD\_HANDLE\_T
  - USBD\_Core, [547](#)
- USE2FS
  - CHIP: LPC5411X DMIC driver, [147](#)
- UTICK\_CTRL\_DELAY\_MASK
  - CHIP: LPC5411X Micro Tick driver, [210](#)
- UTICK\_CTRL\_REPEAT
  - CHIP: LPC5411X Micro Tick driver, [210](#)
- UTICK\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- UTICK\_STATUS\_ACTIVE
  - CHIP: LPC5411X Micro Tick driver, [210](#)
- UTICK\_STATUS\_INTR
  - CHIP: LPC5411X Micro Tick driver, [211](#)
- UTICK\_STATUS\_MASK
  - CHIP: LPC5411X Micro Tick driver, [211](#)
- uart\_5411x.c
  - \_CalcErr, [804](#)
  - \_UART\_CalcDiv, [804](#)
  - \_UART\_CalcMul, [804](#)
  - \_UART\_DivClk, [804](#)
  - \_UART\_GetHighDiv, [804](#)
  - Chip\_UART\_CalculateBaud, [804](#)
  - fifo\_config\_dma, [805](#)
  - fifo\_config\_int, [805](#)
- uart\_break
  - UART\_STATISTICS\_T, [649](#)
- uart\_cts
  - UART\_STATISTICS\_T, [649](#)
- uart\_err\_auto\_baud
  - UART\_STATISTICS\_T, [649](#)
- uart\_err\_frame
  - UART\_STATISTICS\_T, [650](#)
- uart\_err\_parity
  - UART\_STATISTICS\_T, [650](#)
- uart\_err\_rx\_noise
  - UART\_STATISTICS\_T, [650](#)
- uart\_start
  - UART\_STATISTICS\_T, [650](#)
- unused
  - LPC\_MRT\_T, [591](#)
- UsageFault\_Handler
  - cr\_startup\_lpc5411x.c, [808](#)
- UsageFault\_IRQn
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- usbdApiBase
  - LPC\_ROM\_API\_T, [596](#)
- usrData
  - SPIM\_XFER\_T, [641](#)
- V
  - APSR\_Type, [550](#)
  - xPSR\_Type, [668](#)
- VAL
  - SysTick\_Type, [644](#)
- VTOR
  - SCB\_Type, [636](#)
- version
  - PINTABLE\_T, [629](#)
- W
  - \_\_WORD\_BYTE, [666](#)
  - LPC\_GPIO\_T, [582](#)
- w
  - APSR\_Type, [550](#)
  - CONTROL\_Type, [553](#)
  - IPSR\_Type, [566](#)
  - xPSR\_Type, [668](#)
- WAKE
  - LPC\_RTC\_T, [597](#)
- WARNINT
  - LPC\_WWDT\_T, [625](#)
- WB
  - \_\_WORD\_BYTE, [666](#)
- WB\_T, [665](#)
- WBVAL
  - USBD\_Core, [546](#)
- WDT\_BOD\_IRQHandler
  - cr\_startup\_lpc5411x-m0.c, [806](#)
  - cr\_startup\_lpc5411x.c, [808](#)
- WDT\_FREQ\_1000000
  - CHIP: LPC5411X Clock Driver, [81](#)
- WDT\_FREQ\_1200000
  - CHIP: LPC5411X Clock Driver, [81](#)
- WDT\_FREQ\_1300000
  - CHIP: LPC5411X Clock Driver, [81](#)
- WDT\_FREQ\_1400000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_1500000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_1600000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_1700000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_1800000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_1900000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2000000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2050000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2100000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2200000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2250000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2300000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2400000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2450000
  - CHIP: LPC5411X Clock Driver, [82](#)

- WDT\_FREQ\_2500000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2600000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2650000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2700000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2800000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2850000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2900000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_2950000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_3000000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_3050000
  - CHIP: LPC5411X Clock Driver, [82](#)
- WDT\_FREQ\_400000
  - CHIP: LPC5411X Clock Driver, [81](#)
- WDT\_FREQ\_600000
  - CHIP: LPC5411X Clock Driver, [81](#)
- WDT\_FREQ\_750000
  - CHIP: LPC5411X Clock Driver, [81](#)
- WDT\_FREQ\_900000
  - CHIP: LPC5411X Clock Driver, [81](#)
- WDT\_FREQ\_LOOKUP
  - clock\_5411x.c, [777](#)
- WDT\_FREQ\_RESERVED
  - CHIP: LPC5411X Clock Driver, [81](#)
- WDT\_OSC\_FREQ\_T
  - CHIP: LPC5411X Clock Driver, [81](#)
- WDTBOD\_IRQn
  - CHIP\_5411X: LPC5411X M0 core peripheral interrupt numbers, [428](#)
  - CHIP\_5411X: LPC5411X M4 core peripheral interrupt numbers, [430](#)
- WDTOSCCTRL
  - LPC\_SYSCON\_T, [618](#)
- WEAK
  - cr\_startup\_lpc5411x-m0.c, [805](#)
  - cr\_startup\_lpc5411x.c, [807](#)
  - LPC Public Types, [478](#)
- WINDOW
  - LPC\_WWDT\_T, [625](#)
- wIndex
  - \_USB\_SETUP\_PACKET, [664](#)
- wLength
  - \_USB\_SETUP\_PACKET, [664](#)
- wMaxPacketSize
  - \_USB\_ENDPOINT\_DESCRIPTOR, [658](#)
- WORD\_BYTE, [665](#)
- WRDATA16
  - LPC\_CRC\_T, [574](#)
- WRDATA32
  - LPC\_CRC\_T, [575](#)
- WRDATA8
  - LPC\_CRC\_T, [575](#)
- WS\_SYNC\_MASTER
  - i2s\_5411x.h, [730](#)
- WSPol
  - I2S\_AUDIO\_FORMAT\_T, [563](#)
- wTotalLength
  - \_USB\_CONFIGURATION\_DESCRIPTOR, [652](#)
  - \_USB\_OTHER\_SPEED\_CONFIGURATION, [663](#)
- wValue
  - \_USB\_SETUP\_PACKET, [664](#)
- WWDT\_WDMOD\_BITMASK
  - CHIP: LPC5411X Windowed Watchdog driver, [372](#)
- WWDT\_WDMOD\_LOCK
  - CHIP: LPC5411X Windowed Watchdog driver, [372](#)
- WWDT\_WDMOD\_WDEN
  - CHIP: LPC5411X Windowed Watchdog driver, [372](#)
- WWDT\_WDMOD\_WDINT
  - CHIP: LPC5411X Windowed Watchdog driver, [372](#)
- WWDT\_WDMOD\_WDPROTECT
  - CHIP: LPC5411X Windowed Watchdog driver, [372](#)
- WWDT\_WDMOD\_WDRESET
  - CHIP: LPC5411X Windowed Watchdog driver, [372](#)
- WWDT\_WDMOD\_WDTOF
  - CHIP: LPC5411X Windowed Watchdog driver, [372](#)
- width
  - CHIP: LPC5411X DMA Service driver, [132](#)
- WordWidth
  - I2S\_AUDIO\_FORMAT\_T, [563](#)
- write
  - CHIP: LPC5411X DMA Service driver, [132](#)
- XFERCFG
  - LPC\_DMA\_CHANNEL\_T, [575](#)
- xPSR\_Type, [666](#)
  - \_reserved0, [667](#)
  - \_reserved1, [667](#)
  - b, [667](#)
  - C, [667](#)
  - GE, [667](#)
  - ISR, [667](#)
  - IT, [667](#)
  - N, [668](#)
  - Q, [668](#)
  - T, [668](#)
  - V, [668](#)
  - w, [668](#)
  - Z, [668](#)
- xfercfg
  - DMA\_CHDESC\_T, [554](#)
- xorVal
  - PINTABLE\_T, [629](#)
- Z
  - APSR\_Type, [550](#)
  - xPSR\_Type, [668](#)