

DSP Shield Documentation

Generated by Doxygen 1.8.12

Contents

- 1 Class Index** **1**
- 1.1 Class List 1
- 2 File Index** **3**
- 2.1 File List 3
- 3 Class Documentation** **7**
- 3.1 AudioClass Class Reference 7
- 3.1.1 Detailed Description 8
- 3.1.2 Member Function Documentation 8
- 3.1.2.1 attachIntr() 8
- 3.1.2.2 Audio() [1/2] 9
- 3.1.2.3 Audio() [2/2] 9
- 3.1.2.4 audioMute() 10
- 3.1.2.5 audioUnmute() 10
- 3.1.2.6 close() 10
- 3.1.2.7 detachIntr() 11
- 3.1.2.8 HPL_RConF_Routing() 11
- 3.1.2.9 HPR_RConF_Routing() 11
- 3.1.2.10 isrDma() 12
- 3.1.2.11 LOL_RConF_Routing() 12
- 3.1.2.12 LOR_RConF_Routing() 13
- 3.1.2.13 read() 13
- 3.1.2.14 setInputGain() 13

3.1.2.15	setOutputVolume() [1/2]	14
3.1.2.16	setOutputVolume() [2/2]	14
3.1.2.17	setSamplingRate()	15
3.1.2.18	write()	15
3.1.3	Member Data Documentation	15
3.1.3.1	activeInBuf	15
3.1.3.2	activeOutBuf	16
3.1.3.3	audioInLeft	16
3.1.3.4	audioInRight	16
3.1.3.5	audioOutLeft	16
3.1.3.6	audioOutRight	16
3.1.3.7	sampleLeft	16
3.1.3.8	sampleRight	16
3.2	BR_struct Struct Reference	17
3.2.1	Detailed Description	18
3.2.2	Member Data Documentation	18
3.2.2.1	byte10_vol_label	18
3.2.2.2	byte11_vol_label	18
3.2.2.3	byte1_extended_no_of_sectors_on_partition	18
3.2.2.4	byte1_fs_type	18
3.2.2.5	byte1_no_hidden_sectors	19
3.2.2.6	byte1_vol_label	19
3.2.2.7	byte2_extended_no_of_sectors_on_partition	19
3.2.2.8	byte2_fs_type	19
3.2.2.9	byte2_no_hidden_sectors	19
3.2.2.10	byte2_vol_label	19
3.2.2.11	byte3_extended_no_of_sectors_on_partition	19
3.2.2.12	byte3_fs_type	19
3.2.2.13	byte3_no_hidden_sectors	20
3.2.2.14	byte3_vol_label	20

3.2.2.15	byte4_extended_no_of_sectors_on_partition	20
3.2.2.16	byte4_fs_type	20
3.2.2.17	byte4_no_hidden_sectors	20
3.2.2.18	byte4_vol_label	20
3.2.2.19	byte5_fs_type	20
3.2.2.20	byte5_vol_label	20
3.2.2.21	byte6_fs_type	21
3.2.2.22	byte6_vol_label	21
3.2.2.23	byte7_fs_type	21
3.2.2.24	byte7_vol_label	21
3.2.2.25	byte8_fs_type	21
3.2.2.26	byte8_vol_label	21
3.2.2.27	byte9_vol_label	21
3.2.2.28	drive_number	21
3.2.2.29	extended_boot_signature	22
3.2.2.30	LB_bytes_per_sector	22
3.2.2.31	LB_no_of_heads	22
3.2.2.32	LB_no_of_root_dir_entries	22
3.2.2.33	LB_no_of_sectors_on_partition	22
3.2.2.34	LB_reserved_sectors	22
3.2.2.35	LB_sectors_per_fat	22
3.2.2.36	LB_sectors_per_track	22
3.2.2.37	media_descriptor	23
3.2.2.38	no_of_fats	23
3.2.2.39	oem_name_byte_1	23
3.2.2.40	oem_name_byte_2	23
3.2.2.41	oem_name_byte_3	23
3.2.2.42	oem_name_byte_4	23
3.2.2.43	oem_name_byte_5	23
3.2.2.44	oem_name_byte_6	23

3.2.2.45	oem_name_byte_7	24
3.2.2.46	oem_name_byte_8	24
3.2.2.47	opt_partition_boot_code	24
3.2.2.48	reserved	24
3.2.2.49	sectors_per_cluster	24
3.2.2.50	short_jump_instr_byte_1	24
3.2.2.51	short_jump_instr_byte_2	24
3.2.2.52	short_jump_instr_byte_3	24
3.2.2.53	signature	25
3.2.2.54	UB_bytes_per_sector	25
3.2.2.55	UB_no_of_heads	25
3.2.2.56	UB_no_of_root_dir_entries	25
3.2.2.57	UB_no_of_sectors_on_partition	25
3.2.2.58	UB_reserved_sectors	25
3.2.2.59	UB_sectors_per_fat	25
3.2.2.60	UB_sectors_per_track	25
3.2.2.61	vol_serial_number_byte_1	26
3.2.2.62	vol_serial_number_byte_2	26
3.2.2.63	vol_serial_number_byte_3	26
3.2.2.64	vol_serial_number_byte_4	26
3.3	FFTClass Class Reference	26
3.3.1	Detailed Description	26
3.3.2	Member Function Documentation	27
3.3.2.1	FFT_filter()	27
3.3.2.2	FFT_init()	27
3.4	File Class Reference	28
3.4.1	Detailed Description	29
3.4.2	Constructor & Destructor Documentation	29
3.4.2.1	File() [1/2]	29
3.4.2.2	File() [2/2]	30

3.4.3	Member Function Documentation	30
3.4.3.1	available()	30
3.4.3.2	close()	30
3.4.3.3	flush()	31
3.4.3.4	getName()	31
3.4.3.5	isDirectory()	31
3.4.3.6	openNextFile()	31
3.4.3.7	operator bool()	32
3.4.3.8	peek()	32
3.4.3.9	position()	32
3.4.3.10	print() [1/7]	32
3.4.3.11	print() [2/7]	33
3.4.3.12	print() [3/7]	33
3.4.3.13	print() [4/7]	33
3.4.3.14	print() [5/7]	34
3.4.3.15	print() [6/7]	34
3.4.3.16	print() [7/7]	34
3.4.3.17	println() [1/8]	35
3.4.3.18	println() [2/8]	35
3.4.3.19	println() [3/8]	35
3.4.3.20	println() [4/8]	36
3.4.3.21	println() [5/8]	36
3.4.3.22	println() [6/8]	36
3.4.3.23	println() [7/8]	37
3.4.3.24	println() [8/8]	37
3.4.3.25	read() [1/3]	38
3.4.3.26	read() [2/3]	38
3.4.3.27	read() [3/3]	38
3.4.3.28	rewindDirectory()	39
3.4.3.29	seek()	39

3.4.3.30	size()	39
3.4.3.31	write() [1/5]	39
3.4.3.32	write() [2/5]	40
3.4.3.33	write() [3/5]	40
3.4.3.34	write() [4/5]	40
3.4.3.35	write() [5/5]	41
3.4.4	Friends And Related Function Documentation	41
3.4.4.1	SD_Class	41
3.5	fileNodesList Class Reference	41
3.5.1	Detailed Description	42
3.5.2	Constructor & Destructor Documentation	42
3.5.2.1	fileNodesList()	42
3.5.3	Member Data Documentation	42
3.5.3.1	nextFileNode	42
3.5.3.2	startCluster	42
3.6	FONT_CHAR_INFO Struct Reference	42
3.6.1	Detailed Description	43
3.6.2	Member Data Documentation	43
3.6.2.1	heightBits	43
3.6.2.2	offset	43
3.6.2.3	widthBits	43
3.7	FONT_INFO Struct Reference	43
3.7.1	Detailed Description	44
3.7.2	Member Data Documentation	44
3.7.2.1	charInfo	44
3.7.2.2	data	44
3.7.2.3	endChar	44
3.7.2.4	heightPages	44
3.7.2.5	startChar	44
3.8	GPT_Config Struct Reference	44

3.8.1	Detailed Description	45
3.8.2	Member Data Documentation	45
3.8.2.1	autoLoad	45
3.8.2.2	ctrlTim	45
3.8.2.3	prdHigh	45
3.8.2.4	prdLow	45
3.8.2.5	preScaleDiv	45
3.9	MBR_struct Struct Reference	46
3.9.1	Detailed Description	46
3.9.2	Member Data Documentation	46
3.9.2.1	mbr_code	46
3.9.2.2	mmc_id_entry	46
3.9.2.3	partition_four	46
3.9.2.4	partition_one	46
3.9.2.5	partition_three	47
3.9.2.6	partition_two	47
3.9.2.7	signature	47
3.10	OLED Class Reference	47
3.10.1	Detailed Description	48
3.10.2	Member Function Documentation	48
3.10.2.1	autoscroll()	48
3.10.2.2	begin()	48
3.10.2.3	clear() [1/2]	49
3.10.2.4	clear() [2/2]	49
3.10.2.5	display()	49
3.10.2.6	flip()	50
3.10.2.7	init()	50
3.10.2.8	noAutoscroll()	50
3.10.2.9	noDisplay()	51
3.10.2.10	oledInit()	51

3.10.2.11 print() [1/4]	51
3.10.2.12 print() [2/4]	52
3.10.2.13 print() [3/4]	52
3.10.2.14 print() [4/4]	52
3.10.2.15 printchar()	53
3.10.2.16 resetCursor()	53
3.10.2.17 scrollDisplayLeft() [1/2]	53
3.10.2.18 scrollDisplayLeft() [2/2]	54
3.10.2.19 scrollDisplayRight() [1/2]	54
3.10.2.20 scrollDisplayRight() [2/2]	54
3.10.2.21 setline()	55
3.10.2.22 setOrientation()	55
3.10.2.23 setRolling()	55
3.10.2.24 write()	56
3.11 PARTITION_TABLE Struct Reference	56
3.11.1 Detailed Description	57
3.11.2 Member Data Documentation	57
3.11.2.1 boot_descriptor	57
3.11.2.2 byte1_first_sector_position	57
3.11.2.3 byte1_no_of_sectors_in_partition	57
3.11.2.4 byte2_first_sector_position	57
3.11.2.5 byte2_no_of_sectors_in_partition	57
3.11.2.6 byte3_first_sector_position	57
3.11.2.7 byte3_no_of_sectors_in_partition	58
3.11.2.8 byte4_first_sector_position	58
3.11.2.9 byte4_no_of_sectors_in_partition	58
3.11.2.10 fs_descriptor	58
3.11.2.11 partition_end_cylinder	58
3.11.2.12 partition_end_head	58
3.11.2.13 partition_end_sector	58

3.11.2.14 partition_start_cylinder	58
3.11.2.15 partition_start_head	59
3.11.2.16 partition_start_sector	59
3.12 Pipe_Config Class Reference	59
3.12.1 Detailed Description	59
3.12.2 Member Data Documentation	59
3.12.2.1 maxPktSize	59
3.12.2.2 xferType	59
3.13 PLL_Class Class Reference	60
3.13.1 Detailed Description	60
3.13.2 Constructor & Destructor Documentation	60
3.13.2.1 PLL_Class()	60
3.13.3 Member Function Documentation	60
3.13.3.1 configure()	60
3.13.3.2 getConfigure()	60
3.13.3.3 getSystemClock()	61
3.14 SAR_Class Class Reference	61
3.14.1 Detailed Description	61
3.14.2 Member Function Documentation	61
3.14.2.1 begin()	61
3.14.2.2 configChannel()	62
3.14.2.3 end()	62
3.14.2.4 getStatus()	63
3.14.2.5 readData()	63
3.14.2.6 startConversion()	63
3.14.2.7 stopConversion()	64
3.15 SD_Class Class Reference	64
3.15.1 Detailed Description	65
3.15.2 Constructor & Destructor Documentation	65
3.15.2.1 SD_Class()	65

3.15.2.2	~SD_Class()	65
3.15.3	Member Function Documentation	65
3.15.3.1	begin() [1/2]	65
3.15.3.2	begin() [2/2]	66
3.15.3.3	exists()	66
3.15.3.4	mkdir()	66
3.15.3.5	open() [1/2]	67
3.15.3.6	open() [2/2]	67
3.15.3.7	remove()	67
3.15.3.8	rmdir()	68
3.16	SPI_Class Class Reference	68
3.16.1	Detailed Description	68
3.16.2	Member Function Documentation	69
3.16.2.1	begin()	69
3.16.2.2	end()	69
3.16.2.3	read()	69
3.16.2.4	setBitOrder()	70
3.16.2.5	setClockDivider()	70
3.16.2.6	setDataMode()	70
3.16.2.7	setLoopBackMode()	71
3.16.2.8	transfer()	71
3.16.2.9	write()	71
3.17	TimerClass Class Reference	72
3.17.1	Detailed Description	72
3.17.2	Member Function Documentation	72
3.17.2.1	clearInterrupt()	72
3.17.2.2	close() [1/2]	73
3.17.2.3	close() [2/2]	73
3.17.2.4	closeWdt()	73
3.17.2.5	configTimer() [1/2]	73

3.17.2.6	configTimer() [2/2]	73
3.17.2.7	initialize() [1/2]	74
3.17.2.8	initialize() [2/2]	74
3.17.2.9	read() [1/2]	74
3.17.2.10	read() [2/2]	74
3.17.2.11	selectTimer()	74
3.17.2.12	serviceWdt()	75
3.17.2.13	setPeriod() [1/2]	75
3.17.2.14	setPeriod() [2/2]	75
3.17.2.15	setWdt()	75
3.17.2.16	start() [1/2]	75
3.17.2.17	start() [2/2]	76
3.17.2.18	startWdt()	76
3.17.2.19	stop() [1/2]	76
3.17.2.20	stop() [2/2]	76
3.17.2.21	stopWdt()	76
3.18	USB_Config Class Reference	77
3.18.1	Detailed Description	77
3.18.2	Member Data Documentation	77
3.18.2.1	appSuspendCallBack	77
3.18.2.2	cfgDescFSPtr	77
3.18.2.3	cfgDescPtr	77
3.18.2.4	deviceDescPtr	77
3.18.2.5	deviceQualDescPtr	78
3.18.2.6	opMode	78
3.18.2.7	rxCompleteCallback	78
3.18.2.8	rxIntCallback	78
3.18.2.9	strDescPtr	78
3.18.2.10	txIntCallback	78
3.19	USBClass Class Reference	78

3.19.1 Detailed Description	79
3.19.2 Constructor & Destructor Documentation	79
3.19.2.1 USBClass()	79
3.19.3 Member Function Documentation	80
3.19.3.1 clearPipe()	80
3.19.3.2 closePipe()	80
3.19.3.3 config()	81
3.19.3.4 configPipe()	81
3.19.3.5 connect()	82
3.19.3.6 disconnect()	82
3.19.3.7 dmaRxStop()	82
3.19.3.8 dmaTxStop()	83
3.19.3.9 handleInterrupts()	83
3.19.3.10 init()	84
3.19.3.11 readPipe()	84
3.19.3.12 requestPipe()	85
3.19.3.13 setParams()	85
3.19.3.14 stallPipe()	86
3.19.3.15 suspend()	86
3.19.3.16 writePipe()	87
3.19.4 Member Data Documentation	87
3.19.4.1 usbBuffTxRxPtr1	87
3.19.4.2 usbBuffTxRxPtr2	87
3.19.4.3 usbBuffTxRxPtr3	87
3.19.4.4 usbBuffTxRxPtr4	88
3.20 WDT_Config Struct Reference	88
3.20.1 Detailed Description	88
3.20.2 Member Data Documentation	88
3.20.2.1 counter	88
3.20.2.2 prescale	88

4	File Documentation	89
4.1	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/AnalogReadWrite/examples/↵ AnalogRead/AnalogRead.ino File Reference	89
4.2	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/Audio.cpp File Reference	89
4.2.1	Detailed Description	89
4.2.2	Variable Documentation	89
4.2.2.1	AudioC	89
4.2.2.2	i2sDmaLeftBuff1	90
4.2.2.3	i2sDmaLeftBuff2	90
4.2.2.4	i2sDmaRightBuff1	90
4.2.2.5	i2sDmaRightBuff2	90
4.3	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/Audio.h File Reference	90
4.3.1	Detailed Description	91
4.3.2	Macro Definition Documentation	91
4.3.2.1	CHANNEL_MONO	91
4.3.2.2	CHANNEL_STEREO	91
4.3.2.3	DMA_CHAN_ReadL	91
4.3.2.4	DMA_CHAN_ReadR	92
4.3.2.5	DMA_CHAN_WriteL	92
4.3.2.6	DMA_CHAN_WriteR	92
4.3.2.7	I2S_DMA_BUF_LEN	92
4.3.2.8	SAMPLING_RATE_11_KHZ	92
4.3.2.9	SAMPLING_RATE_12_KHZ	92
4.3.2.10	SAMPLING_RATE_16_KHZ	92
4.3.2.11	SAMPLING_RATE_22_KHZ	92
4.3.2.12	SAMPLING_RATE_24_KHZ	93
4.3.2.13	SAMPLING_RATE_32_KHZ	93
4.3.2.14	SAMPLING_RATE_44_KHZ	93
4.3.2.15	SAMPLING_RATE_48_KHZ	93
4.3.2.16	SAMPLING_RATE_8_KHZ	93

4.3.3	Variable Documentation	93
4.3.3.1	AudioC	93
4.4	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Audio↔ Loopback/AudioLoopback.ino File Reference	94
4.5	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Filter/↔ FIR/FIR.ino File Reference	94
4.6	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Filter/↔ IIR/IIR.ino File Reference	94
4.7	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/↔ Player/Player.ino File Reference	94
4.8	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/↔ Recorder/Recorder.ino File Reference	94
4.9	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DigitalReadWrite/examples/↔ DigitalWrite/DigitalWrite.ino File Reference	94
4.10	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DigitalReadWrite/examples/↔ GPIO/GPIO.ino File Reference	94
4.11	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DigitalReadWrite/examples/read↔ DIPSwitch/readDIPSwitch.ino File Reference	94
4.12	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DMA/examples/DMA/↔ DMA.ino File Reference	94
4.13	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/examples/dma/dma.ino File Reference	94
4.14	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/coeffdef_fft.h File Reference	94
4.14.1	Macro Definition Documentation	95
4.14.1.1	FFTLLEN	95
4.14.1.2	NX	95
4.14.1.3	WINLEN	95
4.14.2	Variable Documentation	95
4.14.2.1	fr	95
4.14.2.2	ir	95
4.14.2.3	x_filter	96
4.15	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/examples/FFT/F↔ FT.ino File Reference	96
4.16	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/FFT.cpp File Refer- ence	96

4.16.1 Detailed Description	96
4.16.2 Macro Definition Documentation	96
4.16.2.1 ENABLE_FILTER	96
4.16.3 Function Documentation	97
4.16.3.1 CPLX_Mul()	97
4.16.3.2 downScaleData()	97
4.16.3.3 IR2FR()	98
4.16.3.4 upScaleData()	98
4.16.4 Variable Documentation	98
4.16.4.1 buf	98
4.16.4.2 FFTransform	99
4.17 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/FFT.h File Reference	99
4.17.1 Detailed Description	99
4.17.2 Macro Definition Documentation	100
4.17.2.1 CONVERT_32_TO_16_MASK	100
4.17.2.2 CONVERT_32_TO_16_SHIFT	100
4.17.2.3 DISABLE_SCALE	100
4.17.2.4 ENABLE_SCALE	100
4.17.2.5 FFT_BAD_PARAMS	100
4.17.2.6 FFT_COMPLEX	100
4.17.2.7 FFT_REAL	100
4.17.2.8 FFT_SUCCESS	101
4.17.2.9 FFTQ15	101
4.17.2.10 FFTQ31	101
4.17.2.11 Q15_MULT_SHIFT	101
4.17.3 Variable Documentation	101
4.17.3.1 FFTransform	101
4.18 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Interrupt/examples/dma↵ Interrupt/dmaInterrupt.ino File Reference	101
4.19 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Interrupt/examples/io↵ ExpanderInterrupt/ioExpanderInterrupt.ino File Reference	101

4.20	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/bitmapDb.h File Reference	101
4.20.1	Detailed Description	102
4.21	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/examples/Oled↔Basic/OledBasic.ino File Reference	102
4.22	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/examples/Oled↔Print/OledPrint.ino File Reference	102
4.23	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/examples/Oled↔SerialControl/OledSerialControl.ino File Reference	102
4.24	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/examples/Oled↔SerialMsg/OledSerialMsg.ino File Reference	102
4.25	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/OLED.cpp File Reference	102
4.25.1	Detailed Description	102
4.25.2	Variable Documentation	103
4.25.2.1	disp	103
4.26	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/OLED.h File Reference	103
4.26.1	Detailed Description	103
4.26.2	Typedef Documentation	103
4.26.2.1	NUMBER_BASE	103
4.26.3	Enumeration Type Documentation	103
4.26.3.1	NUMBER_BASE	103
4.26.4	Variable Documentation	104
4.26.4.1	disp	104
4.27	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/temp.h File Reference	104
4.27.1	Function Documentation	104
4.27.1.1	loop()	104
4.27.1.2	oled_test()	104
4.27.1.3	setup()	104
4.28	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/PLL/pll.cpp File Reference	105
4.28.1	Detailed Description	105
4.28.2	Variable Documentation	105

4.28.2.1	PLL	105
4.29	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/PLL/pll.h File Reference	105
4.29.1	Detailed Description	106
4.29.2	Variable Documentation	106
4.29.2.1	PLL	106
4.30	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/RTC/examples/RTC/RTCino File Reference	106
4.31	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SAR/SAR.cpp File Reference	106
4.31.1	Detailed Description	106
4.31.2	Function Documentation	107
4.31.2.1	SAR_LOG_MSG_PRINT() [1/2]	107
4.31.2.2	SAR_LOG_MSG_PRINT() [2/2]	107
4.31.3	Variable Documentation	108
4.31.3.1	SAR	108
4.32	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SAR/SAR.h File Reference	108
4.32.1	Detailed Description	108
4.32.2	Macro Definition Documentation	109
4.32.2.1	SAR_CHANNEL0	109
4.32.2.2	SAR_CHANNEL1	109
4.32.2.3	SAR_CHANNEL2	109
4.32.2.4	SAR_CHANNEL3	109
4.32.2.5	SAR_CHANNEL4	109
4.32.2.6	SAR_CHANNEL5	109
4.32.2.7	SAR_CONTINUOUS_CONVERSION	109
4.32.2.8	SAR_DMA_MODE	110
4.32.2.9	SAR_INTERRUPT_MODE	110
4.32.2.10	SAR_POLL_MODE	110
4.32.2.11	SAR_SINGLE_CONVERSION	110
4.32.3	Variable Documentation	110
4.32.3.1	SAR	110

4.33	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/chk_mmc.c File Reference	110
4.33.1	Function Documentation	111
4.33.1.1	Check_boot_record()	111
4.33.1.2	chk_mmc()	111
4.33.1.3	getMMCSize()	111
4.33.2	Variable Documentation	111
4.33.2.1	boot_record	111
4.33.2.2	master_boot_record	112
4.34	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/chk_mmc.h File Reference	112
4.34.1	Detailed Description	112
4.34.2	Macro Definition Documentation	113
4.34.2.1	FD_BOOT_RECORD	113
4.34.2.2	HD_BOOT_RECORD	113
4.34.3	Enumeration Type Documentation	113
4.34.3.1	MMC_ERR_int16	113
4.34.4	Function Documentation	113
4.34.4.1	Check_boot_record()	113
4.34.4.2	chk_mmc()	114
4.34.4.3	getMMCSize()	114
4.34.4.4	mmc_format()	114
4.35	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Display↵ Contents/DisplayContents.ino File Reference	115
4.36	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Exists↵ Exists.ino File Reference	115
4.37	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/File↵ FolderCreation/FileFolderCreation.ino File Reference	115
4.38	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Filesize↵ Filesize.ino File Reference	115
4.39	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Peek↵ Peek.ino File Reference	115
4.40	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Position↵ Position.ino File Reference	115

4.41	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Print/↔ Print.ino File Reference	115
4.42	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Print/↔ Print.ino File Reference	115
4.43	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Read↔ Write/ReadWrite.ino File Reference	115
4.44	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.cpp File Reference	115
4.44.1	Detailed Description	116
4.44.2	Macro Definition Documentation	116
4.44.2.1	CSL_MMCSA_AT_A_BUF_SIZE	116
4.44.2.2	CSL_SD_CLOCK_MAX_KHZ	116
4.44.2.3	WRITE_CACHE_SIZE	116
4.44.3	Function Documentation	117
4.44.3.1	LOG_MSG_print()	117
4.44.4	Variable Documentation	117
4.44.4.1	AtaWrBuf	117
4.44.4.2	fileListHeadNode	117
4.44.4.3	fileListLastNode	117
4.44.4.4	SD	117
4.44.4.5	writeCacheBuffer	117
4.45	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.h File Reference	118
4.45.1	Detailed Description	118
4.45.2	Typedef Documentation	119
4.45.2.1	FILE_MODE	119
4.45.2.2	NUMBER_FORMAT_BASE	119
4.45.3	Enumeration Type Documentation	119
4.45.3.1	FILE_MODE	119
4.45.3.2	NUMBER_FORMAT_BASE	119
4.45.4	Variable Documentation	119
4.45.4.1	SD	119
4.46	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Find/↔ Find.ino File Reference	120

4.47	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Finduntil/Finduntil.ino File Reference	120
4.48	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/ParseFloat/ParseFloat.ino File Reference	120
4.49	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/ParseInt/ParseInt.ino File Reference	120
4.50	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/PrintFormat/PrintFormat.ino File Reference	120
4.51	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Read/Read.ino File Reference	120
4.52	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/ReadBytes/ReadBytes.ino File Reference	120
4.53	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/ReadBytesUntil/ReadBytesUntil.ino File Reference	120
4.54	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/examples/Arduino_SPI/Arduino_SPI.ino File Reference	120
4.55	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/examples/DSP_SPI/DSP_SPI.ino File Reference	120
4.56	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/SPI.cpp File Reference	120
4.56.1	Detailed Description	121
4.56.2	Function Documentation	121
4.56.2.1	SPI_LOG_MSG_PRINT()	121
4.56.3	Variable Documentation	122
4.56.3.1	SPI	122
4.57	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/SPI.h File Reference	122
4.57.1	Detailed Description	122
4.57.2	Macro Definition Documentation	123
4.57.2.1	LSBFIRST	123
4.57.2.2	MSBFIRST	123
4.57.2.3	SPI_CLK_DIV	123
4.57.2.4	SPI_CLOCK_DIV128	123
4.57.2.5	SPI_CLOCK_DIV16	123
4.57.2.6	SPI_CLOCK_DIV2	123
4.57.2.7	SPI_CLOCK_DIV32	123

4.57.2.8	SPI_CLOCK_DIV4	124
4.57.2.9	SPI_CLOCK_DIV64	124
4.57.2.10	SPI_CLOCK_DIV8	124
4.57.2.11	SPI_FRAME_LENGTH	124
4.57.2.12	SPI_MODE0	124
4.57.2.13	SPI_MODE1	124
4.57.2.14	SPI_MODE2	124
4.57.2.15	SPI_MODE3	124
4.57.3	Variable Documentation	125
4.57.3.1	SPI	125
4.58	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/↔ Timer1/Timer1.ino File Reference	125
4.59	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/↔ Timer2/Timer2.ino File Reference	125
4.60	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/↔ Timer3/Timer3.ino File Reference	125
4.61	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/↔ WatchdogTimer/WatchdogTimer.ino File Reference	125
4.62	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/Timers.cpp File Reference	125
4.62.1	Detailed Description	126
4.62.2	Variable Documentation	126
4.62.2.1	gptObj	126
4.62.2.2	hGpt	126
4.62.2.3	hWdt	126
4.62.2.4	Timer	126
4.62.2.5	wdtObj	126
4.63	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/Timers.h File Reference	126
4.63.1	Detailed Description	127
4.63.2	Macro Definition Documentation	127
4.63.2.1	GPT0	127
4.63.2.2	GPT1	128

4.63.2.3	GPT2	128
4.63.2.4	GPT_PRE_SC_DIV_0	128
4.63.2.5	GPT_PRE_SC_DIV_1	128
4.63.2.6	GPT_PRE_SC_DIV_10	128
4.63.2.7	GPT_PRE_SC_DIV_11	128
4.63.2.8	GPT_PRE_SC_DIV_12	128
4.63.2.9	GPT_PRE_SC_DIV_2	128
4.63.2.10	GPT_PRE_SC_DIV_3	129
4.63.2.11	GPT_PRE_SC_DIV_4	129
4.63.2.12	GPT_PRE_SC_DIV_5	129
4.63.2.13	GPT_PRE_SC_DIV_6	129
4.63.2.14	GPT_PRE_SC_DIV_7	129
4.63.2.15	GPT_PRE_SC_DIV_8	129
4.63.2.16	GPT_PRE_SC_DIV_9	129
4.63.2.17	HIGH_WORD_MASK	129
4.63.2.18	LOW_WORD_MASK	130
4.63.2.19	TIMER_INSTANCE_COUNT	130
4.63.2.20	WORD_LENGTH	130
4.63.3	Variable Documentation	130
4.63.3.1	Timer	130
4.64	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/examples/interrupt/interrupt.ino File Reference	130
4.65	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.cpp File Reference	130
4.65.1	Detailed Description	131
4.65.2	Function Documentation	131
4.65.2.1	completeTransferCallback()	131
4.65.2.2	startTransferCallback()	131
4.65.3	Variable Documentation	132
4.65.3.1	gUsbConfig	132
4.65.3.2	hpdRx	132

4.65.3.3	hpdtx	132
4.65.3.4	hUsbDev	132
4.65.3.5	linking_ram0	132
4.65.3.6	sentLongEp0Pkt	132
4.65.3.7	USB	132
4.65.3.8	usbDataBufferTxRx1	133
4.65.3.9	usbDataBufferTxRx2	133
4.65.3.10	usbDataBufferTxRx3	133
4.65.3.11	usbDataBufferTxRx4	133
4.66	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.h File Reference	133
4.66.1	Detailed Description	134
4.66.2	Macro Definition Documentation	135
4.66.2.1	USB_DATA_SIZE	135
4.66.2.2	USB_EP0	135
4.66.2.3	USB_EP1	135
4.66.2.4	USB_EP2	135
4.66.2.5	USB_EP3	135
4.66.2.6	USB_EP4	135
4.66.2.7	USB_HS_MAX_PACKET_SIZE	135
4.66.2.8	USB_IN	136
4.66.2.9	USB_IN_EP0	136
4.66.2.10	USB_IN_EP1	136
4.66.2.11	USB_IN_EP2	136
4.66.2.12	USB_IN_EP3	136
4.66.2.13	USB_IN_EP4	136
4.66.2.14	USB_LRAM_SIZE	136
4.66.2.15	USB_MAX_CURRENT	136
4.66.2.16	USB_OUT	137
4.66.2.17	USB_OUT_EP0	137
4.66.2.18	USB_OUT_EP1	137

4.66.2.19	USB_OUT_EP2	137
4.66.2.20	USB_OUT_EP3	137
4.66.2.21	USB_OUT_EP4	137
4.66.2.22	USB_PIPE_COUNT	137
4.66.2.23	USB_STRDESC_COUNT	137
4.66.2.24	USB_WAKEUP_DELAY	138
4.66.3	Typedef Documentation	138
4.66.3.1	USB_APP_CALLBACK	138
4.66.3.2	USB_APP_INT_CALLBACK	138
4.66.3.3	USB_opMode	138
4.66.3.4	USB_pipeHandle	138
4.66.3.5	USB_xferType	138
4.66.4	Enumeration Type Documentation	138
4.66.4.1	USB_opMode	138
4.66.4.2	USB_xferType	139
4.66.5	Variable Documentation	139
4.66.5.1	USB	139
4.67	/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Wire/examples/Wire/↔ Wire.ino File Reference	139

Chapter 1

Class Index

1.1 Class List

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

AudioClass		
Audio Class	7
BR_struct	17
FFTClass		
FFT Class	26
File		
File Class	28
fileNodesList		
Node of the Linked List to hold the details of all the Opened files	41
FONT_CHAR_INFO		
This structure describes a single character's display information	42
FONT_INFO		
This structure describes a single font	43
GPT_Config		
Configuration structure	44
MBR_struct	46
OLED		
OLED Class	47
PARTITION_TABLE	56
Pipe_Config		
Pipe configuration Class	59
PLL_Class		
PLL Class	60
SAR_Class		
SAR Class	61
SD_Class		
SD Card Class	64
SPI_Class		
SPI Class	68
TimerClass		
Timers Class	72
USB_Config		
USB configuration Class	77
USBClass		
USB Class	78
WDT_Config		
Configuration structure	88

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/AnalogReadWrite/examples/↔ AnalogRead/ AnalogRead.ino	89
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/ Audio.cpp Audio implementation	89
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/ Audio.h Audio library header file	90
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Audio↔ Loopback/ AudioLoopback.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Filter/FIR/ FIR.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Filter/IIR/ IIR.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Player/ Player.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Recorder/ Recorder.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DigitalReadWrite/examples/↔ DigitalWrite/ DigitalWrite.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DigitalReadWrite/examples/↔ GPIO/ GPIO.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DigitalReadWrite/examples/read↔ DIPSwitch/ readDIPSwitch.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DMA/examples/DMA/ DMA.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/ coeffdef_fft.h	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/ FFT.cpp FFT implementation	96
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/ FFT.h FFT library header file	99
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/examples/FFT/ FFT.ino	96
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Interrupt/examples/dma↔ Interrupt/ dmaInterrupt.ino	101
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Interrupt/examples/ioExpander↔ Interrupt/ ioExpanderInterrupt.ino	101
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/ bitmapDb.h Bitmap database for OLED module	101

/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/OLED.cpp	
OLED implementation	102
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/OLED.h	
OLED library header file	103
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/temp.h	104
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/examples/Oled↔	
Basic/ OledBasic.ino	102
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/examples/Oled↔	
Print/ OledPrint.ino	102
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/examples/OledSerial↔	
Control/ OledSerialControl.ino	102
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/examples/OledSerial↔	
Msg/ OledSerialMsg.ino	102
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/PLL/pll.cpp	
PLL implementation	105
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/PLL/pll.h	
PLL library header file	105
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/RTC/examples/RTC/RTC.ino	106
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SAR/SAR.cpp	
SAR implementation	106
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SAR/SAR.h	
SAR library header file	108
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/chk_mmc.c	110
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/chk_mmc.h	
Header file for MMC checking routines	112
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.cpp	
SD/File implementation	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.h	
SD library header file	118
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/DisplayContents/Display↔	
Contents.ino	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Exists/Exists.ino	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/FileFolder↔	
Creation/ FileFolderCreation.ino	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Filesize/Filesize.↔	
ino	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Peek/Peek.ino	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Position/Position.↔	
ino	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Print/Print.ino	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/ReadWrite/Read↔	
Write.ino	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Find/Find.ino	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Finduntil/Finduntil.↔	
ino	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Parse↔	
Float/ ParseFloat.ino	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Parse↔	
Int/ ParseInt.ino	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Print/Print.ino	115
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/PrintFormat/Print↔	
Format.ino	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Read/Read.↔	
ino	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Read↔	
Bytes/ ReadBytes.ino	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/ReadBytes↔	
Until/ ReadBytesUntil.ino	120

/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/SPI.cpp	
SPI implementation	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/SPI.h	
SPI library header file	122
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/examples/Arduino_SPI.ino	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/examples/DSP_SPI_DSPI.ino	120
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/Timers.cpp	
Timers implementation	125
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/Timers.h	
Timers library header file	126
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/Timer1/Timer1.ino	125
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/Timer2/Timer2.ino	125
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/Timer3/Timer3.ino	125
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/WatchdogTimer/WatchdogTimer.ino	125
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.cpp	
USB implementation	130
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.h	
USB library header file	133
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/examples/dma/dma.ino	94
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/examples/interrupt/interrupt.ino	130
/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Wire/examples/Wire/Wire.ino	139

Chapter 3

Class Documentation

3.1 AudioClass Class Reference

Audio Class.

```
#include <Audio.h>
```

Public Member Functions

- int [Audio](#) (void)
- int [Audio](#) (int process)
- int [close](#) ()
- void [attachIntr](#) (void *function)
- void [detachIntr](#) (void)
- int [read](#) (void)
- int [write](#) (void)
- void [isrDma](#) (void)
- int [setInputGain](#) (int lgain, int rgain)
- int [setOutputVolume](#) (int volume)
- int [setOutputVolume](#) (int lvolume, int rvolume)
- int [audioMute](#) (void)
- int [audioUnmute](#) (void)
- int [setSamplingRate](#) (long)
- int [HPL_RConF_Routing](#) (int left)
- int [HPR_RConF_Routing](#) (int left, int right)
- int [LOL_RConF_Routing](#) (int left, int right)
- int [LOR_RConF_Routing](#) (int right)

Public Attributes

- Uint16 * [audioInLeft](#) [2]
- Uint16 * [audioInRight](#) [2]
- Uint16 * [audioOutLeft](#) [2]
- Uint16 * [audioOutRight](#) [2]
- unsigned short [activeInBuf](#)
- unsigned short [activeOutBuf](#)
- int [sampleLeft](#)
- int [sampleRight](#)

3.1.1 Detailed Description

Audio Class.

Contains prototypes for functions in Audio library

Address of audio data buffers is stored in the below pointers
 audioInLeft - Holds the adress of Audio input (read from codec) data buffers for left channel
 audioInRight - Holds the adress of Audio input (read from codec) data buffers for right channel
 audioOutLeft - Holds the adress of Audio output (write to codec) data buffers for left channel
 audioInRight - Holds the adress of Audio output (write to codec) data buffers for right channel

All the components described above are array of two pointers pointing to the two data buffers of given audio channel. When one data buffer is being used by audio module, other buffer can be used for data processing.

Index of the buffers being used by audio module are indicated by activeInBuf - Index of the buffer being used for audio input
 audioInLeft[activeInBuf] indcates left channel buffer
 audioInRight[activeInBuf] indcates right channel buffer
 activeOutBuf - Index of the buffer being used for audio output
 audioOutLeft[activeOutBuf] indcates left channel buffer
 audioOutRight[activeOutBuf] indcates right channel buffer

Data read from codec can accessed using the data buffer pointers 'audioInLeft', audioInRight and index 'activeInBuf'

When Audio library is configured to operate with same data buffers for read and write operation (initialized using [Audio\(void\)](#)), audioInLeft and audioOutLeft points to same buffers and audioInRight and audioInRight points to same buffers.

When Audio library is configured to operate with independent data buffers for read and write operation (initialized using [Audio\(int process\)](#)), applications need to take care of copying the data from audio input buffers (audioInLeftxx) to audio output buffers (audioOutxx) in DMA ISR.

3.1.2 Member Function Documentation

3.1.2.1 attachIntr()

```
void AudioClass::attachIntr (
    void * function )
```

attachInterrupt

Description

Function to assign ISR fot DMA interrupt.

Arguments

* function - ISR function pointer

Return Value

None

3.1.2.2 Audio() [1/2]

```
int AudioClass::Audio (
    void )
```

Audio

Description

Function initializes Audio Module.

This function is for enabling the audio loopback for routing data received from audio input to the audio output without much processing of the audio data received. Calling this function enables audio library to use same data buffers for audio input and output channels. While audio input process is filling one buffer, audio output process will empty other buffer. Using this function for audio module configuration is recommended for the applications which requires little or no processing of the audio input data. Calling this function requires NO buffer copying in the DMA ISR for routing the audio input data to audio output which will be taken care by audio library internally.

Arguments

Return Value CSL_Status

- CSL_SOK - Initializations are successful
- CSL_ESYS_FAIL - Initializations are not successful

3.1.2.3 Audio() [2/2]

```
int AudioClass::Audio (
    int process )
```

Audio

Description

Function initializes Audio Module.

This function is for enabling the audio loopback for routing data received from audio input to the audio output with processing involved on the audio input data. Calling this function with 'process' parameter set to 'TRUE' enables audio library to use independent data buffers for audio input and output channels. Using this function for audio module configuration is recommended for the applications which requires complex processing of the audio input data. Calling this function requires copying of the processed audio data to audio output buffers in the DMA ISR. Application should read the audio input data on the DMA read right channel interrupt and copy the processed audio data to audio output data buffers on DMA write right channel interrupt.

Note: Calling this function with 'process' parameter set to 'FALSE' is same as calling the function [Audio\(void\)](#).

Arguments

Return Value CSL_Status

- CSL_SOK - Initializations are successful
- CSL_ESYS_FAIL - Initializations are not successful

3.1.2.4 audioMute()

```
int AudioClass::audioMute (  
    void )
```

audioMute

Description

Function to mute audio.

Arguments

Return Value

 CSL_Status

- CSL_SOK - Muting successful
- CSL_ESYS_FAIL - Muting is not successful

3.1.2.5 audioUnmute()

```
int AudioClass::audioUnmute (  
    void )
```

audioUnmute

Description

Function to unmute audio.

Arguments

Return Value

 CSL_Status

- CSL_SOK - Muting successful
- CSL_ESYS_FAIL - Muting is not successful

3.1.2.6 close()

```
int AudioClass::close (  
    void )
```

close

Description

Function to close Audio module.

Arguments

Return Value

 CSL_Status

- CSL_SOK - close is successful
- CSL_ESYS_FAIL - close is not successful

3.1.2.7 detachIntr()

```
void AudioClass::detachIntr (
    void )
```

detachInterrupt
Description

Function to detach ISR function pointer from the DMA interrupt.

Arguments**Return Value**

None

3.1.2.8 HPL_RConF_Routing()

```
int AudioClass::HPL_RConF_Routing (
    int left )
```

HPL_RConF_Routing
Description

Function to select the HPL Routing of the Audio Codec.

Arguments

```
* left - value of the left channel configuration that is to be set for the
*       HPL routing register in order to select the desired routing to
*       HPL
```

Return Value CSL_Status

- CSL_SOK - Selecting the desired HPL routing successful
- CSL_ESYS_FAIL - Selecting the desired HPL routing is not successful

3.1.2.9 HPR_RConF_Routing()

```
int AudioClass::HPR_RConF_Routing (
    int left,
    int right )
```

HPR_RConF_Routing
Description

Function to select the HPR Routing of the Audio Codec.

Arguments

```
* left - value of the left channel configuration that is to be set for the
*       HPR routing register in order to select the desired routing to
*       HPR
* right - value of the right channel configuration that is to be set for
*        the HPR routing register in order to select the desired routing
*        to HPR
```

Return Value CSL_Status

- CSL_SOK - Selecting the desired HPR routing successful
- CSL_ESYS_FAIL - Selecting the desired HPR routing is not successful

3.1.2.10 isrDma()

```
void AudioClass::isrDma (
    void )
```

isrDma

Description

DMA ISR for DMA interrupts.

Arguments

Return Value

None

3.1.2.11 LOL_RConF_Routing()

```
int AudioClass::LOL_RConF_Routing (
    int left,
    int right )
```

LOL_RConF_Routing

Description

Function to select the LOL Routing of the Audio Codec.

Arguments

```
*     left - value of the left channel configuration that is to be set for the
*         LOL routing register in order to select the desired routing to
*         LOL
*     right - value of the right channel configuration that is to be set for
*         the LOL routing register in order to select the desired routing
*         to LOL
```

Return Value CSL_Status

- CSL_SOK - Selecting the desired LOL routing successful
- CSL_ESYS_FAIL - Selecting the desired LOL routing is not successful

3.1.2.12 LOR_RConF_Routing()

```
int AudioClass::LOR_RConF_Routing (
    int right )
```

LOR_RConF_Routing
Description

Function to select the LOR Routing of the Audio Codec.

Arguments

```
*      right - value of the right channel configuration that is to be set for
*            the LOR routing register in order to select the desired routing
*            to LOR
```

Return Value CSL_Status

- CSL_SOK - Selecting the desired LOR routing successful
- CSL_ESYS_FAIL - Selecting the desired LOR routing is not successful

3.1.2.13 read()

```
int AudioClass::read (
    void )
```

read
Description

Function to perform DMA read of left and right channels.

Arguments**Return Value**

None

3.1.2.14 setInputGain()

```
int AudioClass::setInputGain (
    int lgain,
    int rgain )
```

setInputGain
Description

Function to set ADC gain for left and right channels independently.

The ADC gain value will be in the range of 0.0dB to 47.5dB, a increase in 1% ADC gain (left/right) will result in increasing the ADC gain (left/right) by 0.5dB or 1dB

Arguments

```
*      lgain - ADC gain value for left channel in terms of percentage(0 - 100)
*      rgain - ADC gain value for right channel in terms of percentage(0 - 100)
```

Return Value CSL_Status

- CSL_SOK - Gain setting successful
- CSL_ESYS_FAIL - Gain setting is not successful

3.1.2.15 setOutputVolume() [1/2]

```
int AudioClass::setOutputVolume (
    int volume )
```

setOutputVolume

Description

Function to set DAC volume with left channel controlling both left and right.

The DAC volume value will be in the range of : -63.5dB, -63.0dB,, -0.5dB, 0.0dB, +0.5dB,, +23.5dB, +24dB. A increase in 1% Volume (left/right) will result in increasing the DAC Volume (left/right) by 1.5dB or 2dB

Arguments

* volume - DAC volume value in terms of percentage(0 - 100)

Return Value CSL_Status

- CSL_SOK - Volume setting successful
- CSL_ESYS_FAIL - Volume setting is not successful

3.1.2.16 setOutputVolume() [2/2]

```
int AudioClass::setOutputVolume (
    int lvolume,
    int rvolume )
```

setOutputVolume

Description

Function to set DAC volume with left and right independent control.

Arguments

* lvolume - DAC volume value for left channel
 * rvolume - DAC volume value for right channel

Return Value CSL_Status

- CSL_SOK - Volume setting successful
- CSL_ESYS_FAIL - Volume setting is not successful

3.1.2.17 setSamplingRate()

```
int AudioClass::setSamplingRate (
    long samplingRate )
```

setSamplingRate

Description

Function to set Sampling Rate for Audio IN/OUT.

Arguments

* *samplingRate* - Sampling Rate that is to be set

Return Value

 CSL_Status

- CSL_SOK - Setting Sampling rate successful
- CSL_ESYS_FAIL - Setting Sampling rate is not successful

3.1.2.18 write()

```
int AudioClass::write (
    void )
```

write

Description

Function to perform DMA writes of left and right channels.

Arguments

Return Value

None

3.1.3 Member Data Documentation

3.1.3.1 activeInBuf

```
unsigned short AudioClass::activeInBuf
```

Active input buffer

3.1.3.2 activeOutBuf

```
unsigned short AudioClass::activeOutBuf
```

Active output buffer

3.1.3.3 audioInLeft

```
Uint16* AudioClass::audioInLeft [2]
```

Audio input - left channel

3.1.3.4 audioInRight

```
Uint16* AudioClass::audioInRight [2]
```

Audio input - right channel

3.1.3.5 audioOutLeft

```
Uint16* AudioClass::audioOutLeft [2]
```

Audio output - left channel

3.1.3.6 audioOutRight

```
Uint16* AudioClass::audioOutRight [2]
```

Audio output - right channel

3.1.3.7 sampleLeft

```
int AudioClass::sampleLeft
```

Left sample

3.1.3.8 sampleRight

```
int AudioClass::sampleRight
```

Right sample

The documentation for this class was generated from the following files:

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/Audio.h](#)
- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/Audio.cpp](#)

3.2 BR_struct Struct Reference

```
#include <chk_mmc.h>
```

Public Attributes

- unsigned int [short_jump_instr_byte_1](#):8
- unsigned int [short_jump_instr_byte_2](#):8
- unsigned int [short_jump_instr_byte_3](#):8
- unsigned int [oem_name_byte_1](#):8
- unsigned int [oem_name_byte_2](#):8
- unsigned int [oem_name_byte_3](#):8
- unsigned int [oem_name_byte_4](#):8
- unsigned int [oem_name_byte_5](#):8
- unsigned int [oem_name_byte_6](#):8
- unsigned int [oem_name_byte_7](#):8
- unsigned int [oem_name_byte_8](#):8
- unsigned int [LB_bytes_per_sector](#):8
- unsigned int [UB_bytes_per_sector](#):8
- unsigned int [sectors_per_cluster](#):8
- unsigned int [LB_reserved_sectors](#):8
- unsigned int [UB_reserved_sectors](#):8
- unsigned int [no_of_fats](#):8
- unsigned int [LB_no_of_root_dir_entries](#):8
- unsigned int [UB_no_of_root_dir_entries](#):8
- unsigned int [LB_no_of_sectors_on_partition](#):8
- unsigned int [UB_no_of_sectors_on_partition](#):8
- unsigned int [media_descriptor](#):8
- unsigned int [LB_sectors_per_fat](#):8
- unsigned int [UB_sectors_per_fat](#):8
- unsigned int [LB_sectors_per_track](#):8
- unsigned int [UB_sectors_per_track](#):8
- unsigned int [LB_no_of_heads](#):8
- unsigned int [UB_no_of_heads](#):8
- unsigned int [byte1_no_hidden_sectors](#):8
- unsigned int [byte2_no_hidden_sectors](#):8
- unsigned int [byte3_no_hidden_sectors](#):8
- unsigned int [byte4_no_hidden_sectors](#):8
- unsigned int [byte1_extended_no_of_sectors_on_partition](#):8
- unsigned int [byte2_extended_no_of_sectors_on_partition](#):8
- unsigned int [byte3_extended_no_of_sectors_on_partition](#):8
- unsigned int [byte4_extended_no_of_sectors_on_partition](#):8
- unsigned int [drive_number](#):8
- unsigned int [reserved](#):8
- unsigned int [extended_boot_signature](#):8
- unsigned int [vol_serial_number_byte_1](#):8
- unsigned int [vol_serial_number_byte_2](#):8
- unsigned int [vol_serial_number_byte_3](#):8
- unsigned int [vol_serial_number_byte_4](#):8
- unsigned int [byte1_vol_label](#):8
- unsigned int [byte2_vol_label](#):8
- unsigned int [byte3_vol_label](#):8
- unsigned int [byte4_vol_label](#):8

- unsigned int [byte5_vol_label](#):8
- unsigned int [byte6_vol_label](#):8
- unsigned int [byte7_vol_label](#):8
- unsigned int [byte8_vol_label](#):8
- unsigned int [byte9_vol_label](#):8
- unsigned int [byte10_vol_label](#):8
- unsigned int [byte11_vol_label](#):8
- unsigned int [byte1_fs_type](#):8
- unsigned int [byte2_fs_type](#):8
- unsigned int [byte3_fs_type](#):8
- unsigned int [byte4_fs_type](#):8
- unsigned int [byte5_fs_type](#):8
- unsigned int [byte6_fs_type](#):8
- unsigned int [byte7_fs_type](#):8
- unsigned int [byte8_fs_type](#):8
- unsigned int [opt_partition_boot_code](#) [224]
- unsigned int [signature](#)

3.2.1 Detailed Description

Structure to hold Boot Record (BR)

3.2.2 Member Data Documentation

3.2.2.1 [byte10_vol_label](#)

```
unsigned int BR_struct::byte10_vol_label
```

Volume label byte 10

3.2.2.2 [byte11_vol_label](#)

```
unsigned int BR_struct::byte11_vol_label
```

Volume label byte 11

3.2.2.3 [byte1_extended_no_of_sectors_on_partition](#)

```
unsigned int BR_struct::byte1_extended_no_of_sectors_on_partition
```

Extended no of sectors byte 1

3.2.2.4 [byte1_fs_type](#)

```
unsigned int BR_struct::byte1_fs_type
```

FS type byte 1

3.2.2.5 byte1_no_hidden_sectors

```
unsigned int BR_struct::byte1_no_hidden_sectors
```

Number of hidden sectors byte 1

3.2.2.6 byte1_vol_label

```
unsigned int BR_struct::byte1_vol_label
```

Volume label byte 1

3.2.2.7 byte2_extended_no_of_sectors_on_partition

```
unsigned int BR_struct::byte2_extended_no_of_sectors_on_partition
```

Extended no of sectors byte 2

3.2.2.8 byte2_fs_type

```
unsigned int BR_struct::byte2_fs_type
```

FS type byte 2

3.2.2.9 byte2_no_hidden_sectors

```
unsigned int BR_struct::byte2_no_hidden_sectors
```

Number of hidden sectors byte 2

3.2.2.10 byte2_vol_label

```
unsigned int BR_struct::byte2_vol_label
```

Volume label byte 2

3.2.2.11 byte3_extended_no_of_sectors_on_partition

```
unsigned int BR_struct::byte3_extended_no_of_sectors_on_partition
```

Extended no of sectors byte 3

3.2.2.12 byte3_fs_type

```
unsigned int BR_struct::byte3_fs_type
```

FS type byte 3

3.2.2.13 byte3_no_hidden_sectors

```
unsigned int BR_struct::byte3_no_hidden_sectors
```

Number of hidden sectors byte 3

3.2.2.14 byte3_vol_label

```
unsigned int BR_struct::byte3_vol_label
```

Volume label byte 3

3.2.2.15 byte4_extended_no_of_sectors_on_partition

```
unsigned int BR_struct::byte4_extended_no_of_sectors_on_partition
```

Extended no of sectors byte 4

3.2.2.16 byte4_fs_type

```
unsigned int BR_struct::byte4_fs_type
```

FS type byte 4

3.2.2.17 byte4_no_hidden_sectors

```
unsigned int BR_struct::byte4_no_hidden_sectors
```

Number of hidden sectors byte 4

3.2.2.18 byte4_vol_label

```
unsigned int BR_struct::byte4_vol_label
```

Volume label byte 4

3.2.2.19 byte5_fs_type

```
unsigned int BR_struct::byte5_fs_type
```

FS type byte 5

3.2.2.20 byte5_vol_label

```
unsigned int BR_struct::byte5_vol_label
```

Volume label byte 5

3.2.2.21 byte6_fs_type

```
unsigned int BR_struct::byte6_fs_type
```

FS type byte 6

3.2.2.22 byte6_vol_label

```
unsigned int BR_struct::byte6_vol_label
```

Volume label byte 6

3.2.2.23 byte7_fs_type

```
unsigned int BR_struct::byte7_fs_type
```

FS type byte 7

3.2.2.24 byte7_vol_label

```
unsigned int BR_struct::byte7_vol_label
```

Volume label byte 7

3.2.2.25 byte8_fs_type

```
unsigned int BR_struct::byte8_fs_type
```

FS type byte 8

3.2.2.26 byte8_vol_label

```
unsigned int BR_struct::byte8_vol_label
```

Volume label byte 8

3.2.2.27 byte9_vol_label

```
unsigned int BR_struct::byte9_vol_label
```

Volume label byte 9

3.2.2.28 drive_number

```
unsigned int BR_struct::drive_number
```

Drive number

3.2.2.29 extended_boot_signature

```
unsigned int BR_struct::extended_boot_signature
```

Extended boot signature

3.2.2.30 LB_bytes_per_sector

```
unsigned int BR_struct::LB_bytes_per_sector
```

Bytes per sector lower byte

3.2.2.31 LB_no_of_heads

```
unsigned int BR_struct::LB_no_of_heads
```

Number of heads lower byte

3.2.2.32 LB_no_of_root_dir_entries

```
unsigned int BR_struct::LB_no_of_root_dir_entries
```

Number of root directories lower byte

3.2.2.33 LB_no_of_sectors_on_partition

```
unsigned int BR_struct::LB_no_of_sectors_on_partition
```

Sectors in partition lower byte

3.2.2.34 LB_reserved_sectors

```
unsigned int BR_struct::LB_reserved_sectors
```

Reserved sectors lower byte

3.2.2.35 LB_sectors_per_fat

```
unsigned int BR_struct::LB_sectors_per_fat
```

Sectors per FAT lower byte

3.2.2.36 LB_sectors_per_track

```
unsigned int BR_struct::LB_sectors_per_track
```

Sectors per track lower byte

3.2.2.37 media_descriptor

```
unsigned int BR_struct::media_descriptor
```

Media descriptor

3.2.2.38 no_of_fats

```
unsigned int BR_struct::no_of_fats
```

Number of fats

3.2.2.39 oem_name_byte_1

```
unsigned int BR_struct::oem_name_byte_1
```

OEM name byte 1

3.2.2.40 oem_name_byte_2

```
unsigned int BR_struct::oem_name_byte_2
```

OEM name byte 2

3.2.2.41 oem_name_byte_3

```
unsigned int BR_struct::oem_name_byte_3
```

OEM name byte 3

3.2.2.42 oem_name_byte_4

```
unsigned int BR_struct::oem_name_byte_4
```

OEM name byte 4

3.2.2.43 oem_name_byte_5

```
unsigned int BR_struct::oem_name_byte_5
```

OEM name byte 5

3.2.2.44 oem_name_byte_6

```
unsigned int BR_struct::oem_name_byte_6
```

OEM name byte 6

3.2.2.45 oem_name_byte_7

```
unsigned int BR_struct::oem_name_byte_7
```

OEM name byte 7

3.2.2.46 oem_name_byte_8

```
unsigned int BR_struct::oem_name_byte_8
```

OEM name byte 8

3.2.2.47 opt_partition_boot_code

```
unsigned int BR_struct::opt_partition_boot_code[224]
```

Partition boot code

3.2.2.48 reserved

```
unsigned int BR_struct::reserved
```

Reserved sectors

3.2.2.49 sectors_per_cluster

```
unsigned int BR_struct::sectors_per_cluster
```

Sectors per cluster

3.2.2.50 short_jump_instr_byte_1

```
unsigned int BR_struct::short_jump_instr_byte_1
```

Short jump instruction byte 1

3.2.2.51 short_jump_instr_byte_2

```
unsigned int BR_struct::short_jump_instr_byte_2
```

Short jump instruction byte 2

3.2.2.52 short_jump_instr_byte_3

```
unsigned int BR_struct::short_jump_instr_byte_3
```

Short jump instruction byte 3

3.2.2.53 signature

```
unsigned int BR_struct::signature
```

Signature

3.2.2.54 UB_bytes_per_sector

```
unsigned int BR_struct::UB_bytes_per_sector
```

Bytes per sector upper byte

3.2.2.55 UB_no_of_heads

```
unsigned int BR_struct::UB_no_of_heads
```

Number of heads upper byte

3.2.2.56 UB_no_of_root_dir_entries

```
unsigned int BR_struct::UB_no_of_root_dir_entries
```

Number of root directories upper byte

3.2.2.57 UB_no_of_sectors_on_partition

```
unsigned int BR_struct::UB_no_of_sectors_on_partition
```

Sectors in partition upper byte

3.2.2.58 UB_reserved_sectors

```
unsigned int BR_struct::UB_reserved_sectors
```

Reserved sectors upper byte

3.2.2.59 UB_sectors_per_fat

```
unsigned int BR_struct::UB_sectors_per_fat
```

Sectors per FAT upper byte

3.2.2.60 UB_sectors_per_track

```
unsigned int BR_struct::UB_sectors_per_track
```

Sectors per track upper byte

3.2.2.61 vol_serial_number_byte_1

```
unsigned int BR_struct::vol_serial_number_byte_1
```

Volume serial number byte 1

3.2.2.62 vol_serial_number_byte_2

```
unsigned int BR_struct::vol_serial_number_byte_2
```

Volume serial number byte 2

3.2.2.63 vol_serial_number_byte_3

```
unsigned int BR_struct::vol_serial_number_byte_3
```

Volume serial number byte 3

3.2.2.64 vol_serial_number_byte_4

```
unsigned int BR_struct::vol_serial_number_byte_4
```

Volume serial number byte 4

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

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/chk_mmc.h](#)

3.3 FFTClass Class Reference

FFT Class.

```
#include <FFT.h>
```

Public Member Functions

- void [FFT_init](#) (void)
- void [FFT_filter](#) (short *input, short *output, short *overlap, int length)

3.3.1 Detailed Description

FFT Class.

Contains prototypes for functions in FFT library

3.3.2 Member Function Documentation

3.3.2.1 FFT_filter()

```
void FFTClass::FFT_filter (
    short * input,
    short * output,
    short * overlap,
    int length )
```

FFT_filter()

Description

API to perform FFT-filter(lp)-IFFT on the samples passed as input

Arguments

```
input   - Buffer containing the Input samples
output  - Buffer to hold the FFT-filtered output samples
overlap - Buffer to hold the overlap data, which will be used in next
          FFT-filtering
length  - Number of input data samples
```

Return Value

None

3.3.2.2 FFT_init()

```
void FFTClass::FFT_init (
    void )
```

FFT_init()

Description

API to initialize the buffers used by the FFT algorithm

Arguments

Return Value

None

The documentation for this class was generated from the following files:

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/FFT.h](#)
- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/FFT.cpp](#)

3.4 File Class Reference

File Class.

```
#include <SD.h>
```

Public Member Functions

- `operator bool ()` const
- `File ()`
File fileClass.
- `File (const File &fileObj)`
File fileClass.
- unsigned long `available ()`
available()
- void `close ()`
close()
- void `flush ()`
flush()
- char `peek ()`
peek()
- unsigned long `position ()`
position()
- int `print (char character)`
print(character)
- int `print (char *printString)`
print(printString)
- int `print (int integer)`
print(integer)
- int `print (long integer)`
print(integer)
- int `print (float value)`
print(value)
- int `print (double value)`
print(value)
- int `print (long integer, NUMBER_FORMAT_BASE base)`
print(integer, base)
- int `println (void)`
println()
- int `println (char character)`
println(character)
- int `println (char *printString)`
println(printString)
- int `println (int integer)`
println(integer)
- int `println (long integer)`
println(integer)
- int `println (float value)`
print(value)
- int `println (double value)`

- print(value)*
- int [println](#) (long integer, [NUMBER_FORMAT_BASE](#) base)
 - println(integer, base)*
- Bool [seek](#) (unsigned long posValue)
 - seek(posValue)*
- unsigned long [size](#) ()
 - size()*
- char [read](#) ()
 - read()*
- int [read](#) (char *buffer, int length)
 - read(buffer, length)*
- int [read](#) (int *buffer, int length)
 - read(buffer, length)*
- int [write](#) (char character)
 - write(character)*
- int [write](#) (long integer)
 - write(integer)*
- int [write](#) (char *printString)
 - write(printString)*
- int [write](#) (char *buffer, int length)
 - write(buffer, length)*
- int [write](#) (int *buffer, int length)
 - write(buffer, length)*
- Bool [isDirectory](#) ()
 - isDirectory()*
- File [openNextFile](#) ()
 - openNextFile()*
- void [rewindDirectory](#) ()
 - rewindDirectory()*
- void [getName](#) (char *fileName)
 - getName(fileName)*

Friends

- class [SD_Class](#)

3.4.1 Detailed Description

[File](#) Class.

The [File](#) class allows for reading from and writing to individual files on the SD card

3.4.2 Constructor & Destructor Documentation

3.4.2.1 [File\(\)](#) [1/2]

`File::File ()`

[File](#) fileClass.

[File](#) Class Constructor.

Returns

NONE

3.4.2.2 File() [2/2]

```
File::File (
    const File & fileObj )
```

[File](#) fileClass.

[File](#) Class Copy Constructor.

Parameters

<i>fileObj</i>	[IN] An existing File Object
----------------	--

Returns

NONE

3.4.3 Member Function Documentation

3.4.3.1 available()

```
unsigned long File::available ( )
```

[available\(\)](#)

API to Check if there are any bytes available for reading from the file

Returns

Number of bytes available, that can be read

3.4.3.2 close()

```
void File::close (
    void )
```

[close\(\)](#)

API to Close an opened file

Returns

NONE

3.4.3.3 flush()

```
void File::flush ( )
```

flush()

API to ensure that any bytes written to the file are physically saved to the SD card

Returns

NONE

3.4.3.4 getName()

```
void File::getName (
    char * fileName )
```

getName(fileName)

API to get the name of the [File](#)

Parameters

<i>fileName</i>	[IN] Buffer to hold the name of the file
-----------------	--

Returns

NONE

3.4.3.5 isDirectory()

```
Bool File::isDirectory ( )
```

isDirectory()

API to report if the current file is a directory or not

Returns

TRUE - If the current file is a Directory FALSE - If the current file is not a Directory

3.4.3.6 openNextFile()

```
File File::openNextFile ( )
```

openNextFile()

API to report the next file or folder in a directory

Returns

Pointer to the ['File'](#) class of the next file

3.4.3.7 operator bool()

```
File::operator bool ( ) const [inline]
```

To check file open status

3.4.3.8 peek()

```
char File::peek ( )
```

[peek\(\)](#)

API to Read a byte from the file without advancing to the next one

Returns

The character that is read from the file

3.4.3.9 position()

```
unsigned long File::position ( )
```

[position\(\)](#)

API to Get the current position to which the next byte will be read from or written to

Returns

The location to which the next byte will be read from or written to

3.4.3.10 print() [1/7]

```
int File::print (
    char character )
```

[print\(character\)](#)

API to Print an ASCII character to the file

Parameters

<i>character</i>	[IN] The ASCII character to be printed to the file
------------------	--

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.11 print() [2/7]

```
int File::print (
    char * printString )
```

print(*printString*)

API to Print a String(character array) to the file

Parameters

<i>printString</i>	[IN] The String to be printed to the file
--------------------	---

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.12 print() [3/7]

```
int File::print (
    int integer )
```

print(*integer*)

API to Print a 16 Bit integer(int) to the file

Parameters

<i>integer</i>	[IN] The 16 Bit integer(int) to be printed to the file
----------------	--

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.13 print() [4/7]

```
int File::print (
    long integer )
```

print(*integer*)

API to Print a 32 Bit integer(long) to the file

Parameters

<i>integer</i>	[IN] The 32 Bit integer(long) to be printed to the file
----------------	---

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.14 print() [5/7]

```
int File::print (
    float value )
```

print(value)

API to Print a Floating Point Value to the file as human-readable ASCII text

Parameters

<i>value</i>	[IN] Floating Point value that is to be written to the file
--------------	---

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.15 print() [6/7]

```
int File::print (
    double value )
```

print(value)

API to Print a Double Precision Floating Point Value to the file as human-readable ASCII text

Parameters

<i>value</i>	[IN] Double Precision Floating Point value that is to be written to the file
--------------	--

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.16 print() [7/7]

```
int File::print (
    long integer,
    NUMBER_FORMAT_BASE base )
```

print(integer, base)

API to Print an integer in a specified base format to the file

Parameters

<i>integer</i>	[IN] The integer to be printed to the file
<i>base</i>	[IN] The base in which to print numbers

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.17 println() [1/8]

```
int File::println (
    void )
```

println()

API to Print a carriage return and newline to the file

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.18 println() [2/8]

```
int File::println (
    char character )
```

println(character)

API to Print an ASCII character, followed by a carriage return and newline to the file

Parameters

<i>character</i>	[IN] The ASCII character to be printed to the file
------------------	--

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.19 println() [3/8]

```
int File::println (
    char * printString )
```

println(printString)

API to Print a String(character array), followed by a carriage return and newline to the file

Parameters

<i>printString</i>	[IN] The String to be printed to the file
--------------------	---

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.20 println() [4/8]

```
int File::println (
    int integer )
```

println(integer)

API to Print an integer, followed by a carriage return and newline to the file

Parameters

<i>integer</i>	[IN] The integer to be printed to the file
----------------	--

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.21 println() [5/8]

```
int File::println (
    long integer )
```

println(integer)

API to Print an integer, followed by a carriage return and newline to the file

Parameters

<i>integer</i>	[IN] The integer to be printed to the file
----------------	--

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.22 println() [6/8]

```
int File::println (
    float value )
```

print(value)

API to Print a Floating Point Value to the file as human-readable ASCII text, followed by a carriage return and newline to the file

Parameters

<i>value</i>	[IN] Floating Point value that is to be written to the file
--------------	---

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.23 println() [7/8]

```
int File::println (
    double value )
```

print(value)

API to Print a Double Precision Floating Point Value to the file as human-readable ASCII text, followed by a carriage return and newline to the file

Parameters

<i>value</i>	[IN] Double Precision Floating Point value that is to be written to the file
--------------	--

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.24 println() [8/8]

```
int File::println (
    long integer,
    NUMBER_FORMAT_BASE base )
```

println(integer, base)

API to Print an integer in a specified base format, followed by a carriage return and newline to the file

Parameters

<i>integer</i>	[IN] The integer to be printed to the file
<i>base</i>	[IN] The base in which to print numbers

Returns

The number of bytes printed to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.25 read() [1/3]

```
char File::read (
    void )
```

[read\(\)](#)

API to Read a byte from the file

Returns

The next byte (or character) -1 if no data is available

3.4.3.26 read() [2/3]

```
int File::read (
    char * buffer,
    int length )
```

read(buffer, length)

API to Read an array of bytes from the file

Parameters

<i>buffer</i>	[IN] The Buffer to hold the data read from the file
<i>length</i>	[IN] The length of the data to be read, in no of bytes

Returns

The number of bytes read from the file Else zero, if the file is not opened

3.4.3.27 read() [3/3]

```
int File::read (
    int * buffer,
    int length )
```

read(buffer, length)

API to Read an array of Words (16 bits) from the file

Parameters

<i>buffer</i>	[IN] The Buffer to hold the data read from the file
<i>length</i>	[IN] The length of the data to be read, in no of Words (16 bits)

Returns

The number of Words read from the file Else zero, if the file is not opened

3.4.3.28 rewindDirectory()

```
void File::rewindDirectory ( )
```

[rewindDirectory\(\)](#)

API to go back to the first file in the directory

Returns

NONE

3.4.3.29 seek()

```
Bool File::seek (
    unsigned long posValue )
```

[seek\(posValue\)](#)

API to Seek to a new position in the file, which must be between 0 and the size of the file

Parameters

<i>posValue</i>	[IN] The position to which to seek
-----------------	------------------------------------

Returns

TRUE - Successful in seeking to the requested position FALSE - Unsuccessful in seeking to the requested position

3.4.3.30 size()

```
unsigned long File::size ( )
```

[size\(\)](#)

API to Get the size of the file

Returns

The size of the file in bytes

3.4.3.31 write() [1/5]

```
int File::write (
    char character )
```

[write\(character\)](#)

API to Write an ASCII character to the file

Parameters

<i>character</i>	[IN] The ASCII character to be written to the file
------------------	--

Returns

The number of bytes written to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.32 write() [2/5]

```
int File::write (
    long integer )
```

write(integer)

API to Write an integer to the file

Parameters

<i>integer</i>	[IN] The integer to be written to the file
----------------	--

Returns

The number of bytes written to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.33 write() [3/5]

```
int File::write (
    char * printString )
```

write(printString)

API to Write a String(character array) to the file

Parameters

<i>printString</i>	[IN] The String to be written to the file
--------------------	---

Returns

The number of bytes written to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.34 write() [4/5]

```
int File::write (
    char * buffer,
    int length )
```

write(buffer, length)

API to Write a String(character array) of specified length to the file

Parameters

<i>buffer</i>	[IN] The String to be written to the file
<i>length</i>	[IN] The length of the String to be written

Returns

The number of bytes written to the file Else zero, if the file is not opened or for invalid file operation

3.4.3.35 write() [5/5]

```
int File::write (
    int * buffer,
    int length )
```

write(buffer, length)

API to Write an Array of words, of specified length to the file

Parameters

<i>buffer</i>	[IN] The Buffer (containing array of words) to be written to the file
<i>length</i>	[IN] No of Words (16 bits) to be written

Returns

The number of Words written to the file Else zero, if the file is not opened or for invalid file operation

3.4.4 Friends And Related Function Documentation

3.4.4.1 SD_Class

```
friend class SD_Class [friend]
```

Making 'SD_Class' class friend of 'File' class, so that member functions of 'SD_Class' can access private member fields of 'File'

The documentation for this class was generated from the following files:

- /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.h
- /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.cpp

3.5 fileNodesList Class Reference

Node of the Linked List to hold the details of all the Opened files.

Public Member Functions

- [fileNodesList](#) ()

Public Attributes

- unsigned long [startCluster](#)
- struct [fileNodesList](#) * [nextFileNode](#)

3.5.1 Detailed Description

Node of the Linked List to hold the details of all the Opened files.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 [fileNodesList](#)()

```
fileNodesList::fileNodesList ( ) [inline]
```

3.5.3 Member Data Documentation

3.5.3.1 [nextFileNode](#)

```
struct fileNodesList* fileNodesList::nextFileNode
```

Pointer to next file

3.5.3.2 [startCluster](#)

```
unsigned long fileNodesList::startCluster
```

Starting cluster of file

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

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.cpp](#)

3.6 FONT_CHAR_INFO Struct Reference

This structure describes a single character's display information.

```
#include <bitmapDb.h>
```

Public Attributes

- const Uint8 [widthBits](#)
- const Uint8 [heightBits](#)
- const Uint16 [offset](#)

3.6.1 Detailed Description

This structure describes a single character's display information.

3.6.2 Member Data Documentation

3.6.2.1 heightBits

```
const Uint8 FONT_CHAR_INFO::heightBits
```

height, in bits (or pixels), of the character

3.6.2.2 offset

```
const Uint16 FONT_CHAR_INFO::offset
```

offset of the character's bitmap, in bytes, into the the [FONT_INFO](#)'s data array

3.6.2.3 widthBits

```
const Uint8 FONT_CHAR_INFO::widthBits
```

width, in bits (or pixels), of the character

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

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/bitmapDb.h](#)

3.7 FONT_INFO Struct Reference

This structure describes a single font.

```
#include <bitmapDb.h>
```

Public Attributes

- const Uint8 [heightPages](#)
- const Uint8 [startChar](#)
- const Uint8 [endChar](#)
- const [FONT_CHAR_INFO](#) * [charInfo](#)
- const Uint8 * [data](#)

3.7.1 Detailed Description

This structure describes a single font.

3.7.2 Member Data Documentation

3.7.2.1 charInfo

```
const FONT_CHAR_INFO* FONT_INFO::charInfo
```

pointer to array of char information

3.7.2.2 data

```
const Uint8* FONT_INFO::data
```

pointer to generated array of character visual representation

3.7.2.3 endChar

```
const Uint8 FONT_INFO::endChar
```

the last character in the font

3.7.2.4 heightPages

```
const Uint8 FONT_INFO::heightPages
```

height, in pages (8 pixels), of the font's characters

3.7.2.5 startChar

```
const Uint8 FONT_INFO::startChar
```

the first character in the font (e.g. in charInfo and data)

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

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/bitmapDb.h](#)

3.8 GPT_Config Struct Reference

Configuration structure.

```
#include <Timers.h>
```

Public Attributes

- Bool [autoLoad](#)
- unsigned short [preScaleDiv](#)
- Bool [ctrlTim](#)
- unsigned short [prdLow](#)
- unsigned short [prdHigh](#)

3.8.1 Detailed Description

Configuration structure.

Contains the Information of a GPT configuration.

3.8.2 Member Data Documentation

3.8.2.1 autoLoad

```
Bool GPT_Config::autoLoad
```

Auto reload

3.8.2.2 ctrlTim

```
Bool GPT_Config::ctrlTim
```

control timer

3.8.2.3 prdHigh

```
unsigned short GPT_Config::prdHigh
```

period high

3.8.2.4 prdLow

```
unsigned short GPT_Config::prdLow
```

period low

3.8.2.5 preScaleDiv

```
unsigned short GPT_Config::preScaleDiv
```

Prescale division

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

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/Timers.h](#)

3.9 MBR_struct Struct Reference

```
#include <chk_mmc.h>
```

Public Attributes

- unsigned int [mbr_code](#) [216]
- unsigned int [mmc_id_entry](#) [7]
- [PARTITION_TABLE](#) [partition_one](#)
- [PARTITION_TABLE](#) [partition_two](#)
- [PARTITION_TABLE](#) [partition_three](#)
- [PARTITION_TABLE](#) [partition_four](#)
- unsigned int [signature](#)

3.9.1 Detailed Description

Structure to hold Master Boot Record (MBR)

3.9.2 Member Data Documentation

3.9.2.1 mbr_code

```
unsigned int MBR_struct::mbr_code[216]
```

MBR code 432 bytes

3.9.2.2 mmc_id_entry

```
unsigned int MBR_struct::mmc_id_entry[7]
```

MMC ID entry 14 bytes

3.9.2.3 partition_four

```
PARTITION\_TABLE MBR_struct::partition_four
```

Partition 4 entry 16 bytes

3.9.2.4 partition_one

```
PARTITION\_TABLE MBR_struct::partition_one
```

Partition 1 entry 16 bytes

3.9.2.5 partition_three

`PARTITION_TABLE` `MBR_struct::partition_three`

Partition 3 entry 16 bytes

3.9.2.6 partition_two

`PARTITION_TABLE` `MBR_struct::partition_two`

Partition 2 entry 16 bytes

3.9.2.7 signature

`unsigned int` `MBR_struct::signature`

FS signature 2 bytes

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

- `/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/chk_mmc.h`

3.10 OLED Class Reference

`OLED` Class.

```
#include <OLED.h>
```

Public Member Functions

- void `oledInit` (void)
- void `init` ()
- void `begin` ()
- void `clear` ()
- void `clear` (int)
- int `printchar` (unsigned char)
- int `print` (char string[])
- int `print` (long value)
- int `print` (char character)
- int `print` (long value, int base)
- void `write` (unsigned char)
- void `noDisplay` ()
- void `display` ()
- void `scrollDisplayLeft` ()
- void `scrollDisplayLeft` (int line)
- void `scrollDisplayRight` ()
- void `scrollDisplayRight` (int line)
- void `flip` ()
- void `setOrientation` (int newDir)
- void `autoscroll` ()
- void `noAutoscroll` ()
- void `setline` (int line)
- void `setRolling` (int row, int status)
- void `resetCursor` (int page)

3.10.1 Detailed Description

[OLED](#) Class.

Contains prototypes for functions in [OLED](#) library

3.10.2 Member Function Documentation

3.10.2.1 `autoscroll()`

```
void OLED::autoscroll (
    void )
```

`autoscroll()`

Description

Turns on automatic scrolling of the LCD

Arguments

Return Value

None

3.10.2.2 `begin()`

```
void OLED::begin (
    void )
```

`begin()`

Description

Specifies the dimensions (width and height) of the display.

Arguments

```
*      Uint8 cols      <- Number of columns
*      Uint8 rows      <- Number of rows
```

Return Value

None

3.10.2.3 clear() [1/2]

```
void OLED::clear (
    void )
```

clear()

Description

Clears the LCD screen and positions the cursor in the upper-left corner

Arguments

Return Value

None

3.10.2.4 clear() [2/2]

```
void OLED::clear (
    int page )
```

clear(page)

Description

Clears the LCD screen and positions the cursor in the upper-left corner

Arguments

* int page <- Page number

Return Value

None

3.10.2.5 display()

```
void OLED::display ( )
```

display()

Description

Turns on the LCD display

Arguments

Return Value

None

3.10.2.6 flip()

```
void OLED::flip ( )
```

flip()

Description

Flips the screen vertically

Arguments

Return Value

None

3.10.2.7 init()

```
void OLED::init (
    void )
```

init()

Description

Function initializes I2C Module and also LCD.

Arguments

Return Value

None

3.10.2.8 noAutoscroll()

```
void OLED::noAutoscroll (
    void )
```

autoscroll()

Description

Turns off automatic scrolling of the LCD

Arguments

Return Value

None

3.10.2.9 noDisplay()

```
void OLED::noDisplay ( )
```

noDisplay()

Description

Turns off the LCD display

Arguments

Return Value

None

3.10.2.10 oledInit()

```
void OLED::oledInit (
    void )
```

oledInit()

Description

Function initializes I2C Module and also LCD.

Arguments

Return Value

None

3.10.2.11 print() [1/4]

```
int OLED::print (
    char string[] )
```

print(string[])

Description

Prints a string to the LCD

Arguments

* char *printString <- Pointer to the string to be printed

Return Value Returns the number of bytes written to LCD

- strLen - Length of the string written to LCD

3.10.2.12 print() [2/4]

```
int OLED::print (
    long value )
```

```
=====
```

print(value)
Description

Prints a value to the LCD

Arguments

* long value <- Value to be printed

Return Value Returns the number of bytes written to LCD

- strlen - Length of the string written to LCD

3.10.2.13 print() [3/4]

```
int OLED::print (
    char character )
```

```
=====
```

print(character)
Description

Prints a character to the LCD

Arguments

* char character <- Character to be printed

Return Value Returns the number of bytes written to LCD

- 1 - In case of char only 1byte is written to LCD

3.10.2.14 print() [4/4]

```
int OLED::print (
    long value,
    int base )
```

```
=====
```

print(value, base)
Description

Prints a value to the LCD

Arguments

* long value <- Value to be printed
* int base <- Base of the value printed

Return Value Returns the number of bytes written to LCD

- strlen - Length of the string written to LCD

3.10.2.15 printchar()

```
int OLED::printchar (
    unsigned char a )
```

printchar(character)

Description

Prints a character to the LCD

Arguments

* unsigned char character <- Character to be printed

Return Value

None

3.10.2.16 resetCursor()

```
void OLED::resetCursor (
    int line )
```

resetCursor(line)

Description

Resets the cursor position to the beginning in a given line

Arguments

* int line <- line number

Return Value

None

3.10.2.17 scrollDisplayLeft() [1/2]

```
void OLED::scrollDisplayLeft (
    void )
```

scrollDisplayLeft()

Description

Scrolls the contents of the display (text and cursor) to the left

Arguments

Return Value

None

3.10.2.18 scrollDisplayLeft() [2/2]

```
void OLED::scrollDisplayLeft (
    int line )
```

scrollDisplayLeft()

Description

Scrolls the contents of a particular line, of the display (text and cursor) to the left

Arguments

* int line <- Line number

Return Value

None

3.10.2.19 scrollDisplayRight() [1/2]

```
void OLED::scrollDisplayRight (
    void )
```

scrollDisplayRight()

Description

Scrolls the contents of the display (text and cursor) to the right

Arguments

Return Value

None

3.10.2.20 scrollDisplayRight() [2/2]

```
void OLED::scrollDisplayRight (
    int line )
```

scrollDisplayRight(line)

Description

Scrolls the contents of a particular line, of the display (text and cursor) to the right

Arguments

* int line <- Line number

Return Value

None

3.10.2.21 `setline()`

```
void OLED::setline (
    int line )
```

setline(line)

Description

Sets the start line for the display

Arguments

* `int line` <- Line number

Return Value

None

3.10.2.22 `setOrientation()`

```
void OLED::setOrientation (
    int newDir )
```

setOrientation()

Description

Function to set the orientation of LCD.

Arguments

* `int newDir` <- Direction of orientation

Return Value

None

3.10.2.23 `setRolling()`

```
void OLED::setRolling (
    int row,
    int status )
```

setRolling(int row, int status)

Description

Sets the rolling parameters

Arguments

* `int row` <- Row number
* `int status` <- Status of rolling

Return Value

None

3.10.2.24 write()

```
void OLED::write (
    unsigned char byte )
```

write(character)

Description

Writes a character to the LCD

Arguments

* char character <- Character to be printed

Return Value Returns the number of bytes written to LCD

- 1 - In case of char only 1 byte is written to LCD

The documentation for this class was generated from the following files:

- /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/OLED.h
- /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/OLED.cpp

3.11 PARTITION_TABLE Struct Reference

```
#include <chk_mmc.h>
```

Public Attributes

- unsigned int [boot_descriptor](#):8
- unsigned int [partition_start_head](#):8
- unsigned int [partition_start_sector](#):6
- unsigned int [partition_start_cylinder](#):10
- unsigned int [fs_descriptor](#):8
- unsigned int [partition_end_head](#):8
- unsigned int [partition_end_sector](#):6
- unsigned int [partition_end_cylinder](#):10
- unsigned int [byte1_first_sector_position](#):8
- unsigned int [byte2_first_sector_position](#):8
- unsigned int [byte3_first_sector_position](#):8
- unsigned int [byte4_first_sector_position](#):8
- unsigned int [byte1_no_of_sectors_in_partition](#):8
- unsigned int [byte2_no_of_sectors_in_partition](#):8
- unsigned int [byte3_no_of_sectors_in_partition](#):8
- unsigned int [byte4_no_of_sectors_in_partition](#):8

3.11.1 Detailed Description

Structure to hold Partition table

3.11.2 Member Data Documentation

3.11.2.1 boot_descriptor

```
unsigned int PARTITION_TABLE::boot_descriptor
```

Boot descriptor 1 byte

3.11.2.2 byte1_first_sector_position

```
unsigned int PARTITION_TABLE::byte1_first_sector_position
```

Byte 1 first sector position

3.11.2.3 byte1_no_of_sectors_in_partition

```
unsigned int PARTITION_TABLE::byte1_no_of_sectors_in_partition
```

Byte 1 sectors in partition

3.11.2.4 byte2_first_sector_position

```
unsigned int PARTITION_TABLE::byte2_first_sector_position
```

Byte 2 first sector position

3.11.2.5 byte2_no_of_sectors_in_partition

```
unsigned int PARTITION_TABLE::byte2_no_of_sectors_in_partition
```

Byte 2 sectors in partition

3.11.2.6 byte3_first_sector_position

```
unsigned int PARTITION_TABLE::byte3_first_sector_position
```

Byte 3 first sector position

3.11.2.7 byte3_no_of_sectors_in_partition

unsigned int PARTITION_TABLE::byte3_no_of_sectors_in_partition

Byte 3 sectors in partition

3.11.2.8 byte4_first_sector_position

unsigned int PARTITION_TABLE::byte4_first_sector_position

Byte 4 first sector position

3.11.2.9 byte4_no_of_sectors_in_partition

unsigned int PARTITION_TABLE::byte4_no_of_sectors_in_partition

Byte 4 sectors in partition

3.11.2.10 fs_descriptor

unsigned int PARTITION_TABLE::fs_descriptor

FS descriptor 1 byte

3.11.2.11 partition_end_cylinder

unsigned int PARTITION_TABLE::partition_end_cylinder

Partition end cylinder 10 bits

3.11.2.12 partition_end_head

unsigned int PARTITION_TABLE::partition_end_head

Partition end head 1 byte

3.11.2.13 partition_end_sector

unsigned int PARTITION_TABLE::partition_end_sector

Partition end sector 6 bits

3.11.2.14 partition_start_cylinder

unsigned int PARTITION_TABLE::partition_start_cylinder

Partition start cylinder 10 bits

3.11.2.15 partition_start_head

```
unsigned int PARTITION_TABLE::partition_start_head
```

Partition start head 1 byte

3.11.2.16 partition_start_sector

```
unsigned int PARTITION_TABLE::partition_start_sector
```

Partition start sector 6 bits

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

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/chk_mmc.h](#)

3.12 Pipe_Config Class Reference

Pipe configuration Class.

```
#include <USB.h>
```

Public Attributes

- [USB_xferType](#) xferType
- unsigned short [maxPktSize](#)

3.12.1 Detailed Description

Pipe configuration Class.

3.12.2 Member Data Documentation

3.12.2.1 maxPktSize

```
unsigned short Pipe_Config::maxPktSize
```

Max packet size

3.12.2.2 xferType

```
USB_xferType Pipe_Config::xferType
```

Transfer type

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

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.h](#)

3.13 PLL_Class Class Reference

PLL Class.

```
#include <pll.h>
```

Public Member Functions

- [PLL_Class](#) ()
- int [configure](#) (PLL_Config *pConfigInfo)
- int [getConfigure](#) (PLL_Config *pConfigInfo)
- long [getSystemClock](#) ()

3.13.1 Detailed Description

PLL Class.

Contains prototypes for functions in PLL library

3.13.2 Constructor & Destructor Documentation

3.13.2.1 PLL_Class()

```
PLL_Class::PLL_Class ( )
```

PLL pllClass

PLL Class Constructor.

3.13.3 Member Function Documentation

3.13.3.1 configure()

```
int PLL_Class::configure (
    PLL_Config * pConfigInfo )
```

configure(&pllConfig)

Configure PLL PLL_Config *pConfigInfo <- Config Structure, used to configure PLL

3.13.3.2 getConfigure()

```
int PLL_Class::getConfigure (
    PLL_Config * pConfigInfo )
```

getConfigure(&pllConfig)

Get PLL Configuration Values PLL_Config *pConfigInfo <- Config Structure, used to store the PLL configurations

3.13.3.3 `getSystemClock()`

```
long PLL_Class::getSystemClock ( )
```

[getSystemClock\(\)](#)

Get System Clock frequency, it will return the CPU clock in terms of kHz

The documentation for this class was generated from the following files:

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/PLL/pll.h](#)
- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/PLL/pll.cpp](#)

3.14 SAR_Class Class Reference

SAR Class.

```
#include <SAR.h>
```

Public Member Functions

- int [begin](#) ()
- int [end](#) ()
- int [configChannel](#) (int opmode, int chanNum, int conversion)
- int [startConversion](#) ()
- int [stopConversion](#) ()
- Bool [getStatus](#) ()
- unsigned int [readData](#) ()

3.14.1 Detailed Description

SAR Class.

Contains prototypes for functions in SAR library

3.14.2 Member Function Documentation

3.14.2.1 `begin()`

```
int SAR_Class::begin (
    void )
```

=====

[begin\(\)](#)

Description

API to Initialize the SAR Module

Arguments

Return Value

- CSL_SOK - SAR init successful
- CSL_ESYS_INVPARAMS - Invalid parameters

3.14.2.2 configChannel()

```
int SAR_Class::configChannel (
    int opmode,
    int chanNum,
    int conversion )
```

configChannel()

Description

API to Configure the SAR module

Arguments

```
opmode      - SAR Operation Mode
chanNum     - Channel Number to configure
conversion - Conversion mode used to configure the SAR
```

Return Value

- CSL_SOK - SAR Config successful
- CSL_ESYS_INVPARAMS - Invalid parameters

3.14.2.3 end()

```
int SAR_Class::end ( )
```

end()

Description

API to Close and De-initialize the SAR Module

Arguments

Return Value

- CSL_SOK - SAR close successful
- CSL_ESYS_INVPARAMS - Invalid parameters

3.14.2.4 getStatus()

```
Bool SAR_Class::getStatus ( )
```

getStatus()

Description

API to read the status of the ADC conversion

Arguments

Return Value

- TRUE - If Conversion is Complete
- FALSE - If Conversion is Pending

3.14.2.5 readData()

```
unsigned int SAR_Class::readData ( )
```

readData()

Description

API to read the digital data converted by the SAR module

Arguments

Return Value

- The converted Digital data

3.14.2.6 startConversion()

```
int SAR_Class::startConversion ( )
```

startConversion()

Description

API to Start the ADC conversion

Arguments

Return Value

- CSL_SOK - SAR Conversion Start successful
- CSL_ESYS_INVPARAMS - Invalid parameters

3.14.2.7 stopConversion()

```
int SAR_Class::stopConversion ( )
```

stopConversion()

Description

API to Stop the ADC conversion

Arguments

Return Value

- CSL_SOK - SAR Conversion Stop successful
- CSL_ESYS_INVPARAMS - Invalid parameters

The documentation for this class was generated from the following files:

- /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SAR/SAR.h
- /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SAR/SAR.cpp

3.15 SD_Class Class Reference

SD Card Class.

```
#include <SD.h>
```

Public Member Functions

- [SD_Class](#) ()
SD_Class sdClass.
- [~SD_Class](#) ()
delete (sdClass)
- Bool [begin](#) ()
begin()
- Bool [begin](#) (int opmode)
begin()
- Bool [exists](#) (char *fileName)
exists(filePath)
- Bool [mkdir](#) (char *fileName)
mkdir(directoryPath)
- [File open](#) (char *filePath)
open(filePath)
- [File open](#) (char *filePath, [FILE_MODE](#) mode)
open(filePath, mode)
- Bool [remove](#) (char *fileName)
remove(filePath)
- Bool [rmdir](#) (char *fileName)
rmdir(directoryPath)

3.15.1 Detailed Description

SD Card Class.

The SD class provides functions for accessing the SD card and manipulating its files and directories

3.15.2 Constructor & Destructor Documentation

3.15.2.1 SD_Class()

```
SD_Class::SD_Class ( )
```

[SD_Class](#) sdClass.

SD Class Constructor.

Returns

NONE

3.15.2.2 ~SD_Class()

```
SD_Class::~~SD_Class ( )
```

delete (sdClass)

SD Class Destructor.

Returns

NONE

3.15.3 Member Function Documentation

3.15.3.1 begin() [1/2]

```
Bool SD_Class::begin (
    void )
```

[begin\(\)](#)

API to Initialize and Configure the SD Card

Returns

TRUE - Successful in initializing and configuring the SD Card
FALSE - Unsuccessful in initializing and configuring the SD Card

3.15.3.2 begin() [2/2]

```
Bool SD_Class::begin (
    int opmode )
```

begin()

API to Initialize and Configure the SD Card

Returns

TRUE - Successful in initializing and configuring the SD Card
FALSE - Unsuccessful in initializing and configuring the SD Card

3.15.3.3 exists()

```
Bool SD_Class::exists (
    char * filePath )
```

exists(filePath)

API to check whether a particular file exists in the ATA [File](#) System or not

Parameters

<i>filePath</i>	[IN] Path of the File to be checked
-----------------	---

Returns

TRUE - If the file exists
FALSE - If the file doesn't exist

3.15.3.4 mkdir()

```
Bool SD_Class::mkdir (
    char * directoryPath )
```

mkdir(directoryPath)

API to create a new directory

Parameters

<i>directoryPath</i>	[IN] Path of the Directory that is to be created
----------------------	--

Returns

TRUE - Successful in creating the requested directory
FALSE - Unsuccessful in creating the requested directory

3.15.3.5 open() [1/2]

```
File SD_Class::open (
    char * filePath )
```

open(filePath)

API to open an existing file Length of file name accepted by SD library is restricted to 8 characters

Parameters

<i>filePath</i>	[IN] Path of the File that is to be opened
-----------------	--

Returns

Object of the '[File](#)' class of the opened file

3.15.3.6 open() [2/2]

```
File SD_Class::open (
    char * filePath,
    FILE_MODE mode )
```

open(filePath, mode)

API to open/create a new file Length of file name accepted by SD library is restricted to 8 characters

Parameters

<i>filePath</i>	[IN] Path of the File that is to be created
<i>mode</i>	[IN] Mode of the File (Read/Write/RW)

Returns

Object of the '[File](#)' class of the opened file

3.15.3.7 remove()

```
Bool SD_Class::remove (
    char * filePath )
```

remove(filePath)

API to delete a file

Parameters

<i>filePath</i>	[IN] Path of the File to be deleted
-----------------	---

Returns

TRUE - Successful in removing the file FALSE - Unsuccessful in removing the file

3.15.3.8 rmdir()

```
Bool SD_Class::rmdir (
    char * directoryPath )
```

rmdir(directoryPath)

API to delete a directory

Parameters

<i>directoryPath</i>	[IN] Path of the Directory to be deleted
----------------------	--

Returns

TRUE - Successful in removing the directory FALSE - Unsuccessful in removing the directory

The documentation for this class was generated from the following files:

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.h](#)
- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.cpp](#)

3.16 SPI_Class Class Reference

SPI Class.

```
#include <SPI.h>
```

Public Member Functions

- void [begin](#) ()
- void [end](#) ()
- void [setClockDivider](#) (int divider)
- void [setDataMode](#) (int mode)
- void [setLoopBackMode](#) (int value)
- void [setBitOrder](#) (int order)
- int [transfer](#) (int value)
- int [write](#) (unsigned int buffer[], int length)
- int [read](#) (unsigned int buffer[], int length)

3.16.1 Detailed Description

SPI Class.

Contains prototypes for functions in SPI DSP API library

3.16.2 Member Function Documentation

3.16.2.1 begin()

```
void SPI_Class::begin (
    void )
```

begin()

Description

API to Initialize the SPI Module

Arguments

Return Value

None

3.16.2.2 end()

```
void SPI_Class::end ( )
```

end()

Description

API to disable the SPI Module

Arguments

Return Value

None

3.16.2.3 read()

```
int SPI_Class::read (
    unsigned int buffer[],
    int length )
```

read()

Description

API to read an array of data from the SPI

Arguments

buffer [IN] Buffer that will holds the read data
length [IN] Length of the Buffer to be read

Return Value

- CSL_SOK - SPI read successful
- CSL_ESYS_INVPARAMS - Invalid parameters

3.16.2.4 setBitOrder()

```
void SPI_Class::setBitOrder (
    int newOrder )
```

setBitOrder()

Description

API to set the order of the bits shifted out of and into the SPI bus, either Least-significant bit(LSB) first or Most-significant bit(MSB) first.

Arguments

`newOrder` - Value (LSBFIRST or MSBFIRST) that is to be set for the order of data transfer

Return Value

None

3.16.2.5 setClockDivider()

```
void SPI_Class::setClockDivider (
    int divider )
```

setClockDivider()

Description

API to set the SPI clock divider relative to the system clock

Arguments

`divider` - The clock divider value

Return Value

None

3.16.2.6 setDataMode()

```
void SPI_Class::setDataMode (
    int mode )
```

setDataMode()

Description

API to set the clock, polarity and phase

Arguments

`mode` - Mode to be set

Return Value

None

3.16.2.7 setLoopBackMode()

```
void SPI_Class::setLoopBackMode (
    int value )
```

setLoopBackMode()
Description

API to set the Loop Back Mode of the SPI

Arguments

value - Value (0 or 1) that is to be set for the Loop Back field

Return Value

None

3.16.2.8 transfer()

```
int SPI_Class::transfer (
    int value )
```

transfer()
Description

API to transfer one byte over the SPI bus, both sending and receiving

Arguments

value - the byte to send out over the bus

Return Value

- On Success - The byte read from the bus
- On Failure - CSL_ESYS_INVPARAMS

3.16.2.9 write()

```
int SPI_Class::write (
    unsigned int buffer[],
    int length )
```

write()
Description

API to write an array of data to the SPI

Arguments

buffer [IN] Buffer that holds the data to be written
length [IN] Length of the Buffer to be written

Return Value

- CSL_SOK - SPI write successful
- CSL_ESYS_INVPARAMS - Invalid parameters

The documentation for this class was generated from the following files:

- /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/SPI.h
- /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/SPI.cpp

3.17 TimerClass Class Reference

Timers Class.

```
#include <Timers.h>
```

Public Member Functions

- int [selectTimer](#) (unsigned short gptInstance)
- int [configTimer](#) (GPT_Config gptConfig)
- int [configTimer](#) (GPT_Config gptConfig, unsigned short gptInstance)
- void [initialize](#) (long period)
- void [initialize](#) (long period, unsigned short gptInstance)
- void [setPeriod](#) (long period)
- void [setPeriod](#) (long period, unsigned short gptInstance)
- unsigned long [read](#) ()
- unsigned long [read](#) (unsigned short gptInstance)
- void [start](#) ()
- void [start](#) (unsigned short gptInstance)
- void [stop](#) ()
- void [stop](#) (unsigned short gptInstance)
- int [close](#) ()
- int [close](#) (unsigned short gptInstance)
- int [setWdt](#) (unsigned short wdtInstance, WDT_Config wdtConfig)
- int [serviceWdt](#) ()
- int [startWdt](#) ()
- int [stopWdt](#) ()
- int [closeWdt](#) ()
- void [clearInterrupt](#) ()

3.17.1 Detailed Description

Timers Class.

Contains prototypes for functions in Timers DSP API library

3.17.2 Member Function Documentation

3.17.2.1 [clearInterrupt\(\)](#)

```
void TimerClass::clearInterrupt ( )
```

[closeWdt\(\)](#)

Closes the specified handle of watchdog timer

3.17.2.2 close() [1/2]

```
int TimerClass::close (
    void )
```

[close\(\)](#)

Closes the specified handle of GPT

3.17.2.3 close() [2/2]

```
int TimerClass::close (
    unsigned short gptInstance )
```

[close\(unsigned short gptInstance\)](#)

Closes the specified handle of GPT

```
unsigned short timerInstance <- Timers instance number (0 to 2)
```

3.17.2.4 closeWdt()

```
int TimerClass::closeWdt ( )
```

[closeWdt\(\)](#)

Closes the specified handle of watchdog timer

3.17.2.5 configTimer() [1/2]

```
int TimerClass::configTimer (
    GPT_Config gptConfig )
```

[configTimer\(Timers_config\)](#)

Configure the Timers parameters for the given timer.

```
CSL_Config gptConfig <- Timer configurations
```

3.17.2.6 configTimer() [2/2]

```
int TimerClass::configTimer (
    GPT_Config gptConfig,
    unsigned short gptInstance )
```

[configTimer\(GPT_Config gptConfig, unsigned short gptInstance\)](#)

Configure the Timers parameters for the given timer.

```
CSL_Config gptConfig <- Timer configurations
unsigned short timerInstance <- Timers instance number (0 to 2)
```

3.17.2.7 initialize() [1/2]

```
void TimerClass::initialize (
    long period )
```

initialize(period)

Set the given period for the selected timer.

```
long int period <- Timer period
```

3.17.2.8 initialize() [2/2]

```
void TimerClass::initialize (
    long period,
    unsigned short gptInstance )
```

initialize(long period, unsigned short gptInstance)

Set the given period for the selected timer.

```
long int period <- Timer period
unsigned short timerInstance <- Timers instance number (0 to 2)
```

3.17.2.9 read() [1/2]

```
unsigned long TimerClass::read (
    void )
```

read()

Returns timer count value read from the timer.

3.17.2.10 read() [2/2]

```
unsigned long TimerClass::read (
    unsigned short gptInstance )
```

read(unsigned short gptInstance)

Returns timer count value read from the timer.

```
unsigned short timerInstance <- Timers instance number (0 to 2)
```

3.17.2.11 selectTimer()

```
int TimerClass::selectTimer (
    unsigned short gptInstance )
```

selectTimer(GPT_Instance gptInstance)

Initialize Timers Module for the given timer instance.

```
unsigned short timerInstance <- Timers instance number (0 to 2)
```

3.17.2.12 serviceWdt()

```
int TimerClass::serviceWdt ( )
```

[serviceWdt\(\)](#)

Services watch dog timer

3.17.2.13 setPeriod() [1/2]

```
void TimerClass::setPeriod (
    long period )
```

setPeriod(period)

Set the given period for the selected timer.

long period <- Timer period

3.17.2.14 setPeriod() [2/2]

```
void TimerClass::setPeriod (
    long period,
    unsigned short gptInstance )
```

[setPeriod\(long period, unsigned short gptInstance\)](#)

Set the given period for the selected timer.

long period <- Timer period
 unsigned short timerInstance <- Timers instance number (0 to 2)

3.17.2.15 setWdt()

```
int TimerClass::setWdt (
    unsigned short wdtInstance,
    WDT_Config wdtConfig )
```

setWdt(WDTIM_Instance wdtInstance, WDT_Config config)

Configure the Watchdog Timer's parameters for the given instance.

WDTIM_Instance wdtInstance <- Watchdog timer instance
 WDT_Config config <- Watchdog timer configurations

3.17.2.16 start() [1/2]

```
void TimerClass::start ( )
```

[start\(\)](#)

Starts to load value from period register to count down register.

3.17.2.17 start() [2/2]

```
void TimerClass::start (
    unsigned short gptInstance )
```

[start\(unsigned short gptInstance\)](#)

Starts to load value from period register to count down register.

unsigned short timerInstance <- Timers instance number (0 to 2)

3.17.2.18 startWdt()

```
int TimerClass::startWdt ( )
```

[startWdt\(\)](#)

Starts to loading value from counter register to kick register

3.17.2.19 stop() [1/2]

```
void TimerClass::stop ( )
```

[stop\(\)](#)

Stops loading from period register to count down register.

3.17.2.20 stop() [2/2]

```
void TimerClass::stop (
    unsigned short gptInstance )
```

[stop\(unsigned short gptInstance\)](#)

Stops loading from period register to count down register.

unsigned short timerInstance <- Timers instance number (0 to 2)

3.17.2.21 stopWdt()

```
int TimerClass::stopWdt ( )
```

[stopWdt\(\)](#)

Stops loading value from counter register to kick register

The documentation for this class was generated from the following files:

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/Timers.h](#)
- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/Timers.cpp](#)

3.18 USB_Config Class Reference

USB configuration Class.

```
#include <USB.h>
```

Public Attributes

- [USB_opMode](#) opMode
- [USB_APP_CALLBACK](#) appSuspendCallBack
- [USB_APP_INT_CALLBACK](#) rxIntCallback
- [USB_APP_INT_CALLBACK](#) txIntCallback
- [USB_APP_INT_CALLBACK](#) rxCompleteCallback
- unsigned short * [deviceDescPtr](#)
- unsigned short * [deviceQualDescPtr](#)
- unsigned short * [cfgDescPtr](#)
- unsigned short * [cfgDescFSPtr](#)
- unsigned short * [strDescPtr](#) [[USB_STRDESC_COUNT](#)]

3.18.1 Detailed Description

USB configuration Class.

3.18.2 Member Data Documentation

3.18.2.1 appSuspendCallBack

```
USB\_APP\_CALLBACK USB_Config::appSuspendCallBack
```

Callback function for suspend operation

3.18.2.2 cfgDescFSPtr

```
unsigned short* USB_Config::cfgDescFSPtr
```

Full speed config descriptor pointer

3.18.2.3 cfgDescPtr

```
unsigned short* USB_Config::cfgDescPtr
```

Config descriptor pointer

3.18.2.4 deviceDescPtr

```
unsigned short* USB_Config::deviceDescPtr
```

Device descriptor pointer

3.18.2.5 deviceQualDescPtr

```
unsigned short* USB_Config::deviceQualDescPtr
```

Device qualifier descriptor pointer

3.18.2.6 opMode

```
USB_opMode USB_Config::opMode
```

USB operation mode

3.18.2.7 rxCompleteCallback

```
USB_APP_INT_CALLBACK USB_Config::rxCompleteCallback
```

Callback function for RX complete interrupt

3.18.2.8 rxIntCallback

```
USB_APP_INT_CALLBACK USB_Config::rxIntCallback
```

Callback function for RX interrupt

3.18.2.9 strDescPtr

```
unsigned short* USB_Config::strDescPtr[USB_STRDESC_COUNT]
```

String descriptor pointer

3.18.2.10 txIntCallback

```
USB_APP_INT_CALLBACK USB_Config::txIntCallback
```

Callback function for TX interrupt

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

- /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.h

3.19 USBClass Class Reference

USB Class.

```
#include <USB.h>
```


Public Member Functions

- [USBClass](#) ()
- int [init](#) ()
- int [config](#) ([USB_Config](#))
- int [connect](#) ()
- int [disconnect](#) ()
- int [suspend](#) ()
- void * [requestPipe](#) (unsigned short, unsigned short)
- int [closePipe](#) (void *)
- int [configPipe](#) (void *, [Pipe_Config](#))
- int [stallPipe](#) (void *)
- int [clearPipe](#) (void *)
- int [writePipe](#) (void *, unsigned short, unsigned short *)
- int [readPipe](#) (void *, unsigned short, unsigned short *)
- int [handleInterrupts](#) ()
- int [setParams](#) (void *)
- int [dmaRxStop](#) (void *)
- int [dmaTxStop](#) (void *)

Public Attributes

- unsigned short * [usbBuffTxRxPtr1](#)
- unsigned short * [usbBuffTxRxPtr2](#)
- unsigned short * [usbBuffTxRxPtr3](#)
- unsigned short * [usbBuffTxRxPtr4](#)

3.19.1 Detailed Description

USB Class.

Contains prototypes for functions in USB library

3.19.2 Constructor & Destructor Documentation

3.19.2.1 USBClass()

```
USBClass::USBClass ( )
```

USB Class Constructor

3.19.3 Member Function Documentation

3.19.3.1 clearPipe()

```
int USBClass::clearPipe (  
    void * hEp )
```

clearPipe()

Description

This function clears an end point stall

Arguments

```
*      *hEp    <- End point handle  
*
```

Return Value

 int

- 0:Success
- Error status

3.19.3.2 closePipe()

```
int USBClass::closePipe (  
    void * hEp )
```

closePipe()

Description

Function to release an endpoint

Arguments

```
*      *hEp    <- End point handle  
*
```

Return Value

 int

- 0:Success
- Error status

3.19.3.3 config()

```
int USBClass::config (
    USB_Config usbConfig )
```

config()

Description

Configures the USB module

Arguments

```
*      usbConfig    <- Configuration structure
*
*
```

Return Value

 int

- 0:Success
- Error status

3.19.3.4 configPipe()

```
int USBClass::configPipe (
    void * hEp,
    Pipe_Config pipeConfig )
```

configPipe()

Description

Function to configure endpoint

Arguments

```
*      *hEp          <- End point handle
*      pipeConfig    <- End point config structure
*
```

Return Value

 int

- 0:Success
- Error status

3.19.3.5 connect()

```
int USBClass::connect ( )
```

```
=====
```

connect()
Description

This function connects the USB module to upstream port

Arguments

*

Return Value int

- 0:Success
- Error status

3.19.3.6 disconnect()

```
int USBClass::disconnect ( )
```

```
=====
```

disconnect()
Description

This function disconnects the USB module to upstream port

Arguments

*

Return Value int

- 0:Success
- Error status

3.19.3.7 dmaRxStop()

```
int USBClass::dmaRxStop (
    void * hEp )
```

```
=====
```

dmaRxStop()
Description

Stops DMA data receive operation

Arguments

```
*      *hEp    <- End point handle
*
```

Return Value int

- 0:Success
- Error status

3.19.3.8 dmaTxStop()

```
int USBClass::dmaTxStop (
    void * hEp )
```

[dmaTxStop\(\)](#)

Description

Stops DMA data transmit operation

Arguments

```
*      *hEp    <- End point handle
*
```

Return Value int

- 0:Success
- Error status

3.19.3.9 handleInterrupts()

```
int USBClass::handleInterrupts (
    void )
```

[handleInterrupts\(\)](#)

Description

Function to handle the USB interrupts

Arguments

```
*
```

Return Value int

- 0:Success
- Error status

3.19.3.10 init()

```
int USBClass::init (
    void )
```

init()

Description

Function initializes USB module.

Arguments

*

Return Value

 int

- 0:Success
- Error status

3.19.3.11 readPipe()

```
int USBClass::readPipe (
    void * hEp,
    unsigned short bytes,
    unsigned short * data )
```

readPipe()

Description

Function to read data from USB pipe

Arguments

```
*      *hEp  <- End point handle
*      bytes <- Number of bytes to be read
*      *data <- Pointer to data buffer
*
```

Return Value

 int

- 0:Success
- Error status

3.19.3.12 requestPipe()

```
void * USBClass::requestPipe (
    unsigned short pipeNum,
    unsigned short direction )
```

requestPipe()

Description

Function to request the endpoint for data communication

Arguments

```
*     pipeNum    <- Pipe number
*     direction  <- Transfer direction, USB_IN or USB_OUT
*
```

Return Value void *

- Valid pipe handle if success
- NULL if error

3.19.3.13 setParams()

```
int USBClass::setParams (
    void * hEp )
```

setParams()

Description

Function to initialize the endpoint structures

Arguments

```
*     *hEp    <- End point handle
*
```

Return Value int

- 0:Success
- Error status

3.19.3.14 stallPipe()

```
int USBClass::stallPipe (
    void * hEp )
```

stallPipe()

Description

This function stalls an end point

Arguments

```
*      *hEp    <- End point handle
*
```

Return Value

 int

- 0:Success
- Error status

3.19.3.15 suspend()

```
int USBClass::suspend ( )
```

suspend()

Description

This function suspends the USB device. This function informs the application about the device suspend through a call back function.

Arguments

```
*
```

Return Value

 int

- 0:Success
- Error status

3.19.3.16 writePipe()

```
int USBClass::writePipe (
    void * hEp,
    unsigned short bytes,
    unsigned short * data )
```

writePipe()

Description

Function to write data to USB pipe

Arguments

```
*      *hEp  <- End point handle
*      bytes <- Number of bytes to be written
*      *data <- Pointer to data buffer
*
```

Return Value

 int

- 0:Success
- Error status

3.19.4 Member Data Documentation

3.19.4.1 usbBuffTxRxPtr1

```
unsigned short* USBClass::usbBuffTxRxPtr1
```

USB data buffer 1 pointer

3.19.4.2 usbBuffTxRxPtr2

```
unsigned short* USBClass::usbBuffTxRxPtr2
```

USB data buffer 2 pointer

3.19.4.3 usbBuffTxRxPtr3

```
unsigned short* USBClass::usbBuffTxRxPtr3
```

USB data buffer 3 pointer

3.19.4.4 usbBuffTxRxPtr4

```
unsigned short* USBClass::usbBuffTxRxPtr4
```

USB data buffer 4 pointer

The documentation for this class was generated from the following files:

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.h](#)
- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.cpp](#)

3.20 WDT_Config Struct Reference

Configuration structure.

```
#include <Timers.h>
```

Public Attributes

- unsigned short [counter](#)
- unsigned short [prescale](#)

3.20.1 Detailed Description

Configuration structure.

Hardware register configuration structure for WDT

3.20.2 Member Data Documentation

3.20.2.1 counter

```
unsigned short WDT_Config::counter
```

Counter value for the WDT

3.20.2.2 prescale

```
unsigned short WDT_Config::prescale
```

Prescale value for the WDT

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

- [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/Timers.h](#)

Chapter 4

File Documentation

4.1 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/AnalogReadWrite/examples/AnalogRead/AnalogRead.ino](#) File Reference

4.2 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/Audio.cpp](#) File Reference

Audio implementation.

```
#include "Audio.h"
```

Variables

- [AudioClass](#) `AudioC`
- `Uin16` `i2sDmaLeftBuff1` [2][`I2S_DMA_BUF_LEN`]
- `Uin16` `i2sDmaRightBuff1` [2][`I2S_DMA_BUF_LEN`]
- `Uin16` `i2sDmaLeftBuff2` [2][`I2S_DMA_BUF_LEN`]
- `Uin16` `i2sDmaRightBuff2` [2][`I2S_DMA_BUF_LEN`]

4.2.1 Detailed Description

Audio implementation.

4.2.2 Variable Documentation

4.2.2.1 `AudioC`

`AudioClass` `AudioC`

Class identifier declaration

4.2.2.2 i2sDmaLeftBuff1

```
Uint16 i2sDmaLeftBuff1[2][I2S_DMA_BUF_LEN]
```

DMA left channel buffer

4.2.2.3 i2sDmaLeftBuff2

```
Uint16 i2sDmaLeftBuff2[2][I2S_DMA_BUF_LEN]
```

DMA left channel buffer 2

4.2.2.4 i2sDmaRightBuff1

```
Uint16 i2sDmaRightBuff1[2][I2S_DMA_BUF_LEN]
```

DMA right channel buffer

4.2.2.5 i2sDmaRightBuff2

```
Uint16 i2sDmaRightBuff2[2][I2S_DMA_BUF_LEN]
```

DMA right channel buffer 2

4.3 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/↵ Audio.h File Reference

Audio library header file.

```
#include "core.h"
```

Classes

- class [AudioClass](#)
Audio Class.

Macros

- `#define I2S_DMA_BUF_LEN` (512)
- `#define DMA_CHAN_ReadL` (CSL_DMA_CHAN4)
- `#define DMA_CHAN_ReadR` (CSL_DMA_CHAN5)
- `#define DMA_CHAN_WriteL` (CSL_DMA_CHAN6)
- `#define DMA_CHAN_WriteR` (CSL_DMA_CHAN7)
- `#define SAMPLING_RATE_8_KHZ` (8000u)
- `#define SAMPLING_RATE_11_KHZ` (11025u)
- `#define SAMPLING_RATE_12_KHZ` (12000u)
- `#define SAMPLING_RATE_16_KHZ` (16000u)
- `#define SAMPLING_RATE_22_KHZ` (22050u)
- `#define SAMPLING_RATE_24_KHZ` (24000u)
- `#define SAMPLING_RATE_32_KHZ` (32000u)
- `#define SAMPLING_RATE_44_KHZ` (44100u)
- `#define SAMPLING_RATE_48_KHZ` (48000u)
- `#define CHANNEL_MONO` (1)
- `#define CHANNEL_STEREO` (2)

Variables

- [AudioClass AudioC](#)

4.3.1 Detailed Description

Audio library header file.

4.3.2 Macro Definition Documentation

4.3.2.1 CHANNEL_MONO

```
#define CHANNEL_MONO (1)
```

Macro to indicate the Channel type as Mono

4.3.2.2 CHANNEL_STEREO

```
#define CHANNEL_STEREO (2)
```

Macro to indicate the Channel type as Stereo

4.3.2.3 DMA_CHAN_ReadL

```
#define DMA_CHAN_ReadL (CSL_DMA_CHAN4)
```

DMA left read channel

4.3.2.4 DMA_CHAN_ReadR

```
#define DMA_CHAN_ReadR (CSL_DMA_CHAN5)
```

DMA right read channel

4.3.2.5 DMA_CHAN_WriteL

```
#define DMA_CHAN_WriteL (CSL_DMA_CHAN6)
```

DMA left write channel

4.3.2.6 DMA_CHAN_WriteR

```
#define DMA_CHAN_WriteR (CSL_DMA_CHAN7)
```

DMA right write channel

4.3.2.7 I2S_DMA_BUF_LEN

```
#define I2S_DMA_BUF_LEN (512)
```

Buffer length

4.3.2.8 SAMPLING_RATE_11_KHZ

```
#define SAMPLING_RATE_11_KHZ (11025u)
```

Sampling Rate as 11 kHz

4.3.2.9 SAMPLING_RATE_12_KHZ

```
#define SAMPLING_RATE_12_KHZ (12000u)
```

Sampling Rate as 12 kHz

4.3.2.10 SAMPLING_RATE_16_KHZ

```
#define SAMPLING_RATE_16_KHZ (16000u)
```

Sampling Rate as 16 kHz

4.3.2.11 SAMPLING_RATE_22_KHZ

```
#define SAMPLING_RATE_22_KHZ (22050u)
```

Sampling Rate as 22 kHz

4.3.2.12 SAMPLING_RATE_24_KHZ

```
#define SAMPLING_RATE_24_KHZ (24000u)
```

Sampling Rate as 24 kHz

4.3.2.13 SAMPLING_RATE_32_KHZ

```
#define SAMPLING_RATE_32_KHZ (32000u)
```

Sampling Rate as 32 kHz

4.3.2.14 SAMPLING_RATE_44_KHZ

```
#define SAMPLING_RATE_44_KHZ (44100u)
```

Sampling Rate as 44 kHz

4.3.2.15 SAMPLING_RATE_48_KHZ

```
#define SAMPLING_RATE_48_KHZ (48000u)
```

Sampling Rate as 48 kHz

4.3.2.16 SAMPLING_RATE_8_KHZ

```
#define SAMPLING_RATE_8_KHZ (8000u)
```

Sampling Rate as 8 kHz

4.3.3 Variable Documentation

4.3.3.1 AudioC

```
AudioClass AudioC
```

Audio class instance extern which can used by application programs to access Audio DSP APIs

Class identifier declaration

- 4.4 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/AudioLoopback/AudioLoopback.ino](#) File Reference
- 4.5 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Filter/FIR/FIR.ino](#) File Reference
- 4.6 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Filter/IIR/IIR.ino](#) File Reference
- 4.7 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Player/Player.ino](#) File Reference
- 4.8 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Audio/examples/Recorder/Recorder.ino](#) File Reference
- 4.9 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DigitalReadWrite/examples/DigitalWrite/DigitalWrite.ino](#) File Reference
- 4.10 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DigitalReadWrite/examples/GPIO/GPIO.ino](#) File Reference
- 4.11 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DigitalReadWrite/examples/readDIPSwitch/readDIPSwitch.ino](#) File Reference
- 4.12 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/DMA/A/examples/DMA/DMA.ino](#) File Reference
- 4.13 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/examples/dma/dma.ino](#) File Reference
- 4.14 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/coeffdef_fft.h](#) File Reference

Macros

- `#define NX` (1024)
- `#define WINLEN` (64)
- `#define FFTLEN` (2 * WINLEN)

Variables

- short `x_filter` [`NX`]
- short `ir` [`WINLEN`]
- short `fr` [`FFTLEN *2`]

4.14.1 Macro Definition Documentation

4.14.1.1 FFTLEN

```
#define FFTLEN (2 * WINLEN)
```

Size of FFT; $FFTLEN = (\text{data window length}) + (\text{filter order}) - 1$; In this demo, filter order is (data Window length + 1). Therefore, $FFTLEN = 64$

4.14.1.2 NX

```
#define NX (1024)
```

`coeffdef_fft.h`

Created on: Dec 3, 2013 Author: a0132717 Number of data points in the input data.

4.14.1.3 WINLEN

```
#define WINLEN (64)
```

Window size of the data samples

4.14.2 Variable Documentation

4.14.2.1 fr

```
short fr[FFTLEN *2]
```

4.14.2.2 ir

```
short ir[WINLEN]
```

Initial value:

```
= {  
    0, -11, -22, -33, -43, -48, -45, -30, 0, 43, 96, 149, 190, 207, 185, 116, -0, -156, -333, -501, -624, -665, -588, -370, 0,  
    512, 1137, 1827, 2523, 3156, 3664, 3993, 4106, 3993, 3664, 3156, 2523, 1827, 1137, 512, 0, -370, -588, -665, -624, -501, -333, -156,  
    -0, 116, 185, 207, 190, 149, 96, 43, 0, -30, -45, -48, -43, -33, -22, -11  
}
```

4.14.2.3 `x_filter`

```
short x_filter[NX]
```

4.15 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/examples/FFT/FFT.ino](#) File Reference

4.16 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/FFT.cpp](#) File Reference

FFT implementation.

```
#include <stdio.h>
#include "FFT.h"
#include <stdlib.h>
#include <math.h>
#include <dsplib.h>
#include "coeffdef_fft.h"
```

Macros

- `#define ENABLE_FILTER`

Functions

- void `IR2FR` (short `*ir`, short `*fr`)
- long `CPLX_Mul` (short `*op1`, short `*op2`)
- void `downScaleData` (short `*data`, int length, short scale)
- void `upScaleData` (short `*data`, int length, short scale)

Variables

- `FFTClass FFTransform`
- short `buf [FFTLEN *2]`

4.16.1 Detailed Description

FFT implementation.

4.16.2 Macro Definition Documentation

4.16.2.1 `ENABLE_FILTER`

```
#define ENABLE_FILTER
```

Macro to enable/disable Filtering

4.16.3 Function Documentation

4.16.3.1 CPLX_Mul()

```
long CPLX_Mul (
    short * op1,
    short * op2 )
```

CPLX_Mul()

Description

API to perform multiplication between two complex numbers

Arguments

```
op1 - First Input to be used in Complex multiplication
op2 - Second Input to be used in Complex multiplication
```

Return Value

- The product of the complex numbers, passed as inputs to this API

4.16.3.2 downScaleData()

```
void downScaleData (
    short * data,
    int length,
    short scale )
```

downScaleData()

Description

API to down scale data by an amount specified by 'scale' parameter

Arguments

```
data - Data buffer
length - Length of data
scale - Scaling factor
```

Return Value none

4.16.3.3 IR2FR()

```
void IR2FR (
    short * ir,
    short * fr )
```

IR2FR()

Description

API to convert samples from Real Impulse response to complex Frequency response. Real and imaginary parts are alternated

Arguments

ir - Input Buffer containing audio samples in Real form
fr - Output Buffer to hold the converted audio samples in complex form

Return Value

None

4.16.3.4 upScaleData()

```
void upScaleData (
    short * data,
    int length,
    short scale )
```

upScaleData()

Description

API to up scale data by an amount specified by 'scale' parameter

Arguments

data - Data buffer
length - Length of data
scale - Scaling factor

Return Value none

4.16.4 Variable Documentation

4.16.4.1 buf

```
short buf[FFTLN *2]
```

Buffer for FFT processing

4.16.4.2 FFTTransform

[FFTClass](#) FFTTransform

Class identifier declaration

4.17 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/FFT/FFT.h File Reference

FFT library header file.

```
#include "tistdtypes.h"
#include "dsplib.h"
#include "core.h"
```

Classes

- class [FFTClass](#)
FFT Class.

Macros

- #define [FFT_REAL](#) (0)
- #define [FFT_COMPLEX](#) (1)
- #define [FFTQ15](#) (0)
- #define [FFTQ31](#) (1)
- #define [FFT_BAD_PARAMS](#) (-1)
- #define [FFT_SUCCESS](#) (0)
- #define [CONVERT_32_TO_16_MASK](#) (0x0000FFFF)
- #define [CONVERT_32_TO_16_SHIFT](#) (16)
- #define [Q15_MULT_SHIFT](#) (15)
- #define [ENABLE_SCALE](#) (1)
- #define [DISABLE_SCALE](#) (0)

Variables

- [FFTClass](#) FFTTransform

4.17.1 Detailed Description

FFT library header file.

4.17.2 Macro Definition Documentation

4.17.2.1 CONVERT_32_TO_16_MASK

```
#define CONVERT_32_TO_16_MASK (0x0000FFFF)
```

Mask required to convert a 32 bit value to 16 bit

4.17.2.2 CONVERT_32_TO_16_SHIFT

```
#define CONVERT_32_TO_16_SHIFT (16)
```

Shift required to convert a 32 bit value to 16 bit

4.17.2.3 DISABLE_SCALE

```
#define DISABLE_SCALE (0)
```

Scaling is disabled

4.17.2.4 ENABLE_SCALE

```
#define ENABLE_SCALE (1)
```

Scaling is enabled

4.17.2.5 FFT_BAD_PARAMS

```
#define FFT_BAD_PARAMS (-1)
```

Parameters are invalid

4.17.2.6 FFT_COMPLEX

```
#define FFT_COMPLEX (1)
```

Fast Fourier Transform for complex inputs

4.17.2.7 FFT_REAL

```
#define FFT_REAL (0)
```

Fast Fourier Transform for real inputs

4.17.2.8 FFT_SUCCESS

```
#define FFT_SUCCESS (0)
```

Fast Fourier Transform is successful

4.17.2.9 FFTQ15

```
#define FFTQ15 (0)
```

Fast Fourier Transform for 16 bit inputs in Q15 format

4.17.2.10 FFTQ31

```
#define FFTQ31 (1)
```

Fast Fourier Transform for 32 bit inputs in Q31 format

4.17.2.11 Q15_MULT_SHIFT

```
#define Q15_MULT_SHIFT (15)
```

Shift required to convert a 32 bit value to Q15 format

4.17.3 Variable Documentation

4.17.3.1 FFTransform

```
FFTCClass FFTransform
```

FFT class instance extern which can used by application programs to access FFT DSP APIs

Class identifier declaration

4.18 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Interrupt/examples/dma↔

Interrupt/dmaInterrupt.ino File Reference

4.19 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Interrupt/examples/io↔

ExpanderInterrupt/ioExpanderInterrupt.ino File Reference

4.20 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLE↔

D/bitmapDb.h File Reference

Bitmap database for [OLED](#) module.

```
#include "tistdtypes.h"
```

Classes

- struct [FONT_CHAR_INFO](#)
This structure describes a single character's display information.
- struct [FONT_INFO](#)
This structure describes a single font.

4.20.1 Detailed Description

Bitmap database for [OLED](#) module.

4.21 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/Examples/OledBasic/OledBasic.ino](#) File Reference

4.22 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/Examples/OledPrint/OledPrint.ino](#) File Reference

4.23 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/Examples/OledSerialControl/OledSerialControl.ino](#) File Reference

4.24 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/Examples/OledSerialMsg/OledSerialMsg.ino](#) File Reference

4.25 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/OLED.cpp](#) File Reference

[OLED](#) implementation.

```
#include "OLED.h"
```

Variables

- [OLED disp](#)

4.25.1 Detailed Description

[OLED](#) implementation.

4.25.2 Variable Documentation

4.25.2.1 disp

[OLED](#) disp

Class identifier declaration

4.26 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/OLED.h File Reference

[OLED](#) library header file.

```
#include "core.h"
#include "bitmapDb.h"
```

Classes

- class [OLED](#)
[OLED](#) Class.

Typedefs

- typedef enum [NUMBER_BASE](#) [NUMBER_BASE](#)
Enumeration for number format.

Enumerations

- enum [NUMBER_BASE](#) { [BINARY](#), [DECIMAL](#), [OCTAL](#), [HEXADECIMAL](#) }
Enumeration for number format.

Variables

- [OLED](#) disp

4.26.1 Detailed Description

[OLED](#) library header file.

4.26.2 Typedef Documentation

4.26.2.1 NUMBER_BASE

```
typedef enum NUMBER\_BASE NUMBER\_BASE
```

Enumeration for number format.

4.26.3 Enumeration Type Documentation

4.26.3.1 NUMBER_BASE

```
enum NUMBER\_BASE
```

Enumeration for number format.

Enumerator

BINARY	
DECIMAL	
OCTAL	
HEXADECIMAL	

4.26.4 Variable Documentation

4.26.4.1 disp

`OLED disp`

`OLED` class instance extern which can used by application programs to access `OLED` DSP APIs

Class identifier declaration

4.27 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/OLED/temp.h File Reference

```
#include "ezdsp5535_lcd.h"
```

Functions

- int `oled_test` ()
- void `setup` ()
- void `loop` ()

4.27.1 Function Documentation

4.27.1.1 loop()

```
void loop ( )
```

4.27.1.2 oled_test()

```
int oled_test ( )
```

4.27.1.3 setup()

```
void setup ( )
```

4.28 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/PLL/pll.cpp File Reference

PLL implementation.

```
#include "pll.h"
```

Variables

- [PLL_Class PLL](#)

4.28.1 Detailed Description

PLL implementation.

4.28.2 Variable Documentation

4.28.2.1 PLL

[PLL_Class PLL](#)

Defining a global PLL Class Object which can be used by the User Application

4.29 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/PLL/pll.h File Reference

PLL library header file.

```
#include <stdio.h>
#include <stdlib.h>
#include "csl_pllAux.h"
#include "csl_sysctrl.h"
```

Classes

- class [PLL_Class](#)
PLL Class.

Variables

- [PLL_Class PLL](#)

4.29.1 Detailed Description

PLL library header file.

4.29.2 Variable Documentation

4.29.2.1 PLL

[PLL_Class](#) PLL

PLL class instance extern which can used by application programs to access PLL DSP APIs

Defining a global PLL Class Object which can be used by the User Application

4.30 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/RTC](#) [C/examples/RTC/RTC.ino](#) File Reference

4.31 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SAR/S](#) [AR.cpp](#) File Reference

SAR implementation.

```
#include <core.h>
#include <SAR.h>
```

Functions

- void [SAR_LOG_MSG_PRINT](#) (char *printString)
- void [SAR_LOG_MSG_PRINT](#) (int integer)

Variables

- [SAR_Class SAR](#)

4.31.1 Detailed Description

SAR implementation.

4.31.2 Function Documentation

4.31.2.1 SAR_LOG_MSG_PRINT() [1/2]

```
void SAR_LOG_MSG_PRINT (
    char * printString )
```

Macro to enable the Print messages to be displayed on the Serial =====

[SAR_LOG_MSG_PRINT\(\)](#)

Description

API to display debug messages

Arguments

`printString` - String to be displayed

Return Value

None

=====

[SAR_LOG_MSG_PRINT\(\)](#)

Description

API to display debug messages

Arguments

`printString` - String to be displayed

Return Value

None

4.31.2.2 SAR_LOG_MSG_PRINT() [2/2]

```
void SAR_LOG_MSG_PRINT (
    int integer )
```

=====

[SAR_LOG_MSG_PRINT\(\)](#)

Description

API to display debug messages

Arguments

`integer` - Integer to be displayed

Return Value

None

4.31.3 Variable Documentation

4.31.3.1 SAR

[SAR_Class](#) SAR

Defining a global SAR Class Object which can be used by the User Application

4.32 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SAR/SAR.h File Reference

SAR library header file.

```
#include <stdio.h>
#include "csl_sar.h"
```

Classes

- class [SAR_Class](#)
SAR Class.

Macros

- #define [SAR_POLL_MODE](#) (0)
- #define [SAR_INTERRUPT_MODE](#) (1)
- #define [SAR_DMA_MODE](#) (2)
- #define [SAR_CHANNEL0](#) (0)
- #define [SAR_CHANNEL1](#) (1)
- #define [SAR_CHANNEL2](#) (2)
- #define [SAR_CHANNEL3](#) (3)
- #define [SAR_CHANNEL4](#) (4)
- #define [SAR_CHANNEL5](#) (5)
- #define [SAR_CONTINUOUS_CONVERSION](#) (0)
- #define [SAR_SINGLE_CONVERSION](#) (1)

Variables

- [SAR_Class SAR](#)

4.32.1 Detailed Description

SAR library header file.

4.32.2 Macro Definition Documentation

4.32.2.1 SAR_CHANNEL0

```
#define SAR_CHANNEL0 (0)
```

Macro to indicate Channel No 0 for SAR

4.32.2.2 SAR_CHANNEL1

```
#define SAR_CHANNEL1 (1)
```

Macro to indicate Channel No 1 for SAR

4.32.2.3 SAR_CHANNEL2

```
#define SAR_CHANNEL2 (2)
```

Macro to indicate Channel No 2 for SAR

4.32.2.4 SAR_CHANNEL3

```
#define SAR_CHANNEL3 (3)
```

Macro to indicate Channel No 3 for SAR

4.32.2.5 SAR_CHANNEL4

```
#define SAR_CHANNEL4 (4)
```

Macro to indicate Channel No 4 for SAR

4.32.2.6 SAR_CHANNEL5

```
#define SAR_CHANNEL5 (5)
```

Macro to indicate Channel No 5 for SAR

4.32.2.7 SAR_CONTINUOUS_CONVERSION

```
#define SAR_CONTINUOUS_CONVERSION (0)
```

Macro to indicate SAR to perform Continuous Conversion

4.32.2.8 SAR_DMA_MODE

```
#define SAR_DMA_MODE (2)
```

Macro to indicate the mode of SAR as DMA Mode

4.32.2.9 SAR_INTERRUPT_MODE

```
#define SAR_INTERRUPT_MODE (1)
```

Macro to indicate the mode of SAR as Interrupt Mode

4.32.2.10 SAR_POLL_MODE

```
#define SAR_POLL_MODE (0)
```

Macro to indicate the mode of SAR as Polled Mode

4.32.2.11 SAR_SINGLE_CONVERSION

```
#define SAR_SINGLE_CONVERSION (1)
```

Macro to indicate SAR to perform Single Conversion

4.32.3 Variable Documentation

4.32.3.1 SAR

```
SAR_Class SAR
```

SAR class instance extern which can be used by application programs to access SAR DSP APIs

Defining a global SAR Class Object which can be used by the User Application

4.33 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/chk_mmc.c](#) File Reference

```
#include "chk_mmc.h"
```

Functions

- `AtaUInt32 getMMCSize (AtaState *pAtaDrive)`
- `MMC_ERR_int16 chk_mmc (AtaState *pAtaDrive, unsigned int *disk_type)`
- `MMC_ERR_int16 Check_boot_record (unsigned long boot_record_sector, unsigned long ref_num_of_sectors, unsigned int boot_record_type, AtaState *pAtaDrive, BR_struct *pBootRecord)`

Variables

- [MBR_struct master_boot_record](#)
- [BR_struct boot_record](#)

4.33.1 Function Documentation

4.33.1.1 Check_boot_record()

```
MMC_ERR_int16 Check_boot_record (
    unsigned long boot_record_sector,
    unsigned long ref_num_of_sectors,
    unsigned int boot_record_type,
    AtaState * pAtaDrive,
    BR_struct * pBootRecord )
```

MMC_ERR_int16 Check_boot_record(unsigned long boot_record_sector, unsigned long ref_num_of_sectors, unsigned int boot_record_type, AtaState *pAtaDrive, [BR_struct](#) *pBootRecord

boot_record_sector -> The sector number to fetch the BR from. ref_num_of_sectors -> The reference number of sectors on disk to validate BR against. boot_record_type -> Indicates whether this is floppy type or hard disk type boot record. pAtaDrive -> Pointer to initialised AtaState structure. pBootRecord -> 512 byte buffer to read boot record.

4.33.1.2 chk_mmc()

```
MMC_ERR_int16 chk_mmc (
    AtaState * pAtaDrive,
    unsigned int * disk_type )
```

Function prototypes

4.33.1.3 getMMCSize()

```
AtaUInt32 getMMCSize (
    AtaState * pAtaDrive )
```

[getMMCSize\(AtaState *pAtaDrive\)](#)

This function will returns total number of sectors in the disk, Not the actual size of the disk

4.33.2 Variable Documentation

4.33.2.1 boot_record

```
BR\_struct boot_record
```

4.33.2.2 master_boot_record

`MBR_struct` `master_boot_record`

4.34 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/chk_↵ _mmc.h File Reference

Header file for MMC checking routines.

```
#include "ata.h"
#include "csl_mmc_ssd_ataIf.h"
#include "ata_.h"
```

Classes

- struct [PARTITION_TABLE](#)
- struct [MBR_struct](#)
- struct [BR_struct](#)

Macros

- `#define` [HD_BOOT_RECORD](#) 0
- `#define` [FD_BOOT_RECORD](#) 1

Enumerations

- enum [MMC_ERR_int16](#) {
[MMC_ERR_NONE](#), [MMC_ERR_BAD_CSD_FILE_FORMAT](#), [MMC_ERR_CARD_NOT_READABLE](#), [MMC_↵
C_ERR_MBR_BAD_NUM_PART_SECTORS](#),
[MMC_ERR_MBR_BAD_PARTITIONS_234](#), [MMC_ERR_MBR_DSKSIZE_MISMATCH](#), [MMC_ERR_BR_↵
BAD_JMP_OPCODE](#), [MMC_ERR_BR_BAD_BYTES_PER_SECTOR](#),
[MMC_ERR_BR_BAD_SECTORS_PER_CLUSTER](#), [MMC_ERR_BR_BAD_RESERVED_SECTORS](#), [MMC_↵
C_ERR_BR_BAD_NUM_OF_ROOT_ENTRIES](#), [MMC_ERR_BR_BAD_NUM_PART_SECTORS](#),
[MMC_ERR_BR_BAD_SECTORS_PER_FAT](#), [MMC_ERR_BR_BAD_FILE_SYS_TYPE](#), [MMC_ERR_BR_↵
BAD_SIGNATURE](#) }

Functions

- [MMC_ERR_int16](#) `chk_mmc` (`AtaState *pAtaDrive`, `unsigned int *disk_type`)
- [MMC_ERR_int16](#) `Check_boot_record` (`unsigned long boot_record_sector`, `unsigned long ref_num_of_↵
sectors`, `unsigned int boot_record_type`, `AtaState *pAtaDrive`, [BR_struct](#) *pBootRecord)
- `AtaUInt32` `getMMCSize` (`AtaState *pAtaDrive`)
- `AtaError` `mmc_format` (`AtaState *pAtaDrive`, `AtaUInt16 *MBRptr`, `AtaUInt16 *BRptr`)

4.34.1 Detailed Description

Header file for MMC checking routines.

4.34.2 Macro Definition Documentation

4.34.2.1 FD_BOOT_RECORD

```
#define FD_BOOT_RECORD 1
```

FD boot record

4.34.2.2 HD_BOOT_RECORD

```
#define HD_BOOT_RECORD 0
```

HD boot record

4.34.3 Enumeration Type Documentation

4.34.3.1 MMC_ERR_int16

```
enum MMC_ERR_int16
```

Return Values of chk_mmc.

Enumerator

MMC_ERR_NONE	
MMC_ERR_BAD_CSD_FILE_FORMAT	
MMC_ERR_CARD_NOT_READABLE	
MMC_ERR_MBR_BAD_NUM_PART_SECTORS	
MMC_ERR_MBR_BAD_PARTITIONS_234	
MMC_ERR_MBR_DSKSIZE_MISMATCH	
MMC_ERR_BR_BAD_JMP_OPCODE	
MMC_ERR_BR_BAD_BYTES_PER_SECTOR	
MMC_ERR_BR_BAD_SECTORS_PER_CLUSTER	
MMC_ERR_BR_BAD_RESERVED_SECTORS	
MMC_ERR_BR_BAD_NUM_OF_ROOT_ENTRIES	
MMC_ERR_BR_BAD_NUM_PART_SECTORS	
MMC_ERR_BR_BAD_SECTORS_PER_FAT	
MMC_ERR_BR_BAD_FILE_SYS_TYPE	
MMC_ERR_BR_BAD_SIGNATURE	

4.34.4 Function Documentation

4.34.4.1 Check_boot_record()

```
MMC_ERR_int16 Check_boot_record (
    unsigned long boot_record_sector,
```

```

    unsigned long ref_num_of_sectors,
    unsigned int boot_record_type,
    AtaState * pAtaDrive,
    BR_struct * pBootRecord )

```

MMC_ERR_int16 Check_boot_record(unsigned long boot_record_sector, unsigned long ref_num_of_sectors, unsigned int boot_record_type, AtaState *pAtaDrive, BR_struct *pBootRecord

boot_record_sector -> The sector number to fetch the BR from. ref_num_of_sectors -> The reference number of sectors on disk to validate BR against. boot_record_type -> Indicates whether this is floppy type or hard disk type boot record. pAtaDrive -> Pointer to initialised AtaState structure. pBootRecord -> 512 byte buffer to read boot record.

4.34.4.2 chk_mmc()

```

MMC_ERR_int16 chk_mmc (
    AtaState * pAtaDrive,
    unsigned int * disk_type )

```

Function prototypes

4.34.4.3 getMMCSize()

```

AtaUInt32 getMMCSize (
    AtaState * pAtaDrive )

```

[getMMCSize\(AtaState *pAtaDrive\)](#)

This function will returns total number of sectors in the disk, Not the actual size of the disk

4.34.4.4 mmc_format()

```

AtaError mmc_format (
    AtaState * pAtaDrive,
    AtaUInt16 * MBRptr,
    AtaUInt16 * BRptr )

```

[mmc_format\(AtaState *pAtaDrive, AtaUInt16 *MBRptr, AtaUInt16 *BRptr\)](#)

This function will format the disk

4.35 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/DisplayContents/DisplayContents.ino File Reference](#)

Reference

115

4.35 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/DisplayContents/DisplayContents.ino File Reference](#)

4.36 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Exists/Exists.ino File Reference](#)

4.37 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/FileFolderCreation/FileFolderCreation.ino File Reference](#)

4.38 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Filesize/Filesize.ino File Reference](#)

4.39 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Peek/Peek.ino File Reference](#)

4.40 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Position/Position.ino File Reference](#)

4.41 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/Print/Print.ino File Reference](#)

4.42 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Print/Print.ino File Reference](#)

4.43 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/examples/ReadWrite/ReadWrite.ino File Reference](#)

4.44 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.cpp File Reference](#)

SD/File implementation.

```
#include <SD.h>
#include <core.h>
```

Classes

- class [fileNodesList](#)

Node of the Linked List to hold the details of all the Opened files.

Macros

- #define `CSL_SD_CLOCK_MAX_KHZ` (20000u)
- #define `CSL_MMCSL_ATA_BUF_SIZE` (512u)
- #define `WRITE_CACHE_SIZE` (512)

Functions

- void `LOG_MSG_print` (char *printString)
API to print Debug messages.

Variables

- `fileNodesList` * `fileListHeadNode`
- `fileNodesList` * `fileListLastNode`
- `SD_Class` `SD`
- `AtaUint16` `AtaWrBuf` [`CSL_MMCSL_ATA_BUF_SIZE`]
- `char` `writeCacheBuffer` [`WRITE_CACHE_SIZE+1`]

4.44.1 Detailed Description

SD/File implementation.

4.44.2 Macro Definition Documentation

4.44.2.1 `CSL_MMCSL_ATA_BUF_SIZE`

```
#define CSL_MMCSL_ATA_BUF_SIZE (512u)
```

Macro to enable the Print messages to be displayed on the Serial Macro to indicate the Size of Data buffer used by ATA File System for file Read and Write Operations

4.44.2.2 `CSL_SD_CLOCK_MAX_KHZ`

```
#define CSL_SD_CLOCK_MAX_KHZ (20000u)
```

Macro to indicate the maximum clock value for the SD/MMC Card

4.44.2.3 `WRITE_CACHE_SIZE`

```
#define WRITE_CACHE_SIZE (512)
```

Macro to indicate write cache size

4.44.3 Function Documentation

4.44.3.1 LOG_MSG_print()

```
void LOG_MSG_print (
    char * printString )
```

API to print Debug messages.

This API will print the messages to either the Serial or the Output Console

4.44.4 Variable Documentation

4.44.4.1 AtaWrBuf

```
AtaUInt16 AtaWrBuf[CSL\_MMCSD\_ATA\_BUF\_SIZE]
```

Data Buffer used for Read and Write Operations by the ATA [File](#) System

4.44.4.2 fileListHeadNode

```
fileNodesList* fileListHeadNode
```

Pointer to the first node in the Linked List of [File](#) Nodes

4.44.4.3 fileListLastNode

```
fileNodesList* fileListLastNode
```

Pointer to the last node in the Linked List of [File](#) Nodes

4.44.4.4 SD

```
SD\_Class SD
```

Defining a global SD Class Object which can be used by the User Application

4.44.4.5 writeCacheBuffer

```
char writeCacheBuffer[WRITE\_CACHE\_SIZE+1]
```

Data Buffer used for DSP API write operations

4.45 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SD/SD.h File Reference

SD library header file.

```
#include <string.h>
#include <stdio.h>
#include <ctype.h>
#include "ata.h"
#include "csl_mmc.h"
#include "csl_sysctrl.h"
#include "csl_pll.h"
#include "chk_mmc.h"
```

Classes

- class [File](#)
File Class.
- class [SD_Class](#)
SD Card Class.

Typedefs

- typedef enum [FILE_MODE](#) [FILE_MODE](#)
Enum to indicate the Mode of the file for opening.
- typedef enum [NUMBER_FORMAT_BASE](#) [NUMBER_FORMAT_BASE](#)
Enum to indicate base format for an integer.

Enumerations

- enum [FILE_MODE](#) { [FILE_READ](#), [FILE_WRITE](#), [FILE_APPEND](#) }
Enum to indicate the Mode of the file for opening.
- enum [NUMBER_FORMAT_BASE](#) { [FILE_BIN](#), [FILE_OCT](#), [FILE_DEC](#), [FILE_HEX](#) }
Enum to indicate base format for an integer.

Variables

- [SD_Class](#) [SD](#)

4.45.1 Detailed Description

SD library header file.

4.45.2 Typedef Documentation

4.45.2.1 FILE_MODE

```
typedef enum FILE_MODE FILE_MODE
```

Enum to indicate the Mode of the file for opening.

4.45.2.2 NUMBER_FORMAT_BASE

```
typedef enum NUMBER_FORMAT_BASE NUMBER_FORMAT_BASE
```

Enum to indicate base format for an integer.

4.45.3 Enumeration Type Documentation

4.45.3.1 FILE_MODE

```
enum FILE_MODE
```

Enum to indicate the Mode of the file for opening.

Enumerator

FILE_READ	
FILE_WRITE	
FILE_APPEND	

4.45.3.2 NUMBER_FORMAT_BASE

```
enum NUMBER_FORMAT_BASE
```

Enum to indicate base format for an integer.

Enumerator

FILE_BIN	
FILE_OCT	
FILE_DEC	
FILE_HEX	

4.45.4 Variable Documentation

4.45.4.1 SD

```
SD_Class SD
```

SD class instance extern which can used by application programs to access SD DSP APIs

Defining a global SD Class Object which can be used by the User Application

- 4.46 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Find/Find.ino](#) File Reference
- 4.47 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Finduntil/Finduntil.ino](#) File Reference
- 4.48 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/ParseFloat/ParseFloat.ino](#) File Reference
- 4.49 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/ParseInt/ParseInt.ino](#) File Reference
- 4.50 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/PrintFormat/PrintFormat.ino](#) File Reference
- 4.51 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/Read/Read.ino](#) File Reference
- 4.52 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/ReadBytes/ReadBytes.ino](#) File Reference
- 4.53 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Serial/examples/ReadBytesUntil/ReadBytesUntil.ino](#) File Reference
- 4.54 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/examples/Arduino_SPI/Arduino_SPI.ino](#) File Reference
- 4.55 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/examples/DSP_SPI/DSP_SPI.ino](#) File Reference
- 4.56 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/SPI.cpp](#) File Reference

SPI implementation.

```
#include <core.h>
#include <SPI.h>
```

Functions

- void [SPI_LOG_MSG_PRINT](#) (char *printString)

Variables

- [SPI_Class SPI](#)

4.56.1 Detailed Description

SPI implementation.

4.56.2 Function Documentation

4.56.2.1 SPI_LOG_MSG_PRINT()

```
void SPI_LOG_MSG_PRINT (  
    char * printString )
```

Macro to enable the Print messages to be displayed on the Serial =====

[SPI_LOG_MSG_PRINT\(\)](#)

Description

API to display debug messages

Arguments

`printString` - String to be displayed

Return Value

None

=====

[SPI_LOG_MSG_PRINT\(\)](#)

Description

API to display debug messages

Arguments

`printString` - String to be displayed

Return Value

None

4.56.3 Variable Documentation

4.56.3.1 SPI

[SPI_Class](#) SPI

Defining a global SPI Class Object which can be used by the User Application

4.57 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/SPI/SPI.h File Reference

SPI library header file.

```
#include <stdio.h>
#include "csl_spi.h"
```

Classes

- class [SPI_Class](#)
SPI Class.

Macros

- #define [SPI_CLOCK_DIV2](#) (2)
- #define [SPI_CLOCK_DIV4](#) (4)
- #define [SPI_CLOCK_DIV8](#) (8)
- #define [SPI_CLOCK_DIV16](#) (16)
- #define [SPI_CLOCK_DIV32](#) (32)
- #define [SPI_CLOCK_DIV64](#) (64)
- #define [SPI_CLOCK_DIV128](#) (128)
- #define [SPI_CLK_DIV](#) (25)
- #define [SPI_FRAME_LENGTH](#) (1)
- #define [SPI_MODE0](#) (0)
- #define [SPI_MODE1](#) (1)
- #define [SPI_MODE2](#) (2)
- #define [SPI_MODE3](#) (3)
- #define [LSBFIRST](#) (0)
- #define [MSBFIRST](#) (1)

Variables

- [SPI_Class](#) SPI

4.57.1 Detailed Description

SPI library header file.

4.57.2 Macro Definition Documentation

4.57.2.1 LSBFIRST

```
#define LSBFIRST (0)
```

Macro to indicate the Order of Data transfer as LSB first

4.57.2.2 MSBFIRST

```
#define MSBFIRST (1)
```

Macro to indicate the Order of Data transfer as MSB first

4.57.2.3 SPI_CLK_DIV

```
#define SPI_CLK_DIV (25)
```

SPI Clock Divisor

4.57.2.4 SPI_CLOCK_DIV128

```
#define SPI_CLOCK_DIV128 (128)
```

Macro for SPI clock divider value 128

4.57.2.5 SPI_CLOCK_DIV16

```
#define SPI_CLOCK_DIV16 (16)
```

Macro for SPI clock divider value 16

4.57.2.6 SPI_CLOCK_DIV2

```
#define SPI_CLOCK_DIV2 (2)
```

Macro for SPI clock divider value 2

4.57.2.7 SPI_CLOCK_DIV32

```
#define SPI_CLOCK_DIV32 (32)
```

Macro for SPI clock divider value 32

4.57.2.8 SPI_CLOCK_DIV4

```
#define SPI_CLOCK_DIV4 (4)
```

Macro for SPI clock divider value 4

4.57.2.9 SPI_CLOCK_DIV64

```
#define SPI_CLOCK_DIV64 (64)
```

Macro for SPI clock divider value 64

4.57.2.10 SPI_CLOCK_DIV8

```
#define SPI_CLOCK_DIV8 (8)
```

Macro for SPI clock divider value 8

4.57.2.11 SPI_FRAME_LENGTH

```
#define SPI_FRAME_LENGTH (1)
```

SPI Frame length

4.57.2.12 SPI_MODE0

```
#define SPI_MODE0 (0)
```

Macro to indicate SPI mode 0

4.57.2.13 SPI_MODE1

```
#define SPI_MODE1 (1)
```

Macro to indicate SPI mode 1

4.57.2.14 SPI_MODE2

```
#define SPI_MODE2 (2)
```

Macro to indicate SPI mode 2

4.57.2.15 SPI_MODE3

```
#define SPI_MODE3 (3)
```

Macro to indicate SPI mode 3 The above modes are for the following Clock Polarity(CPOL) and Clock

MODE CPOL CPHA

SPI_MODE0 0 0 SPI_MODE1 0 1 SPI_MODE2 1 0

SPI_MODE3 1 1

4.57.3 Variable Documentation

4.57.3.1 SPI

[SPI_Class](#) SPI

SPI class instance extern which can be used by application programs to access SPI DSP APIs

Defining a global SPI Class Object which can be used by the User Application

4.58 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/↔ Timer1/Timer1.ino File Reference

4.59 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/↔ Timer2/Timer2.ino File Reference

4.60 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/↔ Timer3/Timer3.ino File Reference

4.61 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/examples/↔ WatchdogTimer/WatchdogTimer.ino File Reference

4.62 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/↔ Timers.cpp File Reference

Timers implementation.

```
#include "Timers.h"
```

Variables

- [CSL_GptObj](#) [gptObj](#) [[TIMER_INSTANCE_COUNT](#)]
- [CSL_Handle](#) [hGpt](#)
- [CSL_WdtHandle](#) [hWdt](#)
- [CSL_WdtObj](#) [wdtObj](#)
- [TimerClass](#) [Timer](#)

4.62.1 Detailed Description

Timers implementation.

4.62.2 Variable Documentation

4.62.2.1 gptObj

```
CSL_GptObj gptObj[TIMER_INSTANCE_COUNT]
```

GPT object

4.62.2.2 hGpt

```
CSL_Handle hGpt
```

GPT handle

4.62.2.3 hWdt

```
CSL_WdtHandle hWdt
```

WDT handle

4.62.2.4 Timer

```
TimerClass Timer
```

Class identifier declaration

4.62.2.5 wdtObj

```
CSL_WdtObj wdtObj
```

WDT object

4.63 [/Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Timers/](#) Timers.h File Reference

Timers library header file.

```
#include "core.h"  
#include "csl_gpt.h"  
#include "csl_wdt.h"
```


Classes

- struct [GPT_Config](#)
Configuration structure.
- struct [WDT_Config](#)
Configuration structure.
- class [TimerClass](#)
Timers Class.

Macros

- #define [GPT0](#) (0)
- #define [GPT1](#) (1)
- #define [GPT2](#) (2)
- #define [TIMER_INSTANCE_COUNT](#) (3)
- #define [GPT_PRE_SC_DIV_0](#) (0)
- #define [GPT_PRE_SC_DIV_1](#) (1)
- #define [GPT_PRE_SC_DIV_2](#) (2)
- #define [GPT_PRE_SC_DIV_3](#) (3)
- #define [GPT_PRE_SC_DIV_4](#) (4)
- #define [GPT_PRE_SC_DIV_5](#) (5)
- #define [GPT_PRE_SC_DIV_6](#) (6)
- #define [GPT_PRE_SC_DIV_7](#) (7)
- #define [GPT_PRE_SC_DIV_8](#) (8)
- #define [GPT_PRE_SC_DIV_9](#) (9)
- #define [GPT_PRE_SC_DIV_10](#) (10)
- #define [GPT_PRE_SC_DIV_11](#) (11)
- #define [GPT_PRE_SC_DIV_12](#) (12)
- #define [HIGH_WORD_MASK](#) (0xFFFF0000)
- #define [LOW_WORD_MASK](#) (0x0000FFFF)
- #define [WORD_LENGTH](#) (16)

Variables

- [TimerClass Timer](#)

4.63.1 Detailed Description

Timers library header file.

4.63.2 Macro Definition Documentation

4.63.2.1 GPT0

```
#define GPT0 (0)
```

Timer instace 0

4.63.2.2 GPT1

```
#define GPT1 (1)
```

Timer instace 1

4.63.2.3 GPT2

```
#define GPT2 (2)
```

Timer instace 2

4.63.2.4 GPT_PRE_SC_DIV_0

```
#define GPT_PRE_SC_DIV_0 (0)
```

Pre scale Divide input clock by 2

4.63.2.5 GPT_PRE_SC_DIV_1

```
#define GPT_PRE_SC_DIV_1 (1)
```

Pre scale Divide input clock by 4

4.63.2.6 GPT_PRE_SC_DIV_10

```
#define GPT_PRE_SC_DIV_10 (10)
```

Pre scale Divide input clock by 2048

4.63.2.7 GPT_PRE_SC_DIV_11

```
#define GPT_PRE_SC_DIV_11 (11)
```

Pre scale Divide input clock by 4096

4.63.2.8 GPT_PRE_SC_DIV_12

```
#define GPT_PRE_SC_DIV_12 (12)
```

Pre scale Divide input clock by 8192

4.63.2.9 GPT_PRE_SC_DIV_2

```
#define GPT_PRE_SC_DIV_2 (2)
```

Pre scale Divide input clock by 8

4.63.2.10 GPT_PRE_SC_DIV_3

```
#define GPT_PRE_SC_DIV_3 (3)
```

Pre scale Divide input clock by 16

4.63.2.11 GPT_PRE_SC_DIV_4

```
#define GPT_PRE_SC_DIV_4 (4)
```

Pre scale Divide input clock by 32

4.63.2.12 GPT_PRE_SC_DIV_5

```
#define GPT_PRE_SC_DIV_5 (5)
```

Pre scale Divide input clock by 64

4.63.2.13 GPT_PRE_SC_DIV_6

```
#define GPT_PRE_SC_DIV_6 (6)
```

Pre scale Divide input clock by 128

4.63.2.14 GPT_PRE_SC_DIV_7

```
#define GPT_PRE_SC_DIV_7 (7)
```

Pre scale Divide input clock by 256

4.63.2.15 GPT_PRE_SC_DIV_8

```
#define GPT_PRE_SC_DIV_8 (8)
```

Pre scale Divide input clock by 512

4.63.2.16 GPT_PRE_SC_DIV_9

```
#define GPT_PRE_SC_DIV_9 (9)
```

Pre scale Divide input clock by 1024

4.63.2.17 HIGH_WORD_MASK

```
#define HIGH_WORD_MASK (0xFFFF0000)
```

Mask for upper word of a 32bit value

4.63.2.18 LOW_WORD_MASK

```
#define LOW_WORD_MASK (0x0000FFFF)
```

Mask for lower word of a 32bit value

4.63.2.19 TIMER_INSTANCE_COUNT

```
#define TIMER_INSTANCE_COUNT (3)
```

Timer instace count

4.63.2.20 WORD_LENGTH

```
#define WORD_LENGTH (16)
```

Length of a word

4.63.3 Variable Documentation

4.63.3.1 Timer

```
TimerClass Timer
```

Timers class instance extern which can be used by application programs to access Timers DSP APIs

Class identifier declaration

4.64 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB/examples/interrupt/interrupt.ino File Reference

4.65 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.cpp File Reference

USB implementation.

```
#include "USB.h"
```

Functions

- int [startTransferCallback](#) (void *vpContext, void *vpeps)
Start transfer call back function.
- int [completeTransferCallback](#) (void *vpContext, void *vpeps)
Complete transfer call back function.

Variables

- CSL_UsbHostPktDescr [hpdtx](#)
- CSL_UsbHostPktDescr [hpdrx](#)
- unsigned long [linking_ram0](#) [USB_LRAM_SIZE]
- unsigned short [usbDataBufferTxRx1](#) [USB_DATA_SIZE]
- unsigned short [usbDataBufferTxRx2](#) [USB_DATA_SIZE]
- unsigned short [usbDataBufferTxRx3](#) [USB_DATA_SIZE]
- unsigned short [usbDataBufferTxRx4](#) [USB_DATA_SIZE]
- USBClass [USB](#)
- CSL_UsbDevHandle [hUsbDev](#) = NULL
- USB_Config [gUsbConfig](#)
- bool [sentLongEp0Pkt](#)

4.65.1 Detailed Description

USB implementation.

4.65.2 Function Documentation

4.65.2.1 completeTransferCallback()

```
int completeTransferCallback (  
    void * vpContext,  
    void * vpeps )
```

Complete transfer call back function.

Parameters

<i>vpContext</i>	- USB context structure
<i>vpeps</i>	- End point status structure pointer

Returns

CSL_Status

4.65.2.2 startTransferCallback()

```
int startTransferCallback (  
    void * vpContext,  
    void * vpeps )
```

Start transfer call back function.

Parameters

<i>vpContext</i>	- USB context structure
<i>vpeps</i>	- End point status structure pointer

Returns

CSL_Status

4.65.3 Variable Documentation**4.65.3.1 gUsbConfig**

`USB_Config` gUsbConfig

USB configuration structure

4.65.3.2 hpdrx

CSL_UsbHostPktDescr hpdrx

USB host packet descriptor for reading

4.65.3.3 hpdtx

CSL_UsbHostPktDescr hpdtx

USB host packet descriptor for writing

4.65.3.4 hUsbDev

CSL_UsbDevHandle hUsbDev = NULL

USB handle

4.65.3.5 linking_ram0

unsigned long linking_ram0[[USB_LRAM_SIZE](#)]

Linking RAM buffer

4.65.3.6 sentLongEp0Pkt

bool sentLongEp0Pkt

Flag to indicate if a long data packet is being sent

4.65.3.7 USB

`USBClass` USB

USB class object

4.65.3.8 usbDataBufferTxRx1

```
unsigned short usbDataBufferTxRx1[USB_DATA_SIZE]
```

USB data buffer 1

4.65.3.9 usbDataBufferTxRx2

```
unsigned short usbDataBufferTxRx2[USB_DATA_SIZE]
```

USB data buffer 2

4.65.3.10 usbDataBufferTxRx3

```
unsigned short usbDataBufferTxRx3[USB_DATA_SIZE]
```

USB data buffer 3

4.65.3.11 usbDataBufferTxRx4

```
unsigned short usbDataBufferTxRx4[USB_DATA_SIZE]
```

USB data buffer 4

4.66 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/USB/USB.h File Reference

USB library header file.

```
#include <stdio.h>
#include <string.h>
#include "tistdtypes.h"
#include "csl_usb.h"
#include "csl_usbAux.h"
#include "csl_pll.h"
#include "core.h"
```

Classes

- class [USB_Config](#)
USB configuration Class.
- class [Pipe_Config](#)
Pipe configuration Class.
- class [USBClass](#)
USB Class.

Macros

- #define `USB_HS_MAX_PACKET_SIZE` (64)
- #define `USB_DATA_SIZE` (`USB_HS_MAX_PACKET_SIZE / 2`)
- #define `USB_LRAM_SIZE` (256)
- #define `USB_MAX_CURRENT` (50)
- #define `USB_WAKEUP_DELAY` (10)
- #define `USB_OUT` (1)
- #define `USB_IN` (2)
- #define `USB_PIPE_COUNT` (`CSL_USB_ENDPOINT_COUNT + 2`)
- #define `USB_STRDESC_COUNT` (4)
- #define `USB_EP0` (`CSL_USB_EP0`)
- #define `USB_EP1` (`CSL_USB_EP1`)
- #define `USB_EP2` (`CSL_USB_EP2`)
- #define `USB_EP3` (`CSL_USB_EP3`)
- #define `USB_EP4` (`CSL_USB_EP4`)
- #define `USB_OUT_EP0` (`CSL_USB_OUT_EP0`)
- #define `USB_OUT_EP1` (`CSL_USB_OUT_EP1`)
- #define `USB_OUT_EP2` (`CSL_USB_OUT_EP2`)
- #define `USB_OUT_EP3` (`CSL_USB_OUT_EP3`)
- #define `USB_OUT_EP4` (`CSL_USB_OUT_EP4`)
- #define `USB_IN_EP0` (`CSL_USB_IN_EP0`)
- #define `USB_IN_EP1` (`CSL_USB_IN_EP1`)
- #define `USB_IN_EP2` (`CSL_USB_IN_EP2`)
- #define `USB_IN_EP3` (`CSL_USB_IN_EP3`)
- #define `USB_IN_EP4` (`CSL_USB_IN_EP4`)

Typedefs

- typedef void(* `USB_APP_CALLBACK`) (Uint16 flag)
- typedef int(* `USB_APP_INT_CALLBACK`) (void *, unsigned short, unsigned short)
- typedef void * `USB_pipeHandle`
- typedef enum `USB_opMode` `USB_opMode`
Enum to indicate operation mode.
- typedef enum `USB_xferType` `USB_xferType`
Enum to indicate the transfer type.

Enumerations

- enum `USB_opMode` { `USB_OPMODE_POLLED` = 0, `USB_OPMODE_DMA` }
Enum to indicate operation mode.
- enum `USB_xferType` { `USB_CTRL` = 0, `USB_BULK` }
Enum to indicate the transfer type.

Variables

- `USBClass` `USB`

4.66.1 Detailed Description

USB library header file.

4.66.2 Macro Definition Documentation

4.66.2.1 USB_DATA_SIZE

```
#define USB_DATA_SIZE (USB_HS_MAX_PACKET_SIZE / 2)
```

Maximum USB data size

4.66.2.2 USB_EP0

```
#define USB_EP0 (CSL_USB_EP0)
```

Macro to indicate Endpoint 0

4.66.2.3 USB_EP1

```
#define USB_EP1 (CSL_USB_EP1)
```

Macro to indicate Endpoint 1

4.66.2.4 USB_EP2

```
#define USB_EP2 (CSL_USB_EP2)
```

Macro to indicate Endpoint 2

4.66.2.5 USB_EP3

```
#define USB_EP3 (CSL_USB_EP3)
```

Macro to indicate Endpoint 3

4.66.2.6 USB_EP4

```
#define USB_EP4 (CSL_USB_EP4)
```

Macro to indicate Endpoint 4

4.66.2.7 USB_HS_MAX_PACKET_SIZE

```
#define USB_HS_MAX_PACKET_SIZE (64)
```

USB highspeed maximum packet size

4.66.2.8 USB_IN

```
#define USB_IN (2)
```

USB IN endpoint number

4.66.2.9 USB_IN_EP0

```
#define USB_IN_EP0 (CSL_USB_IN_EP0)
```

Macro to indicate IN Endpoint 0

4.66.2.10 USB_IN_EP1

```
#define USB_IN_EP1 (CSL_USB_IN_EP1)
```

Macro to indicate IN Endpoint 1

4.66.2.11 USB_IN_EP2

```
#define USB_IN_EP2 (CSL_USB_IN_EP2)
```

Macro to indicate IN Endpoint 2

4.66.2.12 USB_IN_EP3

```
#define USB_IN_EP3 (CSL_USB_IN_EP3)
```

Macro to indicate IN Endpoint 3

4.66.2.13 USB_IN_EP4

```
#define USB_IN_EP4 (CSL_USB_IN_EP4)
```

Macro to indicate IN Endpoint 4

4.66.2.14 USB_LRAM_SIZE

```
#define USB_LRAM_SIZE (256)
```

Linking RAM size

4.66.2.15 USB_MAX_CURRENT

```
#define USB_MAX_CURRENT (50)
```

USB maximum current

4.66.2.16 USB_OUT

```
#define USB_OUT (1)
```

USB OUT endpoint number

4.66.2.17 USB_OUT_EP0

```
#define USB_OUT_EP0 (CSL_USB_OUT_EP0)
```

Macro to indicate OUT Endpoint 0

4.66.2.18 USB_OUT_EP1

```
#define USB_OUT_EP1 (CSL_USB_OUT_EP1)
```

Macro to indicate OUT Endpoint 1

4.66.2.19 USB_OUT_EP2

```
#define USB_OUT_EP2 (CSL_USB_OUT_EP2)
```

Macro to indicate OUT Endpoint 2

4.66.2.20 USB_OUT_EP3

```
#define USB_OUT_EP3 (CSL_USB_OUT_EP3)
```

Macro to indicate OUT Endpoint 3

4.66.2.21 USB_OUT_EP4

```
#define USB_OUT_EP4 (CSL_USB_OUT_EP4)
```

Macro to indicate OUT Endpoint 4

4.66.2.22 USB_PIPE_COUNT

```
#define USB_PIPE_COUNT (CSL_USB_ENDPOINT_COUNT + 2)
```

Number of pipes supported

4.66.2.23 USB_STRDESC_COUNT

```
#define USB_STRDESC_COUNT (4)
```

String descriptor count

4.66.2.24 USB_WAKEUP_DELAY

```
#define USB_WAKEUP_DELAY (10)
```

USB wakeup delay

4.66.3 Typedef Documentation

4.66.3.1 USB_APP_CALLBACK

```
typedef void(* USB_APP_CALLBACK) (Uint16 flag)
```

USB application call back pointer

4.66.3.2 USB_APP_INT_CALLBACK

```
typedef int(* USB_APP_INT_CALLBACK) (void *, unsigned short, unsigned short)
```

USB application call back pointer for tx and rx interrupts

4.66.3.3 USB_opMode

```
typedef enum USB_opMode USB_opMode
```

Enum to indicate operation mode.

4.66.3.4 USB_pipeHandle

```
typedef void* USB_pipeHandle
```

USB pipe handle

4.66.3.5 USB_xferType

```
typedef enum USB_xferType USB_xferType
```

Enum to indicate the transfer type.

4.66.4 Enumeration Type Documentation

4.66.4.1 USB_opMode

```
enum USB_opMode
```

Enum to indicate operation mode.

Enumerator

USB_OPMODE_POLLED	
USB_OPMODE_DMA	

4.66.4.2 USB_xferType

```
enum USB_xferType
```

Enum to indicate the transfer type.

Enumerator

USB_CTRL	
USB_BULK	

4.66.5 Variable Documentation

4.66.5.1 USB

```
USBClass USB
```

USB class instance extern which can used by application programs to access USB DSP APIs

USB class object

4.67 /Users/fmujica/Downloads/C5000-selected/libraries_documented/libraries/Wire/examples/Wire/Wire.ino File Reference

