

My Project

Generated by Doxygen 1.8.6

Wed Jul 15 2015 11:53:53

Contents

1	README	1
2	Todo List	3
3	Module Index	5
3.1	Modules	5
4	Namespace Index	7
4.1	Namespace List	7
5	Hierarchical Index	9
5.1	Class Hierarchy	9
6	Class Index	11
6.1	Class List	11
7	File Index	13
7.1	File List	13
8	Module Documentation	17
8.1	hidapi API	17
8.1.1	Detailed Description	18
8.1.2	Function Documentation	18
8.1.2.1	hid_close	18
8.1.2.2	hid_enumerate	18
8.1.2.3	hid_error	18
8.1.2.4	hid_exit	19
8.1.2.5	hid_free_enumeration	19
8.1.2.6	hid_get_feature_report	19
8.1.2.7	hid_get_indexed_string	19
8.1.2.8	hid_get_manufacturer_string	20
8.1.2.9	hid_get_product_string	20
8.1.2.10	hid_get_serial_number_string	20
8.1.2.11	hid_init	21

8.1.2.12	hid_open	21
8.1.2.13	hid_open	21
8.1.2.14	hid_open_path	21
8.1.2.15	hid_read	22
8.1.2.16	hid_read_timeout	22
8.1.2.17	hid_send_feature_report	22
8.1.2.18	hid_set_nonblocking	23
8.1.2.19	hid_write	23
9	Namespace Documentation	25
9.1	DTU Namespace Reference	25
9.1.1	Detailed Description	25
9.2	JAMA Namespace Reference	25
9.3	TNT Namespace Reference	26
9.3.1	Typedef Documentation	29
9.3.1.1	Subscript	29
9.3.2	Function Documentation	29
9.3.2.1	dot_prod	29
9.3.2.2	hypot	29
9.3.2.3	matmult	29
9.3.2.4	matmult	29
9.3.2.5	matmult	29
9.3.2.6	matmult	29
9.3.2.7	mult_element	29
9.3.2.8	operator*	29
9.3.2.9	operator*	30
9.3.2.10	operator*	30
9.3.2.11	operator*	30
9.3.2.12	operator*	30
9.3.2.13	operator*	30
9.3.2.14	operator*	30
9.3.2.15	operator*	30
9.3.2.16	operator*	30
9.3.2.17	operator*=operator*+=	30
9.3.2.18	operator*=operator*+=	30
9.3.2.19	operator*=operator*+=	30
9.3.2.20	operator*=operator*+=	30
9.3.2.21	operator*=operator*+=	30
9.3.2.22	operator*=operator*+=	30
9.3.2.23	operator+	30

9.3.2.24	operator+	30
9.3.2.25	operator+	30
9.3.2.26	operator+	30
9.3.2.27	operator+	30
9.3.2.28	operator+	30
9.3.2.29	operator+	30
9.3.2.30	operator+	30
9.3.2.31	operator+=	30
9.3.2.32	operator+=	30
9.3.2.33	operator+=	31
9.3.2.34	operator+=	31
9.3.2.35	operator+=	31
9.3.2.36	operator+=	31
9.3.2.37	operator-	31
9.3.2.38	operator-	31
9.3.2.39	operator-	31
9.3.2.40	operator-	31
9.3.2.41	operator-	31
9.3.2.42	operator-	31
9.3.2.43	operator-	31
9.3.2.44	operator-	31
9.3.2.45	operator-=	31
9.3.2.46	operator-=	31
9.3.2.47	operator-=	31
9.3.2.48	operator-=	31
9.3.2.49	operator-=	31
9.3.2.50	operator-=	31
9.3.2.51	operator/	31
9.3.2.52	operator/	31
9.3.2.53	operator/	31
9.3.2.54	operator/	31
9.3.2.55	operator/	31
9.3.2.56	operator/	32
9.3.2.57	operator/=	32
9.3.2.58	operator/=	32
9.3.2.59	operator/=	32
9.3.2.60	operator/=	32
9.3.2.61	operator/=	32
9.3.2.62	operator/=	32
9.3.2.63	operator<<	32

9.3.2.64	operator<<	32
9.3.2.65	operator<<	32
9.3.2.66	operator<<	32
9.3.2.67	operator<<	32
9.3.2.68	operator<<	32
9.3.2.69	operator<<	32
9.3.2.70	operator<<	32
9.3.2.71	operator>>	32
9.3.2.72	operator>>	32
9.3.2.73	operator>>	32
9.3.2.74	operator>>	32
9.3.2.75	operator>>	32
9.3.2.76	operator>>	32
9.3.2.77	operator>>	33
9.3.2.78	operator>>	33
9.3.2.79	transpose	33
10	Class Documentation	35
10.1	TNT::Array1D< T > Class Template Reference	35
10.1.1	Member Typedef Documentation	35
10.1.1.1	value_type	35
10.1.2	Constructor & Destructor Documentation	35
10.1.2.1	Array1D	35
10.1.2.2	Array1D	36
10.1.2.3	Array1D	36
10.1.2.4	Array1D	36
10.1.2.5	Array1D	36
10.1.2.6	~Array1D	36
10.1.3	Member Function Documentation	36
10.1.3.1	copy	36
10.1.3.2	dim	36
10.1.3.3	dim1	36
10.1.3.4	inject	36
10.1.3.5	operator const T *	36
10.1.3.6	operator T *	36
10.1.3.7	operator=	36
10.1.3.8	operator=	36
10.1.3.9	operator[]	36
10.1.3.10	operator[]	36
10.1.3.11	ref	36

10.1.3.12 ref_count	36
10.1.3.13 subarray	36
10.2 TNT::Array2D< T > Class Template Reference	36
10.2.1 Member Typedef Documentation	37
10.2.1.1 value_type	37
10.2.2 Constructor & Destructor Documentation	37
10.2.2.1 Array2D	37
10.2.2.2 Array2D	37
10.2.2.3 Array2D	37
10.2.2.4 Array2D	37
10.2.2.5 Array2D	37
10.2.2.6 ~Array2D	37
10.2.3 Member Function Documentation	38
10.2.3.1 copy	38
10.2.3.2 dim1	38
10.2.3.3 dim2	38
10.2.3.4 inject	38
10.2.3.5 operator const T **	38
10.2.3.6 operator T **	38
10.2.3.7 operator=	38
10.2.3.8 operator=	38
10.2.3.9 operator[]	38
10.2.3.10 operator[]	38
10.2.3.11 ref	38
10.2.3.12 ref_count	38
10.2.3.13 ref_count_data	38
10.2.3.14 ref_count_dim1	38
10.2.3.15 subarray	38
10.2.4 Member Data Documentation	38
10.2.4.1 data_	38
10.2.4.2 m_	38
10.2.4.3 n_	38
10.2.4.4 v_	38
10.3 TNT::Array3D< T > Class Template Reference	38
10.3.1 Member Typedef Documentation	39
10.3.1.1 value_type	39
10.3.2 Constructor & Destructor Documentation	39
10.3.2.1 Array3D	39
10.3.2.2 Array3D	39
10.3.2.3 Array3D	39

10.3.2.4	Array3D	39
10.3.2.5	Array3D	39
10.3.2.6	~Array3D	39
10.3.3	Member Function Documentation	39
10.3.3.1	copy	39
10.3.3.2	dim1	39
10.3.3.3	dim2	39
10.3.3.4	dim3	39
10.3.3.5	inject	40
10.3.3.6	operator const T ***	40
10.3.3.7	operator T ***	40
10.3.3.8	operator=	40
10.3.3.9	operator=	40
10.3.3.10	operator[]	40
10.3.3.11	operator[]	40
10.3.3.12	ref	40
10.3.3.13	ref_count	40
10.3.3.14	subarray	40
10.4	JAMA::Cholesky< Real > Class Template Reference	40
10.4.1	Detailed Description	40
10.4.2	Constructor & Destructor Documentation	41
10.4.2.1	Cholesky	41
10.4.2.2	Cholesky	41
10.4.3	Member Function Documentation	41
10.4.3.1	getL	41
10.4.3.2	is_spd	41
10.4.3.3	solve	41
10.4.3.4	solve	41
10.5	Class Class Reference	42
10.5.1	Detailed Description	42
10.6	CRijndael Class Reference	43
10.6.1	Member Enumeration Documentation	43
10.6.1.1	anonymous enum	43
10.6.2	Constructor & Destructor Documentation	44
10.6.2.1	CRijndael	44
10.6.2.2	~CRijndael	44
10.6.3	Member Function Documentation	44
10.6.3.1	Decrypt	44
10.6.3.2	DecryptBlock	44
10.6.3.3	Encrypt	44

10.6.3.4	EncryptBlock	44
10.6.3.5	GetBlockSize	44
10.6.3.6	GetKeyLength	44
10.6.3.7	GetRounds	44
10.6.3.8	MakeKey	44
10.6.3.9	ResetChain	44
10.6.4	Member Data Documentation	44
10.6.4.1	sm_chain0	44
10.7	DTU::DtuArray2D< T > Class Template Reference	44
10.7.1	Member Typedef Documentation	45
10.7.1.1	value_type	45
10.7.2	Constructor & Destructor Documentation	45
10.7.2.1	DtuArray2D	45
10.7.2.2	DtuArray2D	45
10.7.2.3	DtuArray2D	45
10.7.2.4	DtuArray2D	45
10.7.2.5	DtuArray2D	45
10.7.3	Member Function Documentation	45
10.7.3.1	add	45
10.7.3.2	dim1	46
10.7.3.3	dim2	46
10.7.3.4	getSVD	46
10.7.3.5	multiply	46
10.7.3.6	multiply	46
10.7.3.7	multiply	46
10.7.3.8	multiplyR	46
10.7.3.9	operator=	46
10.7.3.10	pinv	46
10.7.3.11	print	46
10.7.3.12	subtract	46
10.7.3.13	toIdentityMatrix	46
10.7.3.14	toTntArray2D	46
10.7.3.15	trace	46
10.7.3.16	transpose	46
10.7.3.17	transpose	46
10.7.3.18	transpose_insitu	46
10.8	JAMA::Eigenvalue< Real > Class Template Reference	46
10.8.1	Detailed Description	47
10.8.2	Constructor & Destructor Documentation	47
10.8.2.1	Eigenvalue	47

10.8.3 Member Function Documentation	48
10.8.3.1 getD	48
10.8.3.2 getImagEigenvalues	48
10.8.3.3 getRealEigenvalues	48
10.8.3.4 getV	49
10.9 FFTReal Class Reference	49
10.9.1 Member Typedef Documentation	49
10.9.1.1 flt_t	49
10.9.2 Constructor & Destructor Documentation	49
10.9.2.1 FFTReal	49
10.9.2.2 ~FFTReal	49
10.9.3 Member Function Documentation	49
10.9.3.1 do_fft	49
10.9.3.2 do_ifft	49
10.9.3.3 rescale	49
10.10 TNT::Fortran_Array1D< T > Class Template Reference	50
10.10.1 Member Typedef Documentation	50
10.10.1.1 value_type	50
10.10.2 Constructor & Destructor Documentation	50
10.10.2.1 Fortran_Array1D	50
10.10.2.2 Fortran_Array1D	50
10.10.2.3 Fortran_Array1D	50
10.10.2.4 Fortran_Array1D	50
10.10.2.5 Fortran_Array1D	50
10.10.2.6 ~Fortran_Array1D	50
10.10.3 Member Function Documentation	50
10.10.3.1 copy	50
10.10.3.2 dim	50
10.10.3.3 dim1	51
10.10.3.4 inject	51
10.10.3.5 operator()	51
10.10.3.6 operator()	51
10.10.3.7 operator=	51
10.10.3.8 operator=	51
10.10.3.9 ref	51
10.10.3.10 ref_count	51
10.10.3.11 subarray	51
10.11 TNT::Fortran_Array2D< T > Class Template Reference	51
10.11.1 Member Typedef Documentation	52
10.11.1.1 value_type	52

10.11.2 Constructor & Destructor Documentation	52
10.11.2.1 Fortran_Array2D	52
10.11.2.2 Fortran_Array2D	52
10.11.2.3 Fortran_Array2D	52
10.11.2.4 Fortran_Array2D	52
10.11.2.5 Fortran_Array2D	52
10.11.2.6 ~Fortran_Array2D	52
10.11.3 Member Function Documentation	52
10.11.3.1 copy	52
10.11.3.2 dim1	52
10.11.3.3 dim2	52
10.11.3.4 inject	52
10.11.3.5 operator()	52
10.11.3.6 operator()	52
10.11.3.7 operator=	52
10.11.3.8 operator=	52
10.11.3.9 ref	52
10.11.3.10ref_count	52
10.12TNT::Fortran_Array3D< T > Class Template Reference	52
10.12.1 Member Typedef Documentation	53
10.12.1.1 value_type	53
10.12.2 Constructor & Destructor Documentation	53
10.12.2.1 Fortran_Array3D	53
10.12.2.2 Fortran_Array3D	53
10.12.2.3 Fortran_Array3D	53
10.12.2.4 Fortran_Array3D	53
10.12.2.5 Fortran_Array3D	53
10.12.2.6 ~Fortran_Array3D	53
10.12.3 Member Function Documentation	53
10.12.3.1 copy	53
10.12.3.2 dim1	53
10.12.3.3 dim2	53
10.12.3.4 dim3	53
10.12.3.5 inject	54
10.12.3.6 operator()	54
10.12.3.7 operator()	54
10.12.3.8 operator=	54
10.12.3.9 operator=	54
10.12.3.10ref	54
10.12.3.11ref_count	54

10.13hid_device_ Struct Reference	54
10.13.1 Member Data Documentation	54
10.13.1.1 barrier	54
10.13.1.2 blocking	54
10.13.1.3 condition	54
10.13.1.4 device_handle	55
10.13.1.5 device_handle	55
10.13.1.6 disconnected	55
10.13.1.7 input_report_buf	55
10.13.1.8 input_reports	55
10.13.1.9 max_input_report_len	55
10.13.1.10mutex	55
10.13.1.11next	55
10.13.1.12run_loop	55
10.13.1.13run_loop_mode	55
10.13.1.14shutdown_barrier	55
10.13.1.15shutdown_thread	55
10.13.1.16source	55
10.13.1.17thread	55
10.13.1.18uses_numbered_reports	55
10.14hid_device_info Struct Reference	55
10.14.1 Detailed Description	56
10.14.2 Member Data Documentation	56
10.14.2.1 interface_number	56
10.14.2.2 manufacturer_string	56
10.14.2.3 next	56
10.14.2.4 path	56
10.14.2.5 product_id	56
10.14.2.6 product_string	56
10.14.2.7 release_number	56
10.14.2.8 serial_number	56
10.14.2.9 usage	56
10.14.2.10usage_page	56
10.14.2.11vendor_id	57
10.15TNT::i_refvec< T > Class Template Reference	57
10.15.1 Constructor & Destructor Documentation	57
10.15.1.1 i_refvec	57
10.15.1.2 i_refvec	57
10.15.1.3 i_refvec	57
10.15.1.4 i_refvec	57

10.15.1.5 <code>~i_refvec</code>	57
10.15.2 Member Function Documentation	57
10.15.2.1 <code>begin</code>	57
10.15.2.2 <code>begin</code>	57
10.15.2.3 <code>copy_</code>	57
10.15.2.4 <code>destroy</code>	58
10.15.2.5 <code>is_null</code>	58
10.15.2.6 <code>operator=</code>	58
10.15.2.7 <code>operator[]</code>	58
10.15.2.8 <code>operator[]</code>	58
10.15.2.9 <code>ref_count</code>	58
10.15.2.10 <code>set_</code>	58
10.16 <code>input_report</code> Struct Reference	58
10.16.1 Member Data Documentation	58
10.16.1.1 <code>data</code>	58
10.16.1.2 <code>len</code>	58
10.16.1.3 <code>next</code>	58
10.17 <code>JAMA::LU< Real ></code> Class Template Reference	58
10.17.1 Detailed Description	59
10.17.2 Constructor & Destructor Documentation	59
10.17.2.1 <code>LU</code>	59
10.17.3 Member Function Documentation	59
10.17.3.1 <code>det</code>	59
10.17.3.2 <code>getL</code>	59
10.17.3.3 <code>getPivot</code>	59
10.17.3.4 <code>getU</code>	60
10.17.3.5 <code>isNonsingular</code>	60
10.17.3.6 <code>solve</code>	60
10.17.3.7 <code>solve</code>	60
10.18 <code>TNT::Matrix< T ></code> Class Template Reference	60
10.18.1 Member Typedef Documentation	62
10.18.1.1 <code>const_iterator</code>	62
10.18.1.2 <code>const_reference</code>	62
10.18.1.3 <code>element_type</code>	62
10.18.1.4 <code>iterator</code>	62
10.18.1.5 <code>pointer</code>	62
10.18.1.6 <code>reference</code>	62
10.18.1.7 <code>size_type</code>	62
10.18.1.8 <code>value_type</code>	62
10.18.2 Constructor & Destructor Documentation	62

10.18.2.1 Matrix	62
10.18.2.2 Matrix	62
10.18.2.3 Matrix	62
10.18.2.4 Matrix	62
10.18.2.5 Matrix	62
10.18.2.6 ~Matrix	62
10.18.3 Member Function Documentation	62
10.18.3.1 copy	62
10.18.3.2 destroy	62
10.18.3.3 dim	62
10.18.3.4 initialize	62
10.18.3.5 lbound	62
10.18.3.6 newsize	62
10.18.3.7 num_cols	62
10.18.3.8 num_rows	62
10.18.3.9 operator T **	62
10.18.3.10operator T **	62
10.18.3.11operator()	63
10.18.3.12operator()	63
10.18.3.13operator()	63
10.18.3.14operator()	63
10.18.3.15operator=	63
10.18.3.16operator=	63
10.18.3.17operator[]	63
10.18.3.18operator[]	63
10.18.3.19set	63
10.18.3.20size	63
10.18.4 Member Data Documentation	63
10.18.4.1 m_	63
10.18.4.2 mn_	63
10.18.4.3 n_	63
10.18.4.4 row_	63
10.18.4.5 rowm1_	63
10.18.4.6 v_	63
10.18.4.7 vm1_	63
10.19pthread_barrier Struct Reference	63
10.19.1 Member Data Documentation	63
10.19.1.1 cond	64
10.19.1.2 count	64
10.19.1.3 mutex	64

10.19.1.4 trip_count	64
10.20QmlApplicationViewer Class Reference	64
10.20.1 Member Enumeration Documentation	64
10.20.1.1 ScreenOrientation	64
10.20.2 Constructor & Destructor Documentation	65
10.20.2.1 QmlApplicationViewer	65
10.20.2.2 ~QmlApplicationViewer	65
10.20.3 Member Function Documentation	65
10.20.3.1 addImportPath	65
10.20.3.2 create	65
10.20.3.3 setMainQmlFile	65
10.20.3.4 setOrientation	65
10.20.3.5 showExpanded	65
10.21QmlApplicationViewerPrivate Class Reference	65
10.21.1 Friends And Related Function Documentation	65
10.21.1.1 QmlApplicationViewer	65
10.22JAMA::QR< Real > Class Template Reference	65
10.22.1 Detailed Description	66
10.22.2 Constructor & Destructor Documentation	66
10.22.2.1 QR	66
10.22.3 Member Function Documentation	66
10.22.3.1 getHouseholder	66
10.22.3.2 getQ	66
10.22.3.3 getR	66
10.22.3.4 isFullRank	67
10.22.3.5 solve	67
10.22.3.6 solve	67
10.23Sbs2Callback Class Reference	67
10.23.1 Constructor & Destructor Documentation	69
10.23.1.1 Sbs2Callback	69
10.23.2 Member Function Documentation	69
10.23.2.1 addMessageUdpOutputHost	69
10.23.2.2 addRawDataHost	69
10.23.2.3 batteryValue	69
10.23.2.4 clearMessageUdpOutputHosts	69
10.23.2.5 cqValue	69
10.23.2.6 cqValues	69
10.23.2.7 deviceFound	69
10.23.2.8 deviceFoundSignal	69
10.23.2.9 getCurrentPacket	70

10.23.2.10	getCurrentPacketCounter	70
10.23.2.11	getData	70
10.23.2.12	getNetworkAddresses	70
10.23.2.13	getRawFilename	70
10.23.2.14	hardwareChanged	70
10.23.2.15	insertIntoMetaFile	70
10.23.2.16	networkAddresses	70
10.23.2.17	readMessage	70
10.23.2.18	removeMessageUdpOutputHost	70
10.23.2.19	removeRawDataHost	70
10.23.2.20	sendMessage	70
10.23.2.21	sendMessage	70
10.23.2.22	setHardware	70
10.23.2.23	setPacket	70
10.23.2.24	setSbs2DataHandler	70
10.23.2.25	setWindowType	70
10.23.2.26	setWindowType	70
10.23.2.27	setWindowTypeSignal	70
10.23.2.28	spectrogramUpdated	70
10.23.2.29	spectrogramUpdatedSlot	70
10.23.2.30	startRecording	70
10.23.2.31	stopRecording	70
10.23.2.32	timeTick0	70
10.23.2.33	timeTick10	70
10.23.2.34	timeTick16	70
10.23.2.35	timeTick2	70
10.23.2.36	timeTick4	71
10.23.2.37	timeTick8	71
10.23.2.38	turnChannelSpectrogramOff	71
10.23.2.39	turnChannelSpectrogramOn	71
10.23.2.40	turnFilterOff	71
10.23.2.41	turnFilterOn	71
10.23.2.42	turnOnSourceReconstructionLoreta	71
10.23.2.43	turnOnSourceReconstructioSparse	71
10.23.2.44	turnReceiveMessageOff	71
10.23.2.45	turnReceiveMessageOn	71
10.23.2.46	turnSendRawDataOff	71
10.23.2.47	turnSendRawDataOn	71
10.23.2.48	udpMessageReceived	71
10.23.3	Member Data Documentation	71

10.23.3.1	currentPacket	71
10.23.3.2	currentPacketCounter	71
10.23.3.3	devicePresent	71
10.23.3.4	isRecording	71
10.23.3.5	params	71
10.23.3.6	samplesCollected	71
10.23.3.7	sbs2DataHandler	71
10.23.3.8	sbs2Region	71
10.23.3.9	thisPacket	71
10.24	Sbs2Common Class Reference	72
10.24.1	Member Function Documentation	72
10.24.1.1	channelsNo	72
10.24.1.2	getCatalogPath	72
10.24.1.3	getChannelNames	72
10.24.1.4	getChannels	72
10.24.1.5	getCqs	72
10.24.1.6	getCqsMapping	72
10.24.1.7	getCurrentHardware	72
10.24.1.8	getRootAppPath	72
10.24.1.9	normalize	72
10.24.1.10	rawDataSize	72
10.24.1.11	samplingRate	72
10.24.1.12	setCatalogPath	72
10.24.1.13	setDefaultCatalogPath	72
10.24.1.14	setDefaultRootAppPath	72
10.24.1.15	setHardware	73
10.24.1.16	setRootAppPath	73
10.24.1.17	verticesNo	73
10.25	Sbs2DataHandler Class Reference	73
10.25.1	Constructor & Destructor Documentation	75
10.25.1.1	Sbs2DataHandler	75
10.25.1.2	~Sbs2DataHandler	75
10.25.2	Member Function Documentation	75
10.25.2.1	addMessageUdpOutputHost	75
10.25.2.2	addRawDataHost	75
10.25.2.3	clearMessageUdpOutputHosts	75
10.25.2.4	doSourceReconstruction	75
10.25.2.5	doSourceReconstructionSpectrogram	75
10.25.2.6	filter	75
10.25.2.7	getPacketZero	75

10.25.2.8	getPowerValues	75
10.25.2.9	getRawFilename	75
10.25.2.10	getSourceReconstructionMeanValues	75
10.25.2.11	getSourceReconstructionSpectrogramValues	76
10.25.2.12	insertIntoMetaFile	76
10.25.2.13	readMessage	76
10.25.2.14	record	76
10.25.2.15	removeMessageUdpOutputHost	76
10.25.2.16	removeRawDataHost	76
10.25.2.17	reset	76
10.25.2.18	sendMessage	76
10.25.2.19	sendMessage	76
10.25.2.20	sendRawData	76
10.25.2.21	setSourceReconstructionVerticesToExtract	76
10.25.2.22	setThisPacket	76
10.25.2.23	setVerticesToExtract	76
10.25.2.24	setWindowType	76
10.25.2.25	setWindowTypeSignal	76
10.25.2.26	sourceReconstructionReady	76
10.25.2.27	sourceReconstructionSpectrogramReady	76
10.25.2.28	spectrogramChannel	76
10.25.2.29	spectrogramUpdated	76
10.25.2.30	startRecording	76
10.25.2.31	stopRecording	76
10.25.2.32	turnChannelSpectrogramOff	76
10.25.2.33	turnChannelSpectrogramOn	76
10.25.2.34	turnFilterOff	76
10.25.2.35	turnFilterOn	76
10.25.2.36	turnOffSourceReconstruction	76
10.25.2.37	turnOnSourceReconstructionLoreta	77
10.25.2.38	turnOnSourceReconstructionSparse	77
10.25.2.39	turnReceiveMessageOff	77
10.25.2.40	turnReceiveMessageOn	77
10.25.2.41	turnSendRawDataOff	77
10.25.2.42	turnSendRawDataOn	77
10.25.2.43	udpMessageReceived	77
10.25.3	Member Data Documentation	77
10.25.3.1	fbandHigh	77
10.25.3.2	fbandLow	77
10.25.3.3	filterOn	77

10.25.3.4 filterOrder	77
10.25.3.5 filterResultValues	77
10.25.3.6 hardware	77
10.25.3.7 isSourceReconstructionReady	77
10.25.3.8 networkSendRawDataOn	77
10.25.3.9 packetsSeen	77
10.25.3.10 powerValues	77
10.25.3.11 readyToReconstruct	77
10.25.3.12 recording	77
10.25.3.13 samplesCollected	77
10.25.3.14 sbs2FileHandler	77
10.25.3.15 sbs2Filter	77
10.25.3.16 sbs2NetworkHandler	77
10.25.3.17 sbs2SourceReconstruction	77
10.25.3.18 sbs2Spectrogram	77
10.25.3.19 sourceReconstructionDelta	78
10.25.3.20 sourceReconstructionDeltaCollected	78
10.25.3.21 sourceReconstructionMethod	78
10.25.3.22 sourceReconstructionModelUpdateDelta	78
10.25.3.23 sourceReconstructionModelUpdateLength	78
10.25.3.24 sourceReconstructionOn	78
10.25.3.25 sourceReconstructionSamples	78
10.25.3.26 sourceReconstructionSpectrogramValues	78
10.25.3.27 sourceReconstructionValues	78
10.25.3.28 spectrogramChannelDelta	78
10.25.3.29 spectrogramChannelDeltaCollected	78
10.25.3.30 spectrogramChannelLength	78
10.25.3.31 spectrogramChannelOn	78
10.25.3.32 spectrogramChannelSamples	78
10.25.3.33 spectrogramValues	78
10.25.3.34 thisPacket	78
10.25.3.35 toFilterValues	78
10.25.3.36 toSourceReconstructionValues	78
10.25.3.37 toSpectrogramValues	78
10.26 Sbs2DataReader Class Reference	78
10.26.1 Constructor & Destructor Documentation	79
10.26.1.1 ~Sbs2DataReader	79
10.26.1.2 Sbs2DataReader	79
10.26.2 Member Function Documentation	79
10.26.2.1 aboutToQuit	80

10.26.2.2 deviceFound	80
10.26.2.3 deviceFoundSignal	80
10.26.2.4 deviceLost	80
10.26.2.5 execute	80
10.26.2.6 turnReceiveUdpDataOff	80
10.26.2.7 turnReceiveUdpDataOn	80
10.26.2.8 udpDataReceived	80
10.26.2.9 udpDataReceived	80
10.26.3 Member Data Documentation	80
10.26.3.1 bufferIndex	80
10.26.3.2 bufferSize	80
10.26.3.3 currentIndex	80
10.26.3.4 framesRead	80
10.26.3.5 lastReceiveRawDataCounter	80
10.26.3.6 readOnlyFromNetwork	80
10.26.3.7 running	80
10.26.3.8 sbs2Callback	80
10.26.3.9 sbs2NetworkHandler	80
10.26.3.10 testDummyRead	80
10.27 Sbs2Emocap28DataContainer Class Reference	80
10.27.1 Detailed Description	81
10.27.2 Constructor & Destructor Documentation	81
10.27.2.1 Sbs2Emocap28DataContainer	81
10.27.3 Member Function Documentation	81
10.27.3.1 update	81
10.27.4 Member Data Documentation	81
10.27.4.1 counter	81
10.27.4.2 data	81
10.28 Sbs2Emocap28DataReader Class Reference	81
10.28.1 Constructor & Destructor Documentation	82
10.28.1.1 ~Sbs2Emocap28DataReader	82
10.28.2 Member Function Documentation	82
10.28.2.1 aboutToQuit	82
10.28.2.2 alignedSignal	82
10.28.2.3 amp1FoundSignal	83
10.28.2.4 amp2FoundSignal	83
10.28.2.5 deviceFound	83
10.28.2.6 deviceLost	83
10.28.2.7 inMappingSignal	83
10.28.2.8 mappingFailed	83

10.28.2.9 mappingSuccessful	83
10.28.2.10 New	83
10.28.2.11 readyForData	83
10.28.2.12 turnReceiveUdpDataOff	83
10.28.2.13 turnReceiveUdpDataOn	83
10.28.2.14 udpDataReceived	83
10.28.2.15 udpDataReceived	83
10.29 Sbs2Emocap28Mounter Class Reference	83
10.29.1 Detailed Description	84
10.29.2 Constructor & Destructor Documentation	84
10.29.2.1 ~Sbs2Emocap28Mounter	84
10.29.3 Member Function Documentation	84
10.29.3.1 invalidate	84
10.29.3.2 New	84
10.29.3.3 start	84
10.29.3.4 stop	84
10.30 Sbs2Emocap28Packet Class Reference	85
10.30.1 Detailed Description	85
10.30.2 Constructor & Destructor Documentation	85
10.30.2.1 Sbs2Emocap28Packet	85
10.30.3 Member Function Documentation	85
10.30.3.1 getCounter	86
10.30.3.2 getValue	86
10.30.3.3 update	86
10.30.3.4 update	86
10.31 Sbs2EmocapDataReader Class Reference	86
10.31.1 Detailed Description	87
10.31.2 Constructor & Destructor Documentation	87
10.31.2.1 ~Sbs2EmocapDataReader	87
10.31.3 Member Function Documentation	87
10.31.3.1 aboutToQuit	87
10.31.3.2 deviceFound	87
10.31.3.3 deviceLost	87
10.31.3.4 New	87
10.31.3.5 turnReceiveUdpDataOff	87
10.31.3.6 turnReceiveUdpDataOn	87
10.31.3.7 udpDataReceived	87
10.31.3.8 udpDataReceived	87
10.32 Sbs2EmocapMounter Class Reference	88
10.32.1 Detailed Description	88

10.32.2 Constructor & Destructor Documentation	89
10.32.2.1 ~Sbs2EmocapMounter	89
10.32.3 Member Function Documentation	89
10.32.3.1 invalidate	89
10.32.3.2 New	89
10.32.3.3 start	89
10.32.3.4 stop	89
10.33Sbs2EmocapPacket Class Reference	89
10.33.1 Detailed Description	89
10.33.2 Constructor & Destructor Documentation	90
10.33.2.1 Sbs2EmocapPacket	90
10.33.3 Member Function Documentation	90
10.33.3.1 update	90
10.34Sbs2EmotivDataReader Class Reference	90
10.34.1 Detailed Description	91
10.34.2 Constructor & Destructor Documentation	91
10.34.2.1 ~Sbs2EmotivDataReader	91
10.34.3 Member Function Documentation	91
10.34.3.1 aboutToQuit	91
10.34.3.2 deviceFound	91
10.34.3.3 deviceLost	91
10.34.3.4 New	91
10.34.3.5 turnReceiveUdpDataOff	92
10.34.3.6 turnReceiveUdpDataOn	92
10.34.3.7 udpDataReceived	92
10.34.3.8 udpDataReceived	92
10.35Sbs2EmotivDecryptor Class Reference	92
10.35.1 Detailed Description	92
10.35.2 Constructor & Destructor Documentation	93
10.35.2.1 Sbs2EmotivDecryptor	93
10.35.3 Member Function Documentation	93
10.35.3.1 decrypt	93
10.35.3.2 setSerialNumber	93
10.35.3.3 setSerialNumber	93
10.36Sbs2EmotivMounter Class Reference	93
10.36.1 Detailed Description	93
10.36.2 Constructor & Destructor Documentation	94
10.36.2.1 ~Sbs2EmotivMounter	94
10.36.3 Member Function Documentation	94
10.36.3.1 invalidate	94

10.36.3.2 New	94
10.36.3.3 start	94
10.36.3.4 stop	94
10.37Sbs2EmotivPacket Class Reference	94
10.37.1 Detailed Description	95
10.37.2 Constructor & Destructor Documentation	95
10.37.2.1 Sbs2EmotivPacket	95
10.37.3 Member Function Documentation	95
10.37.3.1 update	95
10.38Sbs2FakeDataReader Class Reference	96
10.38.1 Constructor & Destructor Documentation	96
10.38.1.1 ~Sbs2FakeDataReader	96
10.38.2 Member Function Documentation	96
10.38.2.1 New	96
10.38.2.2 setFilename	96
10.38.2.3 start	96
10.38.2.4 stop	96
10.39Sbs2FakePacket Class Reference	97
10.39.1 Constructor & Destructor Documentation	97
10.39.1.1 Sbs2FakePacket	97
10.39.2 Member Function Documentation	97
10.39.2.1 update	97
10.40Sbs2FileHandler Class Reference	97
10.40.1 Detailed Description	98
10.40.2 Constructor & Destructor Documentation	98
10.40.2.1 ~Sbs2FileHandler	98
10.40.3 Member Function Documentation	98
10.40.3.1 close	98
10.40.3.2 createMetaFile	98
10.40.3.3 dumpRawData	98
10.40.3.4 getPacketZero	98
10.40.3.5 getRawFilename	98
10.40.3.6 insertIntoMetaFile	98
10.40.3.7 New	98
10.41Sbs2Filter Class Reference	99
10.41.1 Constructor & Destructor Documentation	99
10.41.1.1 ~Sbs2Filter	99
10.41.2 Member Function Documentation	99
10.41.2.1 doFilter	99
10.41.2.2 loadFilter	99

10.41.2.3 New	99
10.41.2.4 updateFilter	99
10.42Sbs2HardwareMounter Class Reference	100
10.42.1 Detailed Description	101
10.42.2 Constructor & Destructor Documentation	101
10.42.2.1 ~Sbs2HardwareMounter	101
10.42.2.2 Sbs2HardwareMounter	101
10.42.3 Member Function Documentation	101
10.42.3.1 deviceFound	101
10.42.3.2 deviceLost	101
10.42.3.3 getIdentifier	101
10.42.3.4 init	101
10.42.3.5 invalidate	101
10.42.3.6 mount	101
10.42.3.7 mySleep	101
10.42.3.8 readHardwareParameters	101
10.42.3.9 start	101
10.42.3.10stop	101
10.42.3.11umount	101
10.42.4 Member Data Documentation	101
10.42.4.1 identifier	101
10.42.4.2 mountedHardware	102
10.43Sbs2NetworkHandler Class Reference	102
10.43.1 Constructor & Destructor Documentation	103
10.43.1.1 Sbs2NetworkHandler	103
10.43.2 Member Function Documentation	103
10.43.2.1 addMessageUdpOutputHost	103
10.43.2.2 addRawDataHost	103
10.43.2.3 clearMessageUdpOutputHosts	103
10.43.2.4 messageReceived	103
10.43.2.5 rawDataReceived	103
10.43.2.6 rawDataReceived	103
10.43.2.7 rawDataReceived	103
10.43.2.8 rawDataSentSignal	103
10.43.2.9 readMessage	103
10.43.2.10readRawData	103
10.43.2.11removeMessageUdpOutputHost	103
10.43.2.12removeRawDataHost	103
10.43.2.13sendMessage	103
10.43.2.14sendMessage	103

10.43.2.15	sendRawData	103
10.43.2.16	turnReceiveMessageOff	103
10.43.2.17	turnReceiveMessageOn	103
10.43.2.18	turnReceiveRawDataOff	103
10.43.2.19	turnReceiveRawDataOn	103
10.43.2.20	turnSendRawDataOff	103
10.43.2.21	turnSendRawDataOn	103
10.44	Sbs2Packet Class Reference	104
10.44.1	Constructor & Destructor Documentation	104
10.44.1.1	Sbs2Packet	104
10.44.2	Member Function Documentation	104
10.44.2.1	update	104
10.44.3	Member Data Documentation	105
10.44.3.1	battery	105
10.44.3.2	counter	105
10.44.3.3	cq	105
10.44.3.4	cqIndex	105
10.44.3.5	cqName	105
10.44.3.6	filteredValues	105
10.44.3.7	gyroX	105
10.44.3.8	gyroY	105
10.44.3.9	rawData	105
10.44.3.10	values	105
10.45	Sbs2Region Class Reference	105
10.45.1	Detailed Description	106
10.45.2	Constructor & Destructor Documentation	106
10.45.2.1	Sbs2Region	106
10.45.3	Member Function Documentation	106
10.45.3.1	addRegion	106
10.45.3.2	addRegionsIntersection	106
10.45.3.3	clearVerticesToExtract	106
10.45.3.4	getRegionsToExtract	106
10.45.3.5	getVerticesToExtract	106
10.46	Sbs2SourceReconstrucionLoreta Class Reference	106
10.46.1	Member Enumeration Documentation	107
10.46.1.1	SumType	107
10.46.2	Constructor & Destructor Documentation	107
10.46.2.1	Sbs2SourceReconstrucionLoreta	107
10.46.3	Member Function Documentation	107
10.46.3.1	doRec	107

10.46.3.2 doRecPow	108
10.46.3.3 setAScaling	108
10.46.3.4 setMeanExtraction	108
10.46.3.5 setSumType	108
10.46.3.6 setVerticesToExtract	108
10.46.4 Member Data Documentation	108
10.46.4.1 tempModelUpdatedReady	108
10.47Sbs2SourceReconstruction Class Reference	108
10.47.1 Constructor & Destructor Documentation	109
10.47.1.1 Sbs2SourceReconstruction	109
10.47.2 Member Function Documentation	109
10.47.2.1 doReconstruction	109
10.47.2.2 doReconstructionSpectrogram	109
10.47.2.3 stopReconstruction	109
10.47.2.4 turnOff	109
10.47.2.5 turnOnLoreta	109
10.47.2.6 turnOnSparse	109
10.48Sbs2SourceReconstructionSparse Class Reference	109
10.48.1 Detailed Description	110
10.48.2 Constructor & Destructor Documentation	110
10.48.2.1 Sbs2SourceReconstructionSparse	110
10.48.3 Member Function Documentation	110
10.48.3.1 calculateMean	110
10.48.3.2 calculatePower	110
10.48.3.3 cross_validation_k_channel	111
10.48.3.4 derivative_square_loss_frobenius	111
10.48.3.5 doRec	111
10.48.3.6 doRecPow	111
10.48.3.7 f_objective_general_group_lasso	111
10.48.3.8 fista_method_group_lasso_v2	111
10.48.3.9 preprocessData	111
10.48.3.10proximal_operator_standard_group_lasso	111
10.48.3.11rootMeanSquareError	111
10.49Sbs2Spectrogram Class Reference	111
10.49.1 Member Enumeration Documentation	112
10.49.1.1 WindowType	112
10.49.2 Constructor & Destructor Documentation	112
10.49.2.1 Sbs2Spectrogram	112
10.49.2.2 ~Sbs2Spectrogram	112
10.49.3 Member Function Documentation	112

10.49.3.1 doSpectrogram	112
10.49.3.2 getWindowType	112
10.49.3.3 setWindowType	112
10.50Sbs2Timer Class Reference	112
10.50.1 Detailed Description	113
10.50.2 Constructor & Destructor Documentation	113
10.50.2.1 Sbs2Timer	113
10.50.3 Member Function Documentation	113
10.50.3.1 tic	113
10.50.3.2 tic	113
10.50.3.3 toc	113
10.50.4 Member Data Documentation	113
10.50.4.1 tic_time	113
10.51TNT::Sparse_Matrix_CompRow< T > Class Template Reference	114
10.51.1 Detailed Description	114
10.51.2 Constructor & Destructor Documentation	114
10.51.2.1 Sparse_Matrix_CompRow	114
10.51.2.2 Sparse_Matrix_CompRow	114
10.51.3 Member Function Documentation	115
10.51.3.1 col_ind	115
10.51.3.2 dim1	115
10.51.3.3 dim2	115
10.51.3.4 NumNonzeros	115
10.51.3.5 operator=	115
10.51.3.6 row_ptr	115
10.51.3.7 val	115
10.52TNT::Stopwatch Class Reference	115
10.52.1 Constructor & Destructor Documentation	115
10.52.1.1 Stopwatch	115
10.52.2 Member Function Documentation	115
10.52.2.1 read	115
10.52.2.2 resume	115
10.52.2.3 running	115
10.52.2.4 start	115
10.52.2.5 stop	116
10.53JAMA::SVD< Real > Class Template Reference	116
10.53.1 Detailed Description	116
10.53.2 Constructor & Destructor Documentation	116
10.53.2.1 SVD	116
10.53.3 Member Function Documentation	116

10.53.3.1 cond	116
10.53.3.2 getS	116
10.53.3.3 getSingularValues	117
10.53.3.4 getU	117
10.53.3.5 getV	117
10.53.3.6 norm2	117
10.53.3.7 rank	117
10.54TNT::Vector< T > Class Template Reference	117
10.54.1 Detailed Description	118
10.54.2 Member Typedef Documentation	118
10.54.2.1 const_iterator	118
10.54.2.2 const_reference	118
10.54.2.3 element_type	118
10.54.2.4 iterator	118
10.54.2.5 pointer	118
10.54.2.6 reference	118
10.54.2.7 size_type	118
10.54.2.8 value_type	118
10.54.3 Constructor & Destructor Documentation	119
10.54.3.1 ~Vector	119
10.54.3.2 Vector	119
10.54.3.3 Vector	119
10.54.3.4 Vector	119
10.54.3.5 Vector	119
10.54.3.6 Vector	119
10.54.4 Member Function Documentation	119
10.54.4.1 begin	119
10.54.4.2 begin	119
10.54.4.3 copy	119
10.54.4.4 destroy	119
10.54.4.5 dim	119
10.54.4.6 end	119
10.54.4.7 end	119
10.54.4.8 initialize	119
10.54.4.9 lbound	119
10.54.4.10newsize	119
10.54.4.11operator()	119
10.54.4.12operator()	119
10.54.4.13operator=	119
10.54.4.14operator=	119

10.54.4.15operator[]	119
10.54.4.16operator[]	119
10.54.4.17set	119
10.54.4.18size	119
10.54.5 Member Data Documentation	119
10.54.5.1 n_	119
10.54.5.2 v_	120
10.54.5.3 vm1_	120
10.55 Waiter Class Reference	120
10.55.1 Detailed Description	120
10.55.2 Constructor & Destructor Documentation	120
10.55.2.1 Waiter	120
10.55.3 Member Function Documentation	120
10.55.3.1 run	121
11 File Documentation	123
11.1 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/documentation_static.cpp File Reference	123
11.2 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/dtu_array_2d.h File Reference	123
11.3 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/FFTReal.cpp File Reference	123
11.4 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/FFTReal.h File Reference	123
11.4.1 Macro Definition Documentation	124
11.4.1.1 FFTReal_CURRENT_HEADER	124
11.4.1.2 FFTReal_HEADER_INCLUDED	124
11.4.2 Typedef Documentation	124
11.4.2.1 flt_t	124
11.5 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocapdatareader.cpp File Reference	124
11.6 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocapdatareader.h File Reference	124
11.7 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocapmounter.cpp File Reference	124
11.8 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocapmounter.h File Reference	125
11.9 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocappacket.cpp File Reference	125
11.10 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocappacket.h File Reference	125
11.11 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28datareader.cpp File Reference	125
11.11.1 Function Documentation	125

11.11.1.1 lessThan	125
11.11.1.2 mod	125
11.12/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28datareader.h File Reference	126
11.13/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28mounter.cpp File Reference	126
11.14/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28mounter.h File Reference	126
11.15/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28packet.cpp File Reference	126
11.16/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28packet.h File Reference	126
11.17/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivdatareader.cpp File Reference	127
11.18/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivdatareader.h File Reference	127
11.19/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivdecryptor.h File Reference	127
11.20/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivdecryptor_dummy.cpp File Reference	127
11.21/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivmounter.cpp File Reference	127
11.22/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivmounter.h File Reference	128
11.23/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivpacket.cpp File Reference	128
11.24/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivpacket.h File Reference	128
11.25/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakedatareader.cpp File Reference	128
11.26/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakedatareader.h File Reference	128
11.27/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakepacket.cpp File Reference	129
11.28/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakepacket.h File Reference	129
11.29/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2datareader.cpp File Reference	129
11.30/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2datareader.h File Reference	129
11.31/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2hardwaremounter.cpp File Reference	129
11.32/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2hardwaremounter.h File Reference	130
11.33/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2packet.cpp File Reference	130

11.34/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2packet.h File Reference	130
11.35/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama_cholesky.h File Reference	130
11.36/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama_eig.h File Reference	131
11.37/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama_lu.h File Reference	131
11.38/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama_qr.h File Reference	131
11.39/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama_svd.h File Reference	132
11.40/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt.h File Reference	132
11.41/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array1d.h File Reference	132
11.42/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array1d_utils.h File Reference	133
11.43/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array2d.h File Reference	133
11.44/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array2d_utils.h File Reference	134
11.45/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array3d.h File Reference	134
11.46/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array3d_utils.h File Reference	135
11.47/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_cmat.h File Reference	135
11.48/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array1d.h File Reference	136
11.49/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array1d_utils.h File Reference	137
11.50/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array2d.h File Reference	137
11.51/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array2d_utils.h File Reference	138
11.52/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array3d.h File Reference	138
11.53/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array3d_utils.h File Reference	139
11.54/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_i_refvec.h File Reference	139
11.54.1 Macro Definition Documentation	140
11.54.1.1 NULL	140
11.55/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_math_utils.h File Reference	140

11.56/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_sparse_matrix_csr.h File Reference	140
11.57/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_stopwatch.h File Reference	140
11.58/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_subscript.h File Reference	141
11.58.1 Macro Definition Documentation	141
11.58.1.1 TNT_BASE_OFFSET	141
11.58.1.2 TNT_SUBSCRIPT_TYPE	141
11.59/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_vec.h File Reference	141
11.60/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_version.h File Reference	142
11.60.1 Macro Definition Documentation	142
11.60.1.1 TNT_MAJOR_VERSION	142
11.60.1.2 TNT_MINOR_VERSION	142
11.60.1.3 TNT_SUBMINOR_VERSION	142
11.60.1.4 TNT_VERSION_STRING	142
11.61/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/linux/hid.c File Reference	142
11.61.1 Macro Definition Documentation	144
11.61.1.1 HIDIOCGFEATURE	144
11.61.1.2 HIDIOCSFEATURE	144
11.61.2 Enumeration Type Documentation	144
11.61.2.1 device_string_id	144
11.61.3 Function Documentation	144
11.61.3.1 new_hid_device	144
11.61.3.2 parse_uevent_info	144
11.61.4 Variable Documentation	144
11.61.4.1 device_string_names	144
11.62/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/osx/hid.c File Reference	145
11.62.1 Macro Definition Documentation	146
11.62.1.1 BUF_LEN	146
11.62.2 Typedef Documentation	146
11.62.2.1 pthread_barrier_t	146
11.62.2.2 pthread_barrierattr_t	146
11.63/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/linux/hidapi.h File Reference	146
11.63.1 Macro Definition Documentation	148
11.63.1.1 HID_API_CALL	148
11.63.1.2 HID_API_EXPORT	148

11.63.1.3 HID_API_EXPORT_CALL	148
11.63.2 Typedef Documentation	148
11.63.2.1 hid_device	148
11.64/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/osx/hidapi.h File Reference	148
11.64.1 Macro Definition Documentation	149
11.64.1.1 HID_API_CALL	149
11.64.1.2 HID_API_EXPORT	149
11.64.1.3 HID_API_EXPORT_CALL	150
11.64.2 Typedef Documentation	150
11.64.2.1 hid_device	150
11.65/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/qmlapplicationviewer/qmlapplicationviewer.cpp File Reference	150
11.65.1 Function Documentation	150
11.65.1.1 createApplication	150
11.66/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/qmlapplicationviewer/qmlapplicationviewer.h File Reference	150
11.66.1 Function Documentation	151
11.66.1.1 createApplication	151
11.67/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/README.md File Reference	151
11.68/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2callback.cpp File Reference	151
11.69/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2callback.h File Reference	151
11.70/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2common.cpp File Reference	151
11.71/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2common.h File Reference	151
11.71.1 Macro Definition Documentation	152
11.71.1.1 DEPLOYMENT	152
11.72/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.cpp File Reference	152
11.73/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.h File Reference	152
11.74/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filehandler.cpp File Reference	153
11.75/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filehandler.h File Reference	153
11.76/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filter.cpp File Reference	153
11.77/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filter.h File Reference	153

11.78/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.cpp File Reference	154
11.79/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.h File Reference	154
11.79.1 Macro Definition Documentation	154
11.79.1.1 MAX_BUFFER_SIZE	154
11.80/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2region.cpp File Reference	155
11.81/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2region.h File Reference	155
11.82/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2spectrogram.cpp File Reference	155
11.83/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2spectrogram.h File Reference	155
11.83.1 Macro Definition Documentation	156
11.83.1.1 PI	156
11.84/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/loreta/sbs2sourceconstruction_loreta.cpp File Reference	156
11.85/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/loreta/sbs2sourceconstruction_loreta.h File Reference	156
11.85.1 Macro Definition Documentation	157
11.85.1.1 PI	157
11.86/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sbs2sourceconstruction.cpp File Reference	157
11.87/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sbs2sourceconstruction.h File Reference	157
11.88/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/math_utilities.cpp File Reference	157
11.88.1 Function Documentation	158
11.88.1.1 copyMatrix	158
11.88.1.2 getMean	158
11.88.1.3 loadData	159
11.88.1.4 loadData	159
11.88.1.5 matrixFrobNorm	159
11.88.1.6 matrixL21Norm	159
11.88.1.7 matrixL21NormEachRow	159
11.88.1.8 matrixMultiplicationComponentWise	159
11.88.1.9 printMatrix	159
11.88.1.10 printVector	159
11.88.1.11 scalarDividedbyVectorComponentWise_insitu	159
11.88.1.12 scalarMinusVector_insitu	159
11.88.1.13 thresholding_insitu	159
11.88.1.14 vectorOuterProduct	159

11.89/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/math_utilities.h File Reference	159
11.89.1 Function Documentation	160
11.89.1.1 copyMatrix	160
11.89.1.2 getMean	160
11.89.1.3 loadData	160
11.89.1.4 loadData	160
11.89.1.5 matrixFrobNorm	160
11.89.1.6 matrixL21Norm	160
11.89.1.7 matrixL21NormEachRow	160
11.89.1.8 matrixMultiplicationComponentWise	160
11.89.1.9 printMatrix	160
11.89.1.10 printVector	160
11.89.1.11 scalarDividedbyVectorComponentWise_insitu	160
11.89.1.12 scalarMinusVector_insitu	160
11.89.1.13 thresholding_insitu	160
11.89.1.14 vectorOuterProduct	160
11.90/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/sbs2sourcereconstruction_sparse.cpp File Reference	161
11.91/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/sbs2sourcereconstruction_sparse.h File Reference	161
11.92/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utls/Rijndael.cpp File Reference	161
11.93/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utls/Rijndael.h File Reference	161
11.94/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utls/sbs2timer.cpp File Reference	162
11.95/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utls/sbs2timer.h File Reference	162
11.96/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utls/waiter.h File Reference	162
Index	163

Chapter 1

README

Core source code.

Chapter 2

Todo List

Class **Class**

Meta data as QMap.

Filter should be a container for particular implementations of the filter, either static or adaptive.

Member **DEPLOYMENT**

Loading hardware configuration from a file.

Chapter 3

Module Index

3.1 Modules

Here is a list of all modules:

hidapi API	17
----------------------	----

Chapter 4

Namespace Index

4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

DTU	25
JAMA	25
TNT	26

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

TNT::Array1D< T >	35
TNT::Array1D< double * >	35
TNT::Array1D< double >	35
TNT::Array1D< int >	35
TNT::Array1D< Real * >	35
TNT::Array1D< Real >	35
TNT::Array1D< T * >	35
TNT::Array1D< T ** >	35
TNT::Array2D< T >	36
DTU::DtuArray2D< T >	44
TNT::Array2D< double >	36
DTU::DtuArray2D< double >	44
TNT::Array2D< Real >	36
TNT::Array2D< T * >	36
TNT::Array3D< T >	38
JAMA::Cholesky< Real >	40
Class	42
CRijndael	43
JAMA::Eigenvalue< Real >	46
FFTReal	49
TNT::Fortran_Array1D< T >	50
TNT::Fortran_Array2D< T >	51
TNT::Fortran_Array3D< T >	52
hid_device_	54
hid_device_info	55
TNT::i_refvec< T >	57
TNT::i_refvec< double * >	57
TNT::i_refvec< double >	57
TNT::i_refvec< int >	57
TNT::i_refvec< Real * >	57
TNT::i_refvec< Real >	57
TNT::i_refvec< T * >	57
TNT::i_refvec< T ** >	57
input_report	58
JAMA::LU< Real >	58
TNT::Matrix< T >	60
pthread_barrier	63

QDeclarativeView	
QmlApplicationViewer	64
QmlApplicationViewerPrivate	65
QObject	
Sbs2Callback	67
Sbs2DataHandler	73
Sbs2DataReader	78
Sbs2Emocap28DataReader	81
Sbs2EmocapDataReader	86
Sbs2EmotivDataReader	90
Sbs2FakeDataReader	96
Sbs2EmotivDecryptor	92
Sbs2FileHandler	97
Sbs2Filter	99
Sbs2HardwareMounter	100
Sbs2Emocap28Mounter	83
Sbs2EmocapMounter	88
Sbs2EmotivMounter	93
Sbs2NetworkHandler	102
Sbs2Packet	104
Sbs2Emocap28Packet	85
Sbs2EmocapPacket	89
Sbs2EmotivPacket	94
Sbs2FakePacket	97
Sbs2Region	105
Sbs2SourceReconstrucionLoreta	106
Sbs2SourceReconstruction	108
Sbs2SourceReconstructionSparse	109
Sbs2Spectrogram	111
Sbs2Timer	112
JAMA::QR< Real >	65
QThread	
Waiter	120
Sbs2Common	72
Sbs2Emocap28DataContainer	80
TNT::Sparse_Matrix_CompRow< T >	114
TNT::Stopwatch	115
JAMA::SVD< Real >	116
TNT::Vector< T >	117

Chapter 6

Class Index

6.1 Class List

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

TNT::Array1D< T >	35
TNT::Array2D< T >	36
TNT::Array3D< T >	38
JAMA::Cholesky< Real >	40
Class	42
CRijndael	43
DTU::DtuArray2D< T >	44
JAMA::Eigenvalue< Real >	46
FFTReal	49
TNT::Fortran_Array1D< T >	50
TNT::Fortran_Array2D< T >	51
TNT::Fortran_Array3D< T >	52
hid_device_	54
hid_device_info	55
TNT::i_refvec< T >	57
input_report	58
JAMA::LU< Real >	58
TNT::Matrix< T >	60
pthread_barrier	63
QmlApplicationViewer	64
QmlApplicationViewerPrivate	65
JAMA::QR< Real >	65
Sbs2Callback	67
Sbs2Common	72
Sbs2DataHandler	73
Sbs2DataReader	78
Sbs2Emocap28DataContainer	80
Sbs2Emocap28DataReader	81
Sbs2Emocap28Mounter	83
Sbs2Emocap28Packet	85
Sbs2EmocapDataReader	86
Sbs2EmocapMounter	88
Sbs2EmocapPacket	89
Sbs2EmotivDataReader	90
Sbs2EmotivDecryptor	92
Sbs2EmotivMounter	93
Sbs2EmotivPacket	94
Sbs2FakeDataReader	96

Sbs2FakePacket	97
Sbs2FileHandler	97
Sbs2Filter	99
Sbs2HardwareMounter	100
Sbs2NetworkHandler	102
Sbs2Packet	104
Sbs2Region	105
Sbs2SourceReconstrucionLoreta	106
Sbs2SourceReconstruction	108
Sbs2SourceReconstructionSparse	109
Sbs2Spectrogram	111
Sbs2Timer	112
TNT::Sparse_Matrix_CompRow< T >	114
TNT::Stopwatch	115
JAMA::SVD< Real >	116
TNT::Vector< T >	117
Waiter	120

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/documentation- _static.cpp	123
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/dtu- _array_2d.h	123
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/F- FTReal.cpp	123
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/F- FTReal.h	123
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2callback.- cpp	151
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2callback.- h	151
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2common.- cpp	151
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2common.- h	151
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.- cpp	152
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.- h	152
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filehandler.- cpp	153
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filehandler.- h	153
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filter.- cpp	153
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filter.- h	153
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.- cpp	154
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.- h	154
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2region.- cpp	155
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2region.- h	155
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2spectrogram.- cpp	155

/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2spectrogram.-	
h	155
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2datareader	
cpp	129
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2datareader	
h	129
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2hardwarem	
cpp	129
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2hardwarem	
h	130
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2packet.-	
cpp	130
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2packet.-	
h	130
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2em	
cpp	124
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2em	
h	124
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2em	
cpp	124
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2em	
h	125
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2em	
cpp	125
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2em	
h	125
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2	
cpp	125
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2	
h	126
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2	
cpp	126
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2	
h	126
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2	
cpp	126
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2	
h	126
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2em	
cpp	127
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2em	
h	127
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2em	
h	127
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2em	
_dummy.cpp	127
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2em	
cpp	127
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2em	
h	128
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2em	
cpp	128
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2em	
h	128
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakeda	
cpp	128
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakeda	
h	128

/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakepa	
cpp	129
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakepa	
h	129
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-	
_cholesky.h	130
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-	
_eig.h	131
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-	
_lu.h	131
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-	
_qr.h	131
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-	
_svd.h	132
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
h	132
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_array1d.h	132
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_array1d_utils.h	133
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_array2d.h	133
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_array2d_utils.h	134
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_array3d.h	134
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_array3d_utils.h	135
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_cmat.h	135
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_fortran_array1d.h	136
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_fortran_array1d_utils.h	137
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_fortran_array2d.h	137
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_fortran_array2d_utils.h	138
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_fortran_array3d.h	138
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_fortran_array3d_utils.h	139
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_i_refvec.h	139
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_math_utils.h	140
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_sparse_matrix_csr.h	140
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_stopwatch.h	140
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_subscript.h	141
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_vec.h	141
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-	
_version.h	142
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/linux/hid-	
c	142

/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/linux/hidapi.-	
h	146
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/osx/hid.-	
c	145
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/osx/hidapi.-	
h	148
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/qmlapplicationviewer/qml-	
cpp	150
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/qmlapplicationviewer/qml-	
h	150
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-	
_reconstruction/sbs2sourcereconstruction.cpp	157
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-	
_reconstruction/sbs2sourcereconstruction.h	157
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-	
_reconstruction/loreta/sbs2sourcereconstruction_loreta.cpp	156
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-	
_reconstruction/loreta/sbs2sourcereconstruction_loreta.h	156
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-	
_reconstruction/sparse/math_utilities.cpp	157
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-	
_reconstruction/sparse/math_utilities.h	159
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-	
_reconstruction/sparse/sbs2sourcereconstruction_sparse.cpp	161
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-	
_reconstruction/sparse/sbs2sourcereconstruction_sparse.h	161
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utills/Rijndael.-	
cpp	161
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utills/Rijndael.-	
h	161
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utills/sbs2timer.-	
cpp	162
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utills/sbs2timer.-	
h	162
/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utills/waiter.-	
h	162

Chapter 8

Module Documentation

8.1 hidapi API

Functions

- int [HID_API_EXPORT HID_API_CALL hid_init](#) (void)
Initialize the HIDAPI library.
- int [HID_API_EXPORT HID_API_CALL hid_exit](#) (void)
Finalize the HIDAPI library.
- struct [hid_device_info](#)
[HID_API_EXPORT *HID_API_CALL hid_enumerate](#) (unsigned short vendor_id, unsigned short product_id)
Enumerate the HID Devices.
- void [HID_API_EXPORT HID_API_CALL hid_free_enumeration](#) (struct [hid_device_info](#) *devs)
Free an enumeration Linked List.
- [HID_API_EXPORT hid_device](#)
[*HID_API_CALL hid_open](#) (unsigned short vendor_id, unsigned short product_id, const wchar_t *serial_number)
Open a HID device using a Vendor ID (VID), Product ID (PID) and optionally a serial number.
- [HID_API_EXPORT hid_device](#)
[*HID_API_CALL hid_open_path](#) (const char *path)
Open a HID device by its path name.
- int [HID_API_EXPORT HID_API_CALL hid_write](#) ([hid_device](#) *device, const unsigned char *data, size_t length)
Write an Output report to a HID device.
- int [HID_API_EXPORT HID_API_CALL hid_read_timeout](#) ([hid_device](#) *dev, unsigned char *data, size_t length, int milliseconds)
Read an Input report from a HID device with timeout.
- int [HID_API_EXPORT HID_API_CALL hid_read](#) ([hid_device](#) *device, unsigned char *data, size_t length)
Read an Input report from a HID device.
- int [HID_API_EXPORT HID_API_CALL hid_set_nonblocking](#) ([hid_device](#) *device, int nonblock)
Set the device handle to be non-blocking.
- int [HID_API_EXPORT HID_API_CALL hid_send_feature_report](#) ([hid_device](#) *device, const unsigned char *data, size_t length)
Send a Feature report to the device.
- int [HID_API_EXPORT HID_API_CALL hid_get_feature_report](#) ([hid_device](#) *device, unsigned char *data, size_t length)
Get a feature report from a HID device.
- void [HID_API_EXPORT HID_API_CALL hid_close](#) ([hid_device](#) *device)

Close a HID device.

- int [HID_API_EXPORT_CALL hid_get_manufacturer_string](#) ([hid_device](#) *device, wchar_t *string, size_t maxlen)

Get The Manufacturer String from a HID device.

- int [HID_API_EXPORT_CALL hid_get_product_string](#) ([hid_device](#) *device, wchar_t *string, size_t maxlen)

Get The Product String from a HID device.

- int [HID_API_EXPORT_CALL hid_get_serial_number_string](#) ([hid_device](#) *device, wchar_t *string, size_t maxlen)

Get The Serial Number String from a HID device.

- int [HID_API_EXPORT_CALL hid_get_indexed_string](#) ([hid_device](#) *device, int string_index, wchar_t *string, size_t maxlen)

Get a string from a HID device, based on its string index.

- [HID_API_EXPORT](#) const wchar_t
*[HID_API_CALL hid_error](#) ([hid_device](#) *device)

Get a string describing the last error which occurred.

- [HID_API_EXPORT](#) [hid_device](#)
*[HID_API_CALL hid_open](#) (unsigned short vendor_id, unsigned short product_id, wchar_t *serial_number)

Open a HID device using a Vendor ID (VID), Product ID (PID) and optionally a serial number.

8.1.1 Detailed Description

8.1.2 Function Documentation

8.1.2.1 void [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_close](#) ([hid_device](#) * device)

Close a HID device.

Parameters

<i>device</i>	A device handle returned from hid_open() .
---------------	--

8.1.2.2 struct [hid_device_info](#) [HID_API_EXPORT](#)* [HID_API_CALL](#) [hid_enumerate](#) (unsigned short *vendor_id*, unsigned short *product_id*)

Enumerate the HID Devices.

This function returns a linked list of all the HID devices attached to the system which match *vendor_id* and *product_id*. If *vendor_id* and *product_id* are both set to 0, then all HID devices will be returned.

Parameters

<i>vendor_id</i>	The Vendor ID (VID) of the types of device to open.
<i>product_id</i>	The Product ID (PID) of the types of device to open.

Returns

This function returns a pointer to a linked list of type struct [hid_device](#), containing information about the HID devices attached to the system, or NULL in the case of failure. Free this linked list by calling [hid_free_enumeration\(\)](#).

8.1.2.3 [HID_API_EXPORT](#) const wchar_t* [HID_API_CALL](#) [hid_error](#) ([hid_device](#) * device)

Get a string describing the last error which occurred.

Parameters

<i>device</i>	A device handle returned from hid_open() .
---------------	--

Returns

This function returns a string containing the last error which occurred or NULL if none has occurred.

8.1.2.4 `int HID_API_EXPORT HID_API_CALL hid_exit (void)`

Finalize the HIDAPI library.

This function frees all of the static data associated with HIDAPI. It should be called at the end of execution to avoid memory leaks.

Returns

This function returns 0 on success and -1 on error.

8.1.2.5 `void HID_API_EXPORT HID_API_CALL hid_free_enumeration (struct hid_device_info * devs)`

Free an enumeration Linked List.

This function frees a linked list created by [hid_enumerate\(\)](#).

Parameters

<i>devs</i>	Pointer to a list of struct_device returned from hid_enumerate() .
-------------	--

8.1.2.6 `int HID_API_EXPORT HID_API_CALL hid_get_feature_report (hid_device * device, unsigned char * data, size_t length)`

Get a feature report from a HID device.

Make sure to set the first byte of `data[]` to the Report ID of the report to be read. Make sure to allow space for this extra byte in `data[]`.

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>data</i>	A buffer to put the read data into, including the Report ID. Set the first byte of <code>data[]</code> to the Report ID of the report to be read.
<i>length</i>	The number of bytes to read, including an extra byte for the report ID. The buffer can be longer than the actual report.

Returns

This function returns the number of bytes read and -1 on error.

8.1.2.7 `int HID_API_EXPORT_CALL hid_get_indexed_string (hid_device * device, int string_index, wchar_t * string, size_t maxlen)`

Get a string from a HID device, based on its string index.

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>string_index</i>	The index of the string to get.
<i>string</i>	A wide string buffer to put the data into.
<i>maxlen</i>	The length of the buffer in multiples of <code>wchar_t</code> .

Returns

This function returns 0 on success and -1 on error.

8.1.2.8 `int HID_API_EXPORT_CALL hid_get_manufacturer_string (hid_device * device, wchar_t * string, size_t maxlen)`

Get The Manufacturer String from a HID device.

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>string</i>	A wide string buffer to put the data into.
<i>maxlen</i>	The length of the buffer in multiples of <code>wchar_t</code> .

Returns

This function returns 0 on success and -1 on error.

8.1.2.9 `int HID_API_EXPORT_CALL hid_get_product_string (hid_device * device, wchar_t * string, size_t maxlen)`

Get The Product String from a HID device.

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>string</i>	A wide string buffer to put the data into.
<i>maxlen</i>	The length of the buffer in multiples of <code>wchar_t</code> .

Returns

This function returns 0 on success and -1 on error.

8.1.2.10 `int HID_API_EXPORT_CALL hid_get_serial_number_string (hid_device * device, wchar_t * string, size_t maxlen)`

Get The Serial Number String from a HID device.

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>string</i>	A wide string buffer to put the data into.
<i>maxlen</i>	The length of the buffer in multiples of <code>wchar_t</code> .

Returns

This function returns 0 on success and -1 on error.

8.1.2.11 `int HID_API_EXPORT HID_API_CALL hid_init (void)`

Initialize the HIDAPI library.

This function initializes the HIDAPI library. Calling it is not strictly necessary, as it will be called automatically by [hid_enumerate\(\)](#) and any of the `hid_open_*`() functions if it is needed. This function should be called at the beginning of execution however, if there is a chance of HIDAPI handles being opened by different threads simultaneously.

Returns

This function returns 0 on success and -1 on error.

8.1.2.12 `HID_API_EXPORT hid_device* HID_API_CALL hid_open (unsigned short vendor_id, unsigned short product_id, const wchar_t * serial_number)`

Open a HID device using a Vendor ID (VID), Product ID (PID) and optionally a serial number.

If `serial_number` is NULL, the first device with the specified VID and PID is opened.

Parameters

<i>vendor_id</i>	The Vendor ID (VID) of the device to open.
<i>product_id</i>	The Product ID (PID) of the device to open.
<i>serial_number</i>	The Serial Number of the device to open (Optionally NULL).

Returns

This function returns a pointer to a [hid_device](#) object on success or NULL on failure.

8.1.2.13 `HID_API_EXPORT hid_device* HID_API_CALL hid_open (unsigned short vendor_id, unsigned short product_id, wchar_t * serial_number)`

Open a HID device using a Vendor ID (VID), Product ID (PID) and optionally a serial number.

If `serial_number` is NULL, the first device with the specified VID and PID is opened.

Parameters

<i>vendor_id</i>	The Vendor ID (VID) of the device to open.
<i>product_id</i>	The Product ID (PID) of the device to open.
<i>serial_number</i>	The Serial Number of the device to open (Optionally NULL).

Returns

This function returns a pointer to a [hid_device](#) object on success or NULL on failure.

8.1.2.14 `HID_API_EXPORT hid_device* HID_API_CALL hid_open_path (const char * path)`

Open a HID device by its path name.

The path name be determined by calling [hid_enumerate\(\)](#), or a platform-specific path name can be used (eg: `/dev/hidraw0` on Linux).

Parameters

<i>path</i>	The path name of the device to open
-------------	-------------------------------------

Returns

This function returns a pointer to a [hid_device](#) object on success or NULL on failure.

8.1.2.15 `int HID_API_EXPORT HID_API_CALL hid_read (hid_device * device, unsigned char * data, size_t length)`

Read an Input report from a HID device.

Input reports are returned to the host through the INTERRUPT IN endpoint. The first byte will contain the Report number if the device uses numbered reports.

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>data</i>	A buffer to put the read data into.
<i>length</i>	The number of bytes to read. For devices with multiple reports, make sure to read an extra byte for the report number.

Returns

This function returns the actual number of bytes read and -1 on error.

8.1.2.16 `int HID_API_EXPORT HID_API_CALL hid_read_timeout (hid_device * dev, unsigned char * data, size_t length, int milliseconds)`

Read an Input report from a HID device with timeout.

Input reports are returned to the host through the INTERRUPT IN endpoint. The first byte will contain the Report number if the device uses numbered reports.

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>data</i>	A buffer to put the read data into.
<i>length</i>	The number of bytes to read. For devices with multiple reports, make sure to read an extra byte for the report number.
<i>milliseconds</i>	timeout in milliseconds or -1 for blocking wait.

Returns

This function returns the actual number of bytes read and -1 on error.

8.1.2.17 `int HID_API_EXPORT HID_API_CALL hid_send_feature_report (hid_device * device, const unsigned char * data, size_t length)`

Send a Feature report to the device.

Feature reports are sent over the Control endpoint as a Set_Report transfer. The first byte of `data[]` must contain the Report ID. For devices which only support a single report, this must be set to 0x0. The remaining bytes contain the report data. Since the Report ID is mandatory, calls to [hid_send_feature_report\(\)](#) will always contain one more byte than the report contains. For example, if a hid report is 16 bytes long, 17 bytes must be passed to [hid_send_feature_report\(\)](#): the Report ID (or 0x0, for devices which do not use numbered reports), followed by the report data (16 bytes). In this example, the length passed in would be 17.

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>data</i>	The data to send, including the report number as the first byte.
<i>length</i>	The length in bytes of the data to send, including the report number.

Returns

This function returns the actual number of bytes written and -1 on error.

8.1.2.18 `int HID_API_EXPORT HID_API_CALL hid_set_nonblocking (hid_device * device, int nonblock)`

Set the device handle to be non-blocking.

In non-blocking mode calls to [hid_read\(\)](#) will return immediately with a value of 0 if there is no data to be read. In blocking mode, [hid_read\(\)](#) will wait (block) until there is data to read before returning.

Nonblocking can be turned on and off at any time.

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>nonblock</i>	enable or not the nonblocking reads <ul style="list-style-type: none"> • 1 to enable nonblocking • 0 to disable nonblocking.

Returns

This function returns 0 on success and -1 on error.

8.1.2.19 `int HID_API_EXPORT HID_API_CALL hid_write (hid_device * device, const unsigned char * data, size_t length)`

Write an Output report to a HID device.

The first byte of `data[]` must contain the Report ID. For devices which only support a single report, this must be set to 0x0. The remaining bytes contain the report data. Since the Report ID is mandatory, calls to [hid_write\(\)](#) will always contain one more byte than the report contains. For example, if a hid report is 16 bytes long, 17 bytes must be passed to [hid_write\(\)](#), the Report ID (or 0x0, for devices with a single report), followed by the report data (16 bytes). In this example, the length passed in would be 17.

[hid_write\(\)](#) will send the data on the first OUT endpoint, if one exists. If it does not, it will send the data through the Control Endpoint (Endpoint 0).

Parameters

<i>device</i>	A device handle returned from hid_open() .
<i>data</i>	The data to send, including the report number as the first byte.
<i>length</i>	The length in bytes of the data to send.

Returns

This function returns the actual number of bytes written and -1 on error.

Chapter 9

Namespace Documentation

9.1 DTU Namespace Reference

Classes

- class [DtuArray2D](#)

9.1.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU](#) Informatics, Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
for pow(float, float)

9.2 JAMA Namespace Reference

Classes

- class [Cholesky](#)
- class [Eigenvalue](#)
- class [LU](#)
- class [QR](#)
- class [SVD](#)

9.3 TNT Namespace Reference

Classes

- class [Array1D](#)
- class [Array2D](#)
- class [Array3D](#)
- class [Matrix](#)
- class [Fortran_Array1D](#)
- class [Fortran_Array2D](#)
- class [Fortran_Array3D](#)
- class [i_refvec](#)
- class [Sparse_Matrix_CompRow](#)
- class [Stopwatch](#)
- class [Vector](#)

Typedefs

- typedef [TNT_SUBSCRIPT_TYPE](#) Subscript

Functions

- template<class T >
std::ostream & [operator<<](#) (std::ostream &s, const [Array1D](#)< T > &A)
- template<class T >
std::istream & [operator>>](#) (std::istream &s, [Array1D](#)< T > &A)
- template<class T >
[Array1D](#)< T > [operator+](#) (const [Array1D](#)< T > &A, const [Array1D](#)< T > &B)
- template<class T >
[Array1D](#)< T > [operator-](#) (const [Array1D](#)< T > &A, const [Array1D](#)< T > &B)
- template<class T >
[Array1D](#)< T > [operator*](#) (const [Array1D](#)< T > &A, const [Array1D](#)< T > &B)
- template<class T >
[Array1D](#)< T > [operator/](#) (const [Array1D](#)< T > &A, const [Array1D](#)< T > &B)
- template<class T >
[Array1D](#)< T > & [operator+=](#) ([Array1D](#)< T > &A, const [Array1D](#)< T > &B)
- template<class T >
[Array1D](#)< T > & [operator-=](#) ([Array1D](#)< T > &A, const [Array1D](#)< T > &B)
- template<class T >
[Array1D](#)< T > & [operator*=](#) ([Array1D](#)< T > &A, const [Array1D](#)< T > &B)
- template<class T >
[Array1D](#)< T > & [operator/=](#) ([Array1D](#)< T > &A, const [Array1D](#)< T > &B)
- template<class T >
std::ostream & [operator<<](#) (std::ostream &s, const [Array2D](#)< T > &A)
- template<class T >
std::istream & [operator>>](#) (std::istream &s, [Array2D](#)< T > &A)
- template<class T >
[Array2D](#)< T > [operator+](#) (const [Array2D](#)< T > &A, const [Array2D](#)< T > &B)
- template<class T >
[Array2D](#)< T > [operator-](#) (const [Array2D](#)< T > &A, const [Array2D](#)< T > &B)
- template<class T >
[Array2D](#)< T > [operator*](#) (const [Array2D](#)< T > &A, const [Array2D](#)< T > &B)
- template<class T >
[Array2D](#)< T > [operator/](#) (const [Array2D](#)< T > &A, const [Array2D](#)< T > &B)

- `template<class T >`
`Array2D< T > & operator+= (Array2D< T > &A, const Array2D< T > &B)`
- `template<class T >`
`Array2D< T > & operator-= (Array2D< T > &A, const Array2D< T > &B)`
- `template<class T >`
`Array2D< T > & operator*= (Array2D< T > &A, const Array2D< T > &B)`
- `template<class T >`
`Array2D< T > & operator/= (Array2D< T > &A, const Array2D< T > &B)`
- `template<class T >`
`Array2D< T > matmult (const Array2D< T > &A, const Array2D< T > &B)`
- `template<class T >`
`std::ostream & operator<< (std::ostream &s, const Array3D< T > &A)`
- `template<class T >`
`std::istream & operator>> (std::istream &s, Array3D< T > &A)`
- `template<class T >`
`Array3D< T > operator+ (const Array3D< T > &A, const Array3D< T > &B)`
- `template<class T >`
`Array3D< T > operator- (const Array3D< T > &A, const Array3D< T > &B)`
- `template<class T >`
`Array3D< T > operator* (const Array3D< T > &A, const Array3D< T > &B)`
- `template<class T >`
`Array3D< T > operator/ (const Array3D< T > &A, const Array3D< T > &B)`
- `template<class T >`
`Array3D< T > & operator+= (Array3D< T > &A, const Array3D< T > &B)`
- `template<class T >`
`Array3D< T > & operator-= (Array3D< T > &A, const Array3D< T > &B)`
- `template<class T >`
`Array3D< T > & operator*= (Array3D< T > &A, const Array3D< T > &B)`
- `template<class T >`
`Array3D< T > & operator/= (Array3D< T > &A, const Array3D< T > &B)`
- `template<class T >`
`std::ostream & operator<< (std::ostream &s, const Matrix< T > &A)`
- `template<class T >`
`std::istream & operator>> (std::istream &s, Matrix< T > &A)`
- `template<class T >`
`Matrix< T > operator+ (const Matrix< T > &A, const Matrix< T > &B)`
- `template<class T >`
`Matrix< T > operator- (const Matrix< T > &A, const Matrix< T > &B)`
- `template<class T >`
`Matrix< T > mult_element (const Matrix< T > &A, const Matrix< T > &B)`
- `template<class T >`
`Matrix< T > transpose (const Matrix< T > &A)`
- `template<class T >`
`Matrix< T > matmult (const Matrix< T > &A, const Matrix< T > &B)`
- `template<class T >`
`Matrix< T > operator* (const Matrix< T > &A, const Matrix< T > &B)`
- `template<class T >`
`int matmult (Matrix< T > &C, const Matrix< T > &A, const Matrix< T > &B)`
- `template<class T >`
`Vector< T > matmult (const Matrix< T > &A, const Vector< T > &x)`
- `template<class T >`
`Vector< T > operator* (const Matrix< T > &A, const Vector< T > &x)`
- `template<class T >`
`std::ostream & operator<< (std::ostream &s, const Fortran_Array1D< T > &A)`
- `template<class T >`
`std::istream & operator>> (std::istream &s, Fortran_Array1D< T > &A)`

- `template<class Real >`
Real [hypot](#) (const Real &a, const Real &b)
- `template<class T >`
`std::ostream & operator<<` (std::ostream &s, const [Vector](#)< T > &A)
- `template<class T >`
`std::istream & operator>>` (std::istream &s, [Vector](#)< T > &A)
- `template<class T >`
[Vector](#)< T > [operator+](#) (const [Vector](#)< T > &A, const [Vector](#)< T > &B)
- `template<class T >`
[Vector](#)< T > [operator-](#) (const [Vector](#)< T > &A, const [Vector](#)< T > &B)
- `template<class T >`
[Vector](#)< T > [operator*](#) (const [Vector](#)< T > &A, const [Vector](#)< T > &B)
- `template<class T >`
T [dot_prod](#) (const [Vector](#)< T > &A, const [Vector](#)< T > &B)

9.3.1 Typedef Documentation

9.3.1.1 typedef TNT_SUBSCRIPT_TYPE TNT::Subscript

9.3.2 Function Documentation

9.3.2.1 `template<class T > T TNT::dot_prod (const Vector< T > &A, const Vector< T > &B)`

9.3.2.2 `template<class Real > Real TNT::hypot (const Real &a, const Real &b)`

Returns

hypotenuse of real (non-complex) scalars a and b by avoiding underflow/overflow using $(a * \sqrt{1 + (b/a) * (b/a)})$, rather than $\sqrt{a*a + b*b}$.

9.3.2.3 `template<class T > Array2D<T> TNT::matmult (const Array2D< T > &A, const Array2D< T > &B)`

[Matrix](#) Multiply: compute $C = A*B$, where $C[i][j]$ is the dot-product of row i of A and column j of B.

Parameters

<i>A</i>	an (m x n) array
<i>B</i>	an (n x k) array

Returns

the (m x k) array $A*B$, or a null array (0x0) if the matrices are non-conformant (i.e. the number of columns of A are different than the number of rows of B.)

9.3.2.4 `template<class T > Matrix<T> TNT::matmult (const Matrix< T > &A, const Matrix< T > &B)` `[inline]`

9.3.2.5 `template<class T > int TNT::matmult (Matrix< T > &C, const Matrix< T > &A, const Matrix< T > &B)` `[inline]`

9.3.2.6 `template<class T > Vector<T> TNT::matmult (const Matrix< T > &A, const Vector< T > &x)`

9.3.2.7 `template<class T > Matrix<T> TNT::mult_element (const Matrix< T > &A, const Matrix< T > &B)`

9.3.2.8 `template<class T > Array1D<T> TNT::operator* (const Array1D< T > &A, const Array1D< T > &B)`

- 9.3.2.9 `template<class T> Array3D<T> TNT::operator* (const Array3D< T> & A, const Array3D< T> & B)`
- 9.3.2.10 `template<class T> Array2D<T> TNT::operator* (const Array2D< T> & A, const Array2D< T> & B)`
- 9.3.2.11 `template<class T> Fortran_Array2D<T> TNT::operator* (const Fortran_Array2D< T> & A, const Fortran_Array2D< T> & B)`
- 9.3.2.12 `template<class T> Fortran_Array1D<T> TNT::operator* (const Fortran_Array1D< T> & A, const Fortran_Array1D< T> & B)`
- 9.3.2.13 `template<class T> Fortran_Array3D<T> TNT::operator* (const Fortran_Array3D< T> & A, const Fortran_Array3D< T> & B)`
- 9.3.2.14 `template<class T> Vector<T> TNT::operator* (const Vector< T> & A, const Vector< T> & B)`
- 9.3.2.15 `template<class T> Matrix<T> TNT::operator* (const Matrix< T> & A, const Matrix< T> & B)` `[inline]`
- 9.3.2.16 `template<class T> Vector<T> TNT::operator* (const Matrix< T> & A, const Vector< T> & x)` `[inline]`
- 9.3.2.17 `template<class T> Array1D<T>& TNT::operator*= (Array1D< T> & A, const Array1D< T> & B)`
- 9.3.2.18 `template<class T> Array3D<T>& TNT::operator*= (Array3D< T> & A, const Array3D< T> & B)`
- 9.3.2.19 `template<class T> Fortran_Array2D<T>& TNT::operator*= (Fortran_Array2D< T> & A, const Fortran_Array2D< T> & B)`
- 9.3.2.20 `template<class T> Fortran_Array1D<T>& TNT::operator*= (Fortran_Array1D< T> & A, const Fortran_Array1D< T> & B)`
- 9.3.2.21 `template<class T> Array2D<T>& TNT::operator*= (Array2D< T> & A, const Array2D< T> & B)`
- 9.3.2.22 `template<class T> Fortran_Array3D<T>& TNT::operator*= (Fortran_Array3D< T> & A, const Fortran_Array3D< T> & B)`
- 9.3.2.23 `template<class T> Array3D<T> TNT::operator+ (const Array3D< T> & A, const Array3D< T> & B)`
- 9.3.2.24 `template<class T> Array1D<T> TNT::operator+ (const Array1D< T> & A, const Array1D< T> & B)`
- 9.3.2.25 `template<class T> Array2D<T> TNT::operator+ (const Array2D< T> & A, const Array2D< T> & B)`
- 9.3.2.26 `template<class T> Fortran_Array2D<T> TNT::operator+ (const Fortran_Array2D< T> & A, const Fortran_Array2D< T> & B)`
- 9.3.2.27 `template<class T> Fortran_Array3D<T> TNT::operator+ (const Fortran_Array3D< T> & A, const Fortran_Array3D< T> & B)`
- 9.3.2.28 `template<class T> Fortran_Array1D<T> TNT::operator+ (const Fortran_Array1D< T> & A, const Fortran_Array1D< T> & B)`
- 9.3.2.29 `template<class T> Vector<T> TNT::operator+ (const Vector< T> & A, const Vector< T> & B)`
- 9.3.2.30 `template<class T> Matrix<T> TNT::operator+ (const Matrix< T> & A, const Matrix< T> & B)`
- 9.3.2.31 `template<class T> Array1D<T>& TNT::operator+= (Array1D< T> & A, const Array1D< T> & B)`
- 9.3.2.32 `template<class T> Array3D<T>& TNT::operator+= (Array3D< T> & A, const Array3D< T> & B)`

- 9.3.2.33 `template<class T> Fortran_Array2D<T>& TNT::operator+=(Fortran_Array2D< T> & A, const Fortran_Array2D< T> & B)`
- 9.3.2.34 `template<class T> Array2D<T>& TNT::operator+=(Array2D< T> & A, const Array2D< T> & B)`
- 9.3.2.35 `template<class T> Fortran_Array3D<T>& TNT::operator+=(Fortran_Array3D< T> & A, const Fortran_Array3D< T> & B)`
- 9.3.2.36 `template<class T> Fortran_Array1D<T>& TNT::operator+=(Fortran_Array1D< T> & A, const Fortran_Array1D< T> & B)`
- 9.3.2.37 `template<class T> Array3D<T> TNT::operator-(const Array3D< T> & A, const Array3D< T> & B)`
- 9.3.2.38 `template<class T> Array1D<T> TNT::operator-(const Array1D< T> & A, const Array1D< T> & B)`
- 9.3.2.39 `template<class T> Array2D<T> TNT::operator-(const Array2D< T> & A, const Array2D< T> & B)`
- 9.3.2.40 `template<class T> Fortran_Array2D<T> TNT::operator-(const Fortran_Array2D< T> & A, const Fortran_Array2D< T> & B)`
- 9.3.2.41 `template<class T> Fortran_Array3D<T> TNT::operator-(const Fortran_Array3D< T> & A, const Fortran_Array3D< T> & B)`
- 9.3.2.42 `template<class T> Fortran_Array1D<T> TNT::operator-(const Fortran_Array1D< T> & A, const Fortran_Array1D< T> & B)`
- 9.3.2.43 `template<class T> Vector<T> TNT::operator-(const Vector< T> & A, const Vector< T> & B)`
- 9.3.2.44 `template<class T> Matrix<T> TNT::operator-(const Matrix< T> & A, const Matrix< T> & B)`
- 9.3.2.45 `template<class T> Array1D<T>& TNT::operator=(Array1D< T> & A, const Array1D< T> & B)`
- 9.3.2.46 `template<class T> Array3D<T>& TNT::operator=(Array3D< T> & A, const Array3D< T> & B)`
- 9.3.2.47 `template<class T> Fortran_Array2D<T>& TNT::operator=(Fortran_Array2D< T> & A, const Fortran_Array2D< T> & B)`
- 9.3.2.48 `template<class T> Array2D<T>& TNT::operator=(Array2D< T> & A, const Array2D< T> & B)`
- 9.3.2.49 `template<class T> Fortran_Array1D<T>& TNT::operator=(Fortran_Array1D< T> & A, const Fortran_Array1D< T> & B)`
- 9.3.2.50 `template<class T> Fortran_Array3D<T>& TNT::operator=(Fortran_Array3D< T> & A, const Fortran_Array3D< T> & B)`
- 9.3.2.51 `template<class T> Array1D<T> TNT::operator/(const Array1D< T> & A, const Array1D< T> & B)`
- 9.3.2.52 `template<class T> Array3D<T> TNT::operator/(const Array3D< T> & A, const Array3D< T> & B)`
- 9.3.2.53 `template<class T> Fortran_Array2D<T> TNT::operator/(const Fortran_Array2D< T> & A, const Fortran_Array2D< T> & B)`
- 9.3.2.54 `template<class T> Array2D<T> TNT::operator/(const Array2D< T> & A, const Array2D< T> & B)`
- 9.3.2.55 `template<class T> Fortran_Array1D<T> TNT::operator/(const Fortran_Array1D< T> & A, const Fortran_Array1D< T> & B)`

- 9.3.2.56 `template<class T> Fortran_Array3D<T> TNT::operator/ (const Fortran_Array3D< T> & A, const Fortran_Array3D< T> & B)`
- 9.3.2.57 `template<class T> Array1D<T>& TNT::operator/= (Array1D< T> & A, const Array1D< T> & B)`
- 9.3.2.58 `template<class T> Array3D<T>& TNT::operator/= (Array3D< T> & A, const Array3D< T> & B)`
- 9.3.2.59 `template<class T> Fortran_Array2D<T>& TNT::operator/= (Fortran_Array2D< T> & A, const Fortran_Array2D< T> & B)`
- 9.3.2.60 `template<class T> Fortran_Array1D<T>& TNT::operator/= (Fortran_Array1D< T> & A, const Fortran_Array1D< T> & B)`
- 9.3.2.61 `template<class T> Array2D<T>& TNT::operator/= (Array2D< T> & A, const Array2D< T> & B)`
- 9.3.2.62 `template<class T> Fortran_Array3D<T>& TNT::operator/= (Fortran_Array3D< T> & A, const Fortran_Array3D< T> & B)`
- 9.3.2.63 `template<class T> std::ostream& TNT::operator<< (std::ostream & s, const Array3D< T> & A)`
- 9.3.2.64 `template<class T> std::ostream& TNT::operator<< (std::ostream & s, const Array1D< T> & A)`
- 9.3.2.65 `template<class T> std::ostream& TNT::operator<< (std::ostream & s, const Fortran_Array2D< T> & A)`
- 9.3.2.66 `template<class T> std::ostream& TNT::operator<< (std::ostream & s, const Fortran_Array3D< T> & A)`
- 9.3.2.67 `template<class T> std::ostream& TNT::operator<< (std::ostream & s, const Array2D< T> & A)`
- 9.3.2.68 `template<class T> std::ostream& TNT::operator<< (std::ostream & s, const Fortran_Array1D< T> & A)`

Write an array to a character outstream. Output format is one that can be read back in via the in-stream operator: one integer denoting the array dimension (n), followed by n elements, one per line.

- 9.3.2.69 `template<class T> std::ostream& TNT::operator<< (std::ostream & s, const Vector< T> & A)`
- 9.3.2.70 `template<class T> std::ostream& TNT::operator<< (std::ostream & s, const Matrix< T> & A)`
- 9.3.2.71 `template<class T> std::istream& TNT::operator>> (std::istream & s, Array3D< T> & A)`
- 9.3.2.72 `template<class T> std::istream& TNT::operator>> (std::istream & s, Array1D< T> & A)`
- 9.3.2.73 `template<class T> std::istream& TNT::operator>> (std::istream & s, Fortran_Array2D< T> & A)`
- 9.3.2.74 `template<class T> std::istream& TNT::operator>> (std::istream & s, Array2D< T> & A)`
- 9.3.2.75 `template<class T> std::istream& TNT::operator>> (std::istream & s, Fortran_Array3D< T> & A)`
- 9.3.2.76 `template<class T> std::istream& TNT::operator>> (std::istream & s, Fortran_Array1D< T> & A)`

Read an array from a character stream. Input format is one integer, denoting the dimension (n), followed by n whitespace-separated elements. Newlines are ignored

Note: the array being read into references new memory storage. If the intent is to fill an existing conformant array, use `cin >> B; A.inject(B) ;` instead or read the elements in one-a-time by hand.

Parameters

<i>s</i>	the charater to read from (typically <code>std::in</code>)
<i>A</i>	the array to read into.

9.3.2.77 `template<class T > std::istream& TNT::operator>> (std::istream & s, Vector< T > & A)`

9.3.2.78 `template<class T > std::istream& TNT::operator>> (std::istream & s, Matrix< T > & A)`

9.3.2.79 `template<class T > Matrix<T> TNT::transpose (const Matrix< T > & A)`

Chapter 10

Class Documentation

10.1 TNT::Array1D< T > Class Template Reference

```
#include <tnt_array1d.h>
```

Public Types

- typedef T [value_type](#)

Public Member Functions

- [Array1D](#) ()
- [Array1D](#) (int n)
- [Array1D](#) (int n, const T &a)
- [Array1D](#) (int n, T *a)
- [Array1D](#) (const [Array1D](#) &A)
- [operator T *](#) ()
- [operator const T *](#) ()
- [Array1D](#) & [operator=](#) (const T &a)
- [Array1D](#) & [operator=](#) (const [Array1D](#) &A)
- [Array1D](#) & [ref](#) (const [Array1D](#) &A)
- [Array1D](#) [copy](#) () const
- [Array1D](#) & [inject](#) (const [Array1D](#) &A)
- T & [operator\[\]](#) (int i)
- const T & [operator\[\]](#) (int i) const
- int [dim1](#) () const
- int [dim](#) () const
- [~Array1D](#) ()
- int [ref_count](#) () const
- [Array1D](#)< T > [subarray](#) (int i0, int i1)

10.1.1 Member Typedef Documentation

10.1.1.1 `template<class T> typedef T TNT::Array1D< T >::value_type`

10.1.2 Constructor & Destructor Documentation

10.1.2.1 `template<class T > TNT::Array1D< T >::Array1D ()`

10.1.2.2 `template<class T> TNT::Array1D< T >::Array1D (int n) [explicit]`

10.1.2.3 `template<class T> TNT::Array1D< T >::Array1D (int n, const T & a)`

10.1.2.4 `template<class T> TNT::Array1D< T >::Array1D (int n, T * a)`

10.1.2.5 `template<class T> TNT::Array1D< T >::Array1D (const Array1D< T > & A) [inline]`

10.1.2.6 `template<class T> TNT::Array1D< T >::~~Array1D ()`

10.1.3 Member Function Documentation

10.1.3.1 `template<class T> Array1D< T > TNT::Array1D< T >::copy () const`

10.1.3.2 `template<class T> int TNT::Array1D< T >::dim () const [inline]`

10.1.3.3 `template<class T> int TNT::Array1D< T >::dim1 () const [inline]`

10.1.3.4 `template<class T> Array1D< T > & TNT::Array1D< T >::inject (const Array1D< T > & A)`

10.1.3.5 `template<class T> TNT::Array1D< T >::operator const T * () [inline]`

10.1.3.6 `template<class T> TNT::Array1D< T >::operator T * () [inline]`

10.1.3.7 `template<class T> Array1D< T > & TNT::Array1D< T >::operator= (const T & a) [inline]`

10.1.3.8 `template<class T> Array1D< T > & TNT::Array1D< T >::operator= (const Array1D< T > & A) [inline]`

10.1.3.9 `template<class T> T & TNT::Array1D< T >::operator[] (int i) [inline]`

10.1.3.10 `template<class T> const T & TNT::Array1D< T >::operator[] (int i) const [inline]`

10.1.3.11 `template<class T> Array1D< T > & TNT::Array1D< T >::ref (const Array1D< T > & A) [inline]`

10.1.3.12 `template<class T> int TNT::Array1D< T >::ref_count () const [inline]`

10.1.3.13 `template<class T> Array1D< T > TNT::Array1D< T >::subarray (int i0, int i1) [inline]`

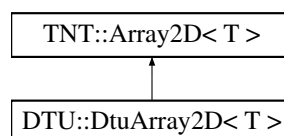
The documentation for this class was generated from the following file:

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array1d.h](#)

10.2 TNT::Array2D< T > Class Template Reference

```
#include <tnt_array2d.h>
```

Inheritance diagram for TNT::Array2D< T >:



Public Types

- typedef T [value_type](#)

Public Member Functions

- [Array2D](#) ()
- [Array2D](#) (int m, int n)
- [Array2D](#) (int m, int n, T *a)
- [Array2D](#) (int m, int n, const T &a)
- [Array2D](#) (const [Array2D](#) &A)
- [operator T **](#) ()
- [operator const T **](#) ()
- [Array2D](#) & [operator=](#) (const T &a)
- [Array2D](#) & [operator=](#) (const [Array2D](#) &A)
- [Array2D](#) & [ref](#) (const [Array2D](#) &A)
- [Array2D](#) [copy](#) () const
- [Array2D](#) & [inject](#) (const [Array2D](#) &A)
- T * [operator\[\]](#) (int i)
- const T * [operator\[\]](#) (int i) const
- int [dim1](#) () const
- int [dim2](#) () const
- [~Array2D](#) ()
- int [ref_count](#) ()
- int [ref_count_data](#) ()
- int [ref_count_dim1](#) ()
- [Array2D](#) subarray (int i0, int i1, int j0, int j1)

Protected Attributes

- [Array1D](#)< T > [data_](#)
- [Array1D](#)< T * > [v_](#)
- int [m_](#)
- int [n_](#)

10.2.1 Member Typedef Documentation

10.2.1.1 `template<class T> typedef T TNT::Array2D< T >::value_type`

10.2.2 Constructor & Destructor Documentation

10.2.2.1 `template<class T> TNT::Array2D< T >::Array2D ()`

10.2.2.2 `template<class T> TNT::Array2D< T >::Array2D (int m, int n)`

10.2.2.3 `template<class T> TNT::Array2D< T >::Array2D (int m, int n, T * a)`

10.2.2.4 `template<class T> TNT::Array2D< T >::Array2D (int m, int n, const T & a)`

10.2.2.5 `template<class T> TNT::Array2D< T >::Array2D (const Array2D< T > & A) [inline]`

10.2.2.6 `template<class T> TNT::Array2D< T >::~~Array2D ()`

10.2.3 Member Function Documentation

- 10.2.3.1 `template<class T> Array2D< T> TNT::Array2D< T>::copy () const`
- 10.2.3.2 `template<class T> int TNT::Array2D< T>::dim1 () const` `[inline]`
- 10.2.3.3 `template<class T> int TNT::Array2D< T>::dim2 () const` `[inline]`
- 10.2.3.4 `template<class T> Array2D< T> & TNT::Array2D< T>::inject (const Array2D< T> & A)`
- 10.2.3.5 `template<class T> TNT::Array2D< T>::operator const T ** ()` `[inline]`
- 10.2.3.6 `template<class T> TNT::Array2D< T>::operator T ** ()` `[inline]`
- 10.2.3.7 `template<class T> Array2D< T> & TNT::Array2D< T>::operator= (const T & a)` `[inline]`
- 10.2.3.8 `template<class T> Array2D< T> & TNT::Array2D< T>::operator= (const Array2D< T> & A)`
`[inline]`
- 10.2.3.9 `template<class T> T * TNT::Array2D< T>::operator[] (int i)` `[inline]`
- 10.2.3.10 `template<class T> const T * TNT::Array2D< T>::operator[] (int i) const` `[inline]`
- 10.2.3.11 `template<class T> Array2D< T> & TNT::Array2D< T>::ref (const Array2D< T> & A)` `[inline]`
- 10.2.3.12 `template<class T> int TNT::Array2D< T>::ref_count ()` `[inline]`
- 10.2.3.13 `template<class T> int TNT::Array2D< T>::ref_count_data ()` `[inline]`
- 10.2.3.14 `template<class T> int TNT::Array2D< T>::ref_count_dim1 ()` `[inline]`
- 10.2.3.15 `template<class T> Array2D< T> TNT::Array2D< T>::subarray (int i0, int i1, int j0, int j1)`

Create a new view to a subarray defined by the boundaries `[i0][i0]` and `[i1][j1]`. The size of the subarray is `(i1-i0)` by `(j1-j0)`. If either of these lengths are zero or negative, the subarray view is null.

10.2.4 Member Data Documentation

- 10.2.4.1 `template<class T> Array1D<T> TNT::Array2D< T>::data_` `[protected]`
- 10.2.4.2 `template<class T> int TNT::Array2D< T>::m_` `[protected]`
- 10.2.4.3 `template<class T> int TNT::Array2D< T>::n_` `[protected]`
- 10.2.4.4 `template<class T> Array1D<T*> TNT::Array2D< T>::v_` `[protected]`

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

- [/media/philipjh/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-_array2d.h](#)

10.3 TNT::Array3D< T> Class Template Reference

```
#include <tnt_array3d.h>
```

Public Types

- typedef T [value_type](#)

Public Member Functions

- [Array3D](#) ()
- [Array3D](#) (int m, int n, int g)
- [Array3D](#) (int m, int n, int g, T val)
- [Array3D](#) (int m, int n, int g, T *a)
- [operator T ***](#) ()
- [operator const T ***](#) ()
- [Array3D](#) (const [Array3D](#) &A)
- [Array3D](#) & [operator=](#) (const T &a)
- [Array3D](#) & [operator=](#) (const [Array3D](#) &A)
- [Array3D](#) & [ref](#) (const [Array3D](#) &A)
- [Array3D](#) [copy](#) () const
- [Array3D](#) & [inject](#) (const [Array3D](#) &A)
- T ** [operator\[\]](#) (int i)
- const T *const * [operator\[\]](#) (int i) const
- int [dim1](#) () const
- int [dim2](#) () const
- int [dim3](#) () const
- [~Array3D](#) ()
- int [ref_count](#) ()
- [Array3D](#) [subarray](#) (int i0, int i1, int j0, int j1, int k0, int k1)

10.3.1 Member Typedef Documentation

10.3.1.1 `template<class T> typedef T TNT::Array3D< T >::value_type`

10.3.2 Constructor & Destructor Documentation

10.3.2.1 `template<class T> TNT::Array3D< T >::Array3D ()`

10.3.2.2 `template<class T> TNT::Array3D< T >::Array3D (int m, int n, int g)`

10.3.2.3 `template<class T> TNT::Array3D< T >::Array3D (int m, int n, int g, T val)`

10.3.2.4 `template<class T> TNT::Array3D< T >::Array3D (int m, int n, int g, T *a)`

10.3.2.5 `template<class T> TNT::Array3D< T >::Array3D (const Array3D< T > &A)` `[inline]`

10.3.2.6 `template<class T> TNT::Array3D< T >::~~Array3D ()`

10.3.3 Member Function Documentation

10.3.3.1 `template<class T> Array3D< T > TNT::Array3D< T >::copy () const`

10.3.3.2 `template<class T> int TNT::Array3D< T >::dim1 () const` `[inline]`

10.3.3.3 `template<class T> int TNT::Array3D< T >::dim2 () const` `[inline]`

10.3.3.4 `template<class T> int TNT::Array3D< T >::dim3 () const` `[inline]`

- 10.3.3.5 `template<class T> Array3D< T> & TNT::Array3D< T>::inject (const Array3D< T> & A)`
- 10.3.3.6 `template<class T> TNT::Array3D< T>::operator const T *** () [inline]`
- 10.3.3.7 `template<class T> TNT::Array3D< T>::operator T *** () [inline]`
- 10.3.3.8 `template<class T> Array3D< T> & TNT::Array3D< T>::operator= (const T & a) [inline]`
- 10.3.3.9 `template<class T> Array3D< T> & TNT::Array3D< T>::operator= (const Array3D< T> & A) [inline]`
- 10.3.3.10 `template<class T> T ** TNT::Array3D< T>::operator[] (int i) [inline]`
- 10.3.3.11 `template<class T> const T *const * TNT::Array3D< T>::operator[] (int i) const [inline]`
- 10.3.3.12 `template<class T> Array3D< T> & TNT::Array3D< T>::ref (const Array3D< T> & A) [inline]`
- 10.3.3.13 `template<class T> int TNT::Array3D< T>::ref_count () [inline]`
- 10.3.3.14 `template<class T> Array3D< T> TNT::Array3D< T>::subarray (int i0, int i1, int j0, int j1, int k0, int k1)`

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-_array3d.h](#)

10.4 JAMA::Cholesky< Real > Class Template Reference

```
#include <jama_cholesky.h>
```

Public Member Functions

- [Cholesky](#) ()
- [Cholesky](#) (const [Array2D](#)< Real > &A)
- [Array2D](#)< Real > [getL](#) () const
- [Array1D](#)< Real > [solve](#) (const [Array1D](#)< Real > &B)
- [Array2D](#)< Real > [solve](#) (const [Array2D](#)< Real > &B)
- int [is_spd](#) () const

10.4.1 Detailed Description

```
template<class Real>class JAMA::Cholesky< Real >
```

For a symmetric, positive definite matrix A, this function computes the [Cholesky](#) factorization, i.e. it computes a lower triangular matrix L such that $A = L * L^T$. If the matrix is not symmetric or positive definite, the function computes only a partial decomposition. This can be tested with the [is_spd\(\)](#) flag.

Typical usage looks like:

```
Array2D<double> A (n, n);
Array2D<double> L;
```

...

```
Cholesky<double> chol(A);

if (chol.is_spd())
    L = chol.getL();

else
    cout << "factorization was not complete.\n";
```

(Adapted from [JAMA](http://math.nist.gov/javanumerics/jama), a Java Matrix Library, developed by jointly by the Mathworks and NIST; see <http://math.nist.gov/javanumerics/jama>).

10.4.2 Constructor & Destructor Documentation

10.4.2.1 `template<class Real > JAMA::Cholesky< Real >::Cholesky ()`

10.4.2.2 `template<class Real > JAMA::Cholesky< Real >::Cholesky (const Array2D< Real > & A)`

Constructs a lower triangular matrix L, such that $L \cdot L' = A$. If A is not symmetric positive-definite (SPD), only a partial factorization is performed. If `is_spd()` evaluate true (1) then the factorization was successful.

10.4.3 Member Function Documentation

10.4.3.1 `template<class Real > Array2D< Real > JAMA::Cholesky< Real >::getL () const`

Returns

the lower triangular factor, L, such that $L \cdot L' = A$.

10.4.3.2 `template<class Real > int JAMA::Cholesky< Real >::is_spd () const`

Returns

1, if original matrix to be factored was symmetric positive-definite (SPD).

10.4.3.3 `template<class Real > Array1D< Real > JAMA::Cholesky< Real >::solve (const Array1D< Real > & b)`

Solve a linear system $A \cdot x = b$, using the previously computed cholesky factorization of A: $L \cdot L'$.

Parameters

<i>B</i>	A Matrix with as many rows as A and any number of columns.
----------	--

Returns

x so that $L \cdot L' \cdot x = b$. If b is nonconformat, or if A was not symmetric positive definite, a null (0x0) array is returned.

10.4.3.4 `template<class Real > Array2D< Real > JAMA::Cholesky< Real >::solve (const Array2D< Real > & B)`

Solve a linear system $A \cdot X = B$, using the previously computed cholesky factorization of A: $L \cdot L'$.

Parameters

B	A Matrix with as many rows as A and any number of columns.
-----	--

Returns

X so that $L * L' * X = B$. If B is nonconformat, or if A was not symmetric posidtive definite, a null (0x0) array is returned.

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

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-[_cholesky.h](#)

10.5 Class Class Reference

10.5.1 Detailed Description

reading the raw data from the device and delivering single well formed packet.

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Todo Meta data as QMap.

and metadata (such as counters, signal quality, gyro). Certain number of empty packets is constructed in the buffer and then continuously updated.

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR

PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Todo Filter should be a container for particular implementations of the filter, either static or adaptive.

temporal filter.

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

- /media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2datareadh

10.6 CRijndael Class Reference

```
#include <Rijndael.h>
```

Public Types

- enum { ECB =0, CBC =1, CFB =2 }

Public Member Functions

- `C Rijndael ()`
- `virtual ~C Rijndael ()`
- `void MakeKey (char const *key, char const *chain, int keylength=DEFAULT_BLOCK_SIZE, int blockSize=DEFAULT_BLOCK_SIZE)`
- `void EncryptBlock (char const *in, char *result)`
- `void DecryptBlock (char const *in, char *result)`
- `void Encrypt (char const *in, char *result, size_t n, int iMode=ECB)`
- `void Decrypt (char const *in, char *result, size_t n, int iMode=ECB)`
- `int GetKeyLength ()`
- `int GetBlockSize ()`
- `int GetRounds ()`
- `void ResetChain ()`

Static Public Attributes

- static char const * sm_chain0 = "\0"

10.6.1 Member Enumeration Documentation

10.6.1.1 anonymous enum

Enumerator

ECB

CBC

CFB

Public Member Functions

- [DtuArray2D](#) ()
- [DtuArray2D](#) (int *m*, int *n*)
- [DtuArray2D](#) (int *m*, int *n*, T **a*)
- [DtuArray2D](#) (int *m*, int *n*, const T &*a*)
- [DtuArray2D](#) (const TNT::Array2D< T > &*A*)
- [DtuArray2D](#) & [operator=](#) (const T &*a*)
- int [add](#) (const [DtuArray2D](#)< T > **B*, [DtuArray2D](#)< T > **out*)
- int [subtract](#) (const [DtuArray2D](#)< T > **B*, [DtuArray2D](#)< T > **out*)
- void [multiply](#) (double scalar, [DtuArray2D](#)< T > **out*)
- int [multiply](#) (const [DtuArray2D](#)< T > **B*, [DtuArray2D](#)< T > **out*)
- int [multiplyR](#) (const [DtuArray2D](#)< T > &*B*, [DtuArray2D](#)< T > &*out*)
- int [multiply](#) (const [DtuArray2D](#)< T > **B*, double scalar, [DtuArray2D](#)< T > **out*)
- void [transpose](#) ([DtuArray2D](#)< T > **A*)
- void [transpose](#) ([DtuArray2D](#)< T > &*A*)
- int [transpose_insitu](#) ()
- int [getSVD](#) (JAMA::SVD< T > &*A*)
- void [pinv](#) ([DtuArray2D](#)< T > **A*)
- T [trace](#) ()
- void [toIdentityMatrix](#) ()
- int [dim1](#) () const
- int [dim2](#) () const
- TNT::Array2D< T > [toTntArray2D](#) ()
- void [print](#) ()

Additional Inherited Members

10.7.1 Member Typedef Documentation

10.7.1.1 `template<class T> typedef T DTU::DtuArray2D< T >::value_type`

10.7.2 Constructor & Destructor Documentation

10.7.2.1 `template<class T> DTU::DtuArray2D< T >::DtuArray2D () [inline]`

10.7.2.2 `template<class T> DTU::DtuArray2D< T >::DtuArray2D (int m, int n) [inline]`

10.7.2.3 `template<class T> DTU::DtuArray2D< T >::DtuArray2D (int m, int n, T *a) [inline]`

10.7.2.4 `template<class T> DTU::DtuArray2D< T >::DtuArray2D (int m, int n, const T &a) [inline]`

10.7.2.5 `template<class T> DTU::DtuArray2D< T >::DtuArray2D (const TNT::Array2D< T > &A) [inline]`

10.7.3 Member Function Documentation

10.7.3.1 `template<class T> int DTU::DtuArray2D< T >::add (const DtuArray2D< T > *B, DtuArray2D< T > *out)`

basic maths

10.7.3.2 `template<class T> int DTU::DtuArray2D< T >::dim1 () const [inline]`

10.7.3.3 `template<class T> int DTU::DtuArray2D< T >::dim2 () const [inline]`

10.7.3.4 `template<class T> int DTU::DtuArray2D< T >::getSVD (JAMA::SVD< T > & A)`

10.7.3.5 `template<class T> void DTU::DtuArray2D< T >::multiply (double scalar, DtuArray2D< T > * out)`

10.7.3.6 `template<class T> int DTU::DtuArray2D< T >::multiply (const DtuArray2D< T > * B, DtuArray2D< T > * out)`

10.7.3.7 `template<class T> int DTU::DtuArray2D< T >::multiply (const DtuArray2D< T > * B, double constant, DtuArray2D< T > * out)`

see `int DtuArray2D<T>::multiply(const Array2D<T> &B, DtuArray2D<T> &out)` for alternative implementations of multiplication

10.7.3.8 `template<class T> int DTU::DtuArray2D< T >::multiplyR (const DtuArray2D< T > & B, DtuArray2D< T > & out)`

10.7.3.9 `template<class T> DtuArray2D< T > & DTU::DtuArray2D< T >::operator= (const T & a) [inline]`

10.7.3.10 `template<class T> void DTU::DtuArray2D< T >::pinv (DtuArray2D< T > * A)`

10.7.3.11 `template<class T> void DTU::DtuArray2D< T >::print () [inline]`

10.7.3.12 `template<class T> int DTU::DtuArray2D< T >::subtract (const DtuArray2D< T > * B, DtuArray2D< T > * out)`

10.7.3.13 `template<class T> void DTU::DtuArray2D< T >::toidentityMatrix () [inline]`

10.7.3.14 `template<class T> TNT::Array2D< T > DTU::DtuArray2D< T >::toTntArray2D () [inline]`

10.7.3.15 `template<class T> T DTU::DtuArray2D< T >::trace () [inline]`

10.7.3.16 `template<class T> void DTU::DtuArray2D< T >::transpose (DtuArray2D< T > * A)`

array operations

10.7.3.17 `template<class T> void DTU::DtuArray2D< T >::transpose (DtuArray2D< T > & A)`

10.7.3.18 `template<class T> int DTU::DtuArray2D< T >::transpose_insitu ()`

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

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/dtu_array_2d.h

10.8 JAMA::Eigenvalue< Real > Class Template Reference

```
#include <jama_eig.h>
```

Public Member Functions

- [Eigenvalue](#) (const TNT::Array2D< Real > &A)
- void [getV](#) (TNT::Array2D< Real > &V_)
- void [getRealEigenvalues](#) (TNT::Array1D< Real > &d_)
- void [getImagEigenvalues](#) (TNT::Array1D< Real > &e_)
- void [getD](#) (TNT::Array2D< Real > &D)

10.8.1 Detailed Description

template<class Real>class JAMA::Eigenvalue< Real >

Computes eigenvalues and eigenvectors of a real (non-complex) matrix.

If A is symmetric, then $A = V \cdot D \cdot V'$ where the eigenvalue matrix D is diagonal and the eigenvector matrix V is orthogonal. That is, the diagonal values of D are the eigenvalues, and $V \cdot V' = I$, where I is the identity matrix. The columns of V represent the eigenvectors in the sense that $A \cdot V = V \cdot D$.

If A is not symmetric, then the eigenvalue matrix D is block diagonal with the real eigenvalues in 1-by-1 blocks and any complex eigenvalues, $a + i \cdot b$, in 2-by-2 blocks, $\begin{bmatrix} a & b \\ -b & a \end{bmatrix}$. That is, if the complex eigenvalues look like

$u + iv$
.	$u - iv$
.	.	$a + ib$.	.	.
.	.	.	$a - ib$.	.
.	.	.	.	x	.
.	y

then D looks like

u	v
-v	u
.	.	a	b	.	.
.	.	-b	a	.	.
.	.	.	.	x	.
.	y

This keeps V a real matrix in both symmetric and non-symmetric cases, and $A \cdot V = V \cdot D$.

<p>

The matrix V may be badly conditioned, or even singular, so the validity of the equation $A = V \cdot D \cdot \text{inverse}(V)$ depends upon the condition number of V.

(Adapted from [JAMA](http://math.nist.gov/javanumerics/jama), a Java Matrix Library, developed by jointly by the Mathworks and NIST; see <http://math.nist.gov/javanumerics/jama>).

10.8.2 Constructor & Destructor Documentation

10.8.2.1 template<class Real > JAMA::Eigenvalue< Real >::Eigenvalue (const TNT::Array2D< Real > & A)
[inline]

Check for symmetry, then construct the eigenvalue decomposition

Parameters

A	Square real (non-complex) matrix
-----	----------------------------------

10.8.3 Member Function Documentation

10.8.3.1 `template<class Real > void JAMA::Eigenvalue< Real >::getD (TNT::Array2D< Real > & D) [inline]`

Computes the block diagonal eigenvalue matrix. If the original matrix A is not symmetric, then the eigenvalue matrix D is block diagonal with the real eigenvalues in 1-by-1 blocks and any complex eigenvalues, $a + i*b$, in 2-by-2 blocks, $[a, b; -b, a]$. That is, if the complex eigenvalues look like

$$\begin{array}{cccccc} u + iv & . & . & . & . & . \\ . & u - iv & . & . & . & . \\ . & . & a + ib & . & . & . \\ . & . & . & a - ib & . & . \\ . & . & . & . & x & . \\ . & . & . & . & . & y \end{array}$$

then D looks like

$$\begin{array}{cccccc} u & v & . & . & . & . \\ -v & u & . & . & . & . \\ . & . & a & b & . & . \\ . & . & -b & a & . & . \\ . & . & . & . & x & . \\ . & . & . & . & . & y \end{array}$$

This keeps V a real matrix in both symmetric and non-symmetric cases, and $A*V = V*D$.

Parameters

D	upon return, the matrix is filled with the block diagonal eigenvalue matrix.
-----	--

10.8.3.2 `template<class Real > void JAMA::Eigenvalue< Real >::getImagEigenvalues (TNT::Array1D< Real > & $e_$) [inline]`

Return the imaginary parts of the eigenvalues in parameter $e_$.

$e_$: new matrix with imaginary parts of the eigenvalues.

10.8.3.3 `template<class Real > void JAMA::Eigenvalue< Real >::getRealEigenvalues (TNT::Array1D< Real > & $d_$) [inline]`

Return the real parts of the eigenvalues

Returns

`real(diag(D))`

10.8.3.4 `template<class Real > void JAMA::Eigenvalue< Real >::getV (TNT::Array2D< Real > & V_) [inline]`

Return the eigenvector matrix

Returns

V

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-_eig.h](#)

10.9 FFTReal Class Reference

```
#include <FFTReal.h>
```

Public Types

- typedef double [flt_t](#)

Public Member Functions

- [FFTReal](#) (const long length)
- [~FFTReal](#) ()
- void [do_fft](#) (flt_t f[], const flt_t x[]) const
- void [do_ifft](#) (const flt_t f[], flt_t x[]) const
- void [rescale](#) (flt_t x[]) const

10.9.1 Member Typedef Documentation

10.9.1.1 typedef double [FFTReal::flt_t](#)

10.9.2 Constructor & Destructor Documentation

10.9.2.1 [FFTReal::FFTReal](#) (const long *length*) [explicit]

10.9.2.2 [FFTReal::~~FFTReal](#) (void)

10.9.3 Member Function Documentation

10.9.3.1 void [FFTReal::do_fft](#) (flt_t *f*[], const flt_t *x*[]) const

10.9.3.2 void [FFTReal::do_ifft](#) (const flt_t *f*[], flt_t *x*[]) const

10.9.3.3 void [FFTReal::rescale](#) (flt_t *x*[]) const

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/FFT-Real.h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/FFT-Real.cpp](#)

10.10 TNT::Fortran_Array1D< T > Class Template Reference

```
#include <tnt_fortran_array1d.h>
```

Public Types

- typedef T [value_type](#)

Public Member Functions

- [Fortran_Array1D](#) ()
- [Fortran_Array1D](#) (int n)
- [Fortran_Array1D](#) (int n, const T &a)
- [Fortran_Array1D](#) (int n, T *a)
- [Fortran_Array1D](#) (const [Fortran_Array1D](#) &A)
- [Fortran_Array1D](#) & [operator=](#) (const T &a)
- [Fortran_Array1D](#) & [operator=](#) (const [Fortran_Array1D](#) &A)
- [Fortran_Array1D](#) & [ref](#) (const [Fortran_Array1D](#) &A)
- [Fortran_Array1D](#) [copy](#) () const
- [Fortran_Array1D](#) & [inject](#) (const [Fortran_Array1D](#) &A)
- T & [operator\(\)](#) (int i)
- const T & [operator\(\)](#) (int i) const
- int [dim1](#) () const
- int [dim](#) () const
- [~Fortran_Array1D](#) ()
- int [ref_count](#) () const
- [Fortran_Array1D](#)< T > [subarray](#) (int i0, int i1)

10.10.1 Member Typedef Documentation

10.10.1.1 `template<class T> typedef T TNT::Fortran_Array1D< T >::value_type`

10.10.2 Constructor & Destructor Documentation

10.10.2.1 `template<class T> TNT::Fortran_Array1D< T >::Fortran_Array1D ()`

10.10.2.2 `template<class T> TNT::Fortran_Array1D< T >::Fortran_Array1D (int n) [explicit]`

10.10.2.3 `template<class T> TNT::Fortran_Array1D< T >::Fortran_Array1D (int n, const T & a)`

10.10.2.4 `template<class T> TNT::Fortran_Array1D< T >::Fortran_Array1D (int n, T * a)`

10.10.2.5 `template<class T> TNT::Fortran_Array1D< T >::Fortran_Array1D (const Fortran_Array1D< T > & A) [inline]`

10.10.2.6 `template<class T> TNT::Fortran_Array1D< T >::~~Fortran_Array1D ()`

10.10.3 Member Function Documentation

10.10.3.1 `template<class T> Fortran_Array1D< T > TNT::Fortran_Array1D< T >::copy () const`

10.10.3.2 `template<class T> int TNT::Fortran_Array1D< T >::dim () const [inline]`

- 10.10.3.3 `template<class T> int TNT::Fortran_Array1D< T >::dim1 () const [inline]`
- 10.10.3.4 `template<class T> Fortran_Array1D< T > & TNT::Fortran_Array1D< T >::inject (const Fortran_Array1D< T > & A)`
- 10.10.3.5 `template<class T> T & TNT::Fortran_Array1D< T >::operator() (int i) [inline]`
- 10.10.3.6 `template<class T> const T & TNT::Fortran_Array1D< T >::operator() (int i) const [inline]`
- 10.10.3.7 `template<class T> Fortran_Array1D< T > & TNT::Fortran_Array1D< T >::operator= (const T & a) [inline]`
- 10.10.3.8 `template<class T> Fortran_Array1D< T > & TNT::Fortran_Array1D< T >::operator= (const Fortran_Array1D< T > & A) [inline]`
- 10.10.3.9 `template<class T> Fortran_Array1D< T > & TNT::Fortran_Array1D< T >::ref (const Fortran_Array1D< T > & A) [inline]`
- 10.10.3.10 `template<class T> int TNT::Fortran_Array1D< T >::ref_count () const [inline]`
- 10.10.3.11 `template<class T> Fortran_Array1D< T > TNT::Fortran_Array1D< T >::subarray (int i0, int i1) [inline]`

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-_fortran_array1d.h](#)

10.11 TNT::Fortran_Array2D< T > Class Template Reference

```
#include <tnt_fortran_array2d.h>
```

Public Types

- typedef T [value_type](#)

Public Member Functions

- [Fortran_Array2D](#) ()
- [Fortran_Array2D](#) (int m, int n)
- [Fortran_Array2D](#) (int m, int n, T *a)
- [Fortran_Array2D](#) (int m, int n, const T &a)
- [Fortran_Array2D](#) (const [Fortran_Array2D](#) &A)
- [Fortran_Array2D](#) & [operator=](#) (const T &a)
- [Fortran_Array2D](#) & [operator=](#) (const [Fortran_Array2D](#) &A)
- [Fortran_Array2D](#) & [ref](#) (const [Fortran_Array2D](#) &A)
- [Fortran_Array2D](#) [copy](#) () const
- [Fortran_Array2D](#) & [inject](#) (const [Fortran_Array2D](#) &A)
- T & [operator\(\)](#) (int i, int j)
- const T & [operator\(\)](#) (int i, int j) const
- int [dim1](#) () const
- int [dim2](#) () const
- [~Fortran_Array2D](#) ()
- int [ref_count](#) () const

10.11.1 Member Typedef Documentation

10.11.1.1 `template<class T> typedef T TNT::Fortran_Array2D< T >::value_type`

10.11.2 Constructor & Destructor Documentation

10.11.2.1 `template<class T> TNT::Fortran_Array2D< T >::Fortran_Array2D ()`

10.11.2.2 `template<class T> TNT::Fortran_Array2D< T >::Fortran_Array2D (int m, int n)`

10.11.2.3 `template<class T> TNT::Fortran_Array2D< T >::Fortran_Array2D (int m, int n, T * a)`

10.11.2.4 `template<class T> TNT::Fortran_Array2D< T >::Fortran_Array2D (int m, int n, const T & a)`

10.11.2.5 `template<class T> TNT::Fortran_Array2D< T >::Fortran_Array2D (const Fortran_Array2D< T > & A) [inline]`

10.11.2.6 `template<class T> TNT::Fortran_Array2D< T >::~~Fortran_Array2D ()`

10.11.3 Member Function Documentation

10.11.3.1 `template<class T> Fortran_Array2D< T > TNT::Fortran_Array2D< T >::copy () const`

10.11.3.2 `template<class T> int TNT::Fortran_Array2D< T >::dim1 () const [inline]`

10.11.3.3 `template<class T> int TNT::Fortran_Array2D< T >::dim2 () const [inline]`

10.11.3.4 `template<class T> Fortran_Array2D< T > & TNT::Fortran_Array2D< T >::inject (const Fortran_Array2D< T > & A)`

10.11.3.5 `template<class T> T & TNT::Fortran_Array2D< T >::operator() (int i, int j) [inline]`

10.11.3.6 `template<class T> const T & TNT::Fortran_Array2D< T >::operator() (int i, int j) const [inline]`

10.11.3.7 `template<class T> Fortran_Array2D< T > & TNT::Fortran_Array2D< T >::operator= (const T & a) [inline]`

10.11.3.8 `template<class T> Fortran_Array2D< T > & TNT::Fortran_Array2D< T >::operator= (const Fortran_Array2D< T > & A) [inline]`

10.11.3.9 `template<class T> Fortran_Array2D< T > & TNT::Fortran_Array2D< T >::ref (const Fortran_Array2D< T > & A) [inline]`

10.11.3.10 `template<class T> int TNT::Fortran_Array2D< T >::ref_count () const [inline]`

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array2d.h](#)

10.12 TNT::Fortran_Array3D< T > Class Template Reference

```
#include <tnt_fortran_array3d.h>
```


Public Types

- typedef T [value_type](#)

Public Member Functions

- [Fortran_Array3D](#) ()
- [Fortran_Array3D](#) (int m, int n, int k)
- [Fortran_Array3D](#) (int m, int n, int k, T *a)
- [Fortran_Array3D](#) (int m, int n, int k, const T &a)
- [Fortran_Array3D](#) (const [Fortran_Array3D](#) &A)
- [Fortran_Array3D](#) & [operator=](#) (const T &a)
- [Fortran_Array3D](#) & [operator=](#) (const [Fortran_Array3D](#) &A)
- [Fortran_Array3D](#) & [ref](#) (const [Fortran_Array3D](#) &A)
- [Fortran_Array3D](#) [copy](#) () const
- [Fortran_Array3D](#) & [inject](#) (const [Fortran_Array3D](#) &A)
- T & [operator\(\)](#) (int i, int j, int k)
- const T & [operator\(\)](#) (int i, int j, int k) const
- int [dim1](#) () const
- int [dim2](#) () const
- int [dim3](#) () const
- int [ref_count](#) () const
- [~Fortran_Array3D](#) ()

10.12.1 Member Typedef Documentation

10.12.1.1 `template<class T> typedef T TNT::Fortran_Array3D< T >::value_type`

10.12.2 Constructor & Destructor Documentation

10.12.2.1 `template<class T> TNT::Fortran_Array3D< T >::Fortran_Array3D ()`

10.12.2.2 `template<class T> TNT::Fortran_Array3D< T >::Fortran_Array3D (int m, int n, int k)`

10.12.2.3 `template<class T> TNT::Fortran_Array3D< T >::Fortran_Array3D (int m, int n, int k, T * a)`

10.12.2.4 `template<class T> TNT::Fortran_Array3D< T >::Fortran_Array3D (int m, int n, int k, const T & a)`

10.12.2.5 `template<class T> TNT::Fortran_Array3D< T >::Fortran_Array3D (const Fortran_Array3D< T > & A) [inline]`

10.12.2.6 `template<class T> TNT::Fortran_Array3D< T >::~~Fortran_Array3D ()`

10.12.3 Member Function Documentation

10.12.3.1 `template<class T> Fortran_Array3D< T > TNT::Fortran_Array3D< T >::copy () const`

10.12.3.2 `template<class T> int TNT::Fortran_Array3D< T >::dim1 () const [inline]`

10.12.3.3 `template<class T> int TNT::Fortran_Array3D< T >::dim2 () const [inline]`

10.12.3.4 `template<class T> int TNT::Fortran_Array3D< T >::dim3 () const [inline]`

- 10.12.3.5 `template<class T> Fortran_Array3D< T> & TNT::Fortran_Array3D< T>::inject (const Fortran_Array3D< T> & A)`
- 10.12.3.6 `template<class T> T & TNT::Fortran_Array3D< T>::operator() (int i, int j, int k) [inline]`
- 10.12.3.7 `template<class T> const T & TNT::Fortran_Array3D< T>::operator() (int i, int j, int k) const [inline]`
- 10.12.3.8 `template<class T> Fortran_Array3D< T> & TNT::Fortran_Array3D< T>::operator= (const T & a) [inline]`
- 10.12.3.9 `template<class T> Fortran_Array3D< T> & TNT::Fortran_Array3D< T>::operator= (const Fortran_Array3D< T> & A) [inline]`
- 10.12.3.10 `template<class T> Fortran_Array3D< T> & TNT::Fortran_Array3D< T>::ref (const Fortran_Array3D< T> & A) [inline]`
- 10.12.3.11 `template<class T> int TNT::Fortran_Array3D< T>::ref_count () const [inline]`

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

- /media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-_fortran_array3d.h

10.13 hid_device_ Struct Reference

Public Attributes

- int [device_handle](#)
- int [blocking](#)
- int [uses_numbered_reports](#)
- IOHIDDeviceRef [device_handle](#)
- int [disconnected](#)
- CFStringRef [run_loop_mode](#)
- CFRunLoopRef [run_loop](#)
- CFRunLoopSourceRef [source](#)
- uint8_t * [input_report_buf](#)
- CFIndex [max_input_report_len](#)
- struct [input_report](#) * [input_reports](#)
- pthread_t [thread](#)
- pthread_mutex_t [mutex](#)
- pthread_cond_t [condition](#)
- pthread_barrier_t [barrier](#)
- pthread_barrier_t [shutdown_barrier](#)
- int [shutdown_thread](#)
- [hid_device](#) * [next](#)

10.13.1 Member Data Documentation

- 10.13.1.1 `pthread_barrier_t hid_device_::barrier`
- 10.13.1.2 `int hid_device_::blocking`
- 10.13.1.3 `pthread_cond_t hid_device_::condition`

- 10.13.1.4 int hid_device::device_handle
- 10.13.1.5 IOHIDDeviceRef hid_device::device_handle
- 10.13.1.6 int hid_device::disconnected
- 10.13.1.7 uint8_t* hid_device::input_report_buf
- 10.13.1.8 struct input_report* hid_device::input_reports
- 10.13.1.9 CFIndex hid_device::max_input_report_len
- 10.13.1.10 pthread_mutex_t hid_device::mutex
- 10.13.1.11 hid_device* hid_device::next
- 10.13.1.12 CFRunLoopRef hid_device::run_loop
- 10.13.1.13 CFStringRef hid_device::run_loop_mode
- 10.13.1.14 pthread_barrier_t hid_device::shutdown_barrier
- 10.13.1.15 int hid_device::shutdown_thread
- 10.13.1.16 CFRunLoopSourceRef hid_device::source
- 10.13.1.17 pthread_t hid_device::thread
- 10.13.1.18 int hid_device::uses_numbered_reports

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

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/linux/hid-c](#)

10.14 hid_device_info Struct Reference

```
#include <hidapi.h>
```

Public Attributes

- char * [path](#)
- unsigned short [vendor_id](#)
- unsigned short [product_id](#)
- wchar_t * [serial_number](#)
- unsigned short [release_number](#)
- wchar_t * [manufacturer_string](#)
- wchar_t * [product_string](#)
- unsigned short [usage_page](#)
- unsigned short [usage](#)
- int [interface_number](#)
- struct [hid_device_info](#) * [next](#)

10.14.1 Detailed Description

hidapi info structure

10.14.2 Member Data Documentation

10.14.2.1 `int hid_device_info::interface_number`

The USB interface which this logical device represents. Valid on both Linux implementations in all cases, and valid on the Windows implementation only if the device contains more than one interface.

10.14.2.2 `wchar_t * hid_device_info::manufacturer_string`

Manufacturer String

10.14.2.3 `struct hid_device_info * hid_device_info::next`

Pointer to the next device

10.14.2.4 `char * hid_device_info::path`

Platform-specific device path

10.14.2.5 `unsigned short hid_device_info::product_id`

Device Product ID

10.14.2.6 `wchar_t * hid_device_info::product_string`

Product string

10.14.2.7 `unsigned short hid_device_info::release_number`

Device Release Number in binary-coded decimal, also known as Device Version Number

10.14.2.8 `wchar_t * hid_device_info::serial_number`

Serial Number

10.14.2.9 `unsigned short hid_device_info::usage`

Usage for this Device/Interface (Windows/Mac only).

10.14.2.10 `unsigned short hid_device_info::usage_page`

Usage Page for this Device/Interface (Windows/Mac only).

10.14.2.11 unsigned short hid_device_info::vendor_id

Device Vendor ID

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

- [/media/philipjh/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/linux/hidapi.h](#)

10.15 TNT::i_refvec< T > Class Template Reference

```
#include <tnt_i_refvec.h>
```

Public Member Functions

- [i_refvec](#) ()
- [i_refvec](#) (int n)
- [i_refvec](#) (T *data)
- [i_refvec](#) (const [i_refvec](#) &v)
- T * [begin](#) ()
- const T * [begin](#) () const
- T & [operator\[\]](#) (int i)
- const T & [operator\[\]](#) (int i) const
- [i_refvec](#)< T > & [operator=](#) (const [i_refvec](#)< T > &V)
- void [copy_](#) (T *p, const T *q, const T *e)
- void [set_](#) (T *p, const T *b, const T *e)
- int [ref_count](#) () const
- int [is_null](#) () const
- void [destroy](#) ()
- [~i_refvec](#) ()

10.15.1 Constructor & Destructor Documentation

10.15.1.1 `template<class T> TNT::i_refvec< T >::i_refvec ()`

10.15.1.2 `template<class T> TNT::i_refvec< T >::i_refvec (int n) [explicit]`

In case n is 0 or negative, it does NOT call new.

10.15.1.3 `template<class T> TNT::i_refvec< T >::i_refvec (T * data) [inline]`

10.15.1.4 `template<class T> TNT::i_refvec< T >::i_refvec (const i_refvec< T > & v) [inline]`

10.15.1.5 `template<class T> TNT::i_refvec< T >::~~i_refvec ()`

10.15.2 Member Function Documentation

10.15.2.1 `template<class T> T * TNT::i_refvec< T >::begin () [inline]`

10.15.2.2 `template<class T> const T * TNT::i_refvec< T >::begin () const [inline]`

10.15.2.3 `template<class T> void TNT::i_refvec< T >::copy_ (T * p, const T * q, const T * e)`

```

10.15.2.4  template<class T> void TNT::i_refvec<T>::destroy( ) [inline]

10.15.2.5  template<class T> int TNT::i_refvec<T>::is_null( ) const [inline]

10.15.2.6  template<class T> i_refvec<T> & TNT::i_refvec<T>::operator=( const i_refvec<T> & V )
           [inline]

10.15.2.7  template<class T> T & TNT::i_refvec<T>::operator[]( int i ) [inline]

10.15.2.8  template<class T> const T & TNT::i_refvec<T>::operator[]( int i ) const [inline]

10.15.2.9  template<class T> int TNT::i_refvec<T>::ref_count( ) const [inline]

10.15.2.10 template<class T> void TNT::i_refvec<T>::set( T * p, const T * b, const T * e )

```

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

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-_i_refvec.h](#)

10.16 input_report Struct Reference

Public Attributes

- `uint8_t * data`
- `size_t len`
- `struct input_report * next`

10.16.1 Member Data Documentation

```

10.16.1.1  uint8_t* input_report::data

10.16.1.2  size_t input_report::len

10.16.1.3  struct input_report* input_report::next

```

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

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/osx/hid.-c](#)

10.17 JAMA::LU< Real > Class Template Reference

```
#include <jama_lu.h>
```

Public Member Functions

- `LU (const Array2D< Real > &A)`
- `int isNonsingular ()`
- `Array2D< Real > getL ()`
- `Array2D< Real > getU ()`
- `Array1D< int > getPivot ()`

- Real [det](#) ()
- [Array2D](#)< Real > [solve](#) (const [Array2D](#)< Real > &B)
- [Array1D](#)< Real > [solve](#) (const [Array1D](#)< Real > &b)

10.17.1 Detailed Description

`template<class Real>class JAMA::LU< Real >`

[LU](#) Decomposition.

For an m-by-n matrix A with $m \geq n$, the [LU](#) decomposition is an m-by-n unit lower triangular matrix L, an n-by-n upper triangular matrix U, and a permutation vector piv of length m so that $A(\text{piv},:) = L*U$. If $m < n$, then L is m-by-m and U is m-by-n.

The [LU](#) decomposition with pivoting always exists, even if the matrix is singular, so the constructor will never fail. The primary use of the [LU](#) decomposition is in the solution of square systems of simultaneous linear equations. This will fail if [isNonsingular\(\)](#) returns false.

10.17.2 Constructor & Destructor Documentation

10.17.2.1 `template<class Real > JAMA::LU< Real >::LU (const Array2D< Real > &A) [inline]`

[LU](#) Decomposition

Parameters

A	Rectangular matrix
---	--------------------

Returns

[LU](#) Decomposition object to access L, U and piv.

10.17.3 Member Function Documentation

10.17.3.1 `template<class Real > Real JAMA::LU< Real >::det () [inline]`

Compute determinant using [LU](#) factors.

Returns

determinant of A, or 0 if A is not square.

10.17.3.2 `template<class Real > Array2D<Real> JAMA::LU< Real >::getL () [inline]`

Return lower triangular factor

Returns

L

10.17.3.3 `template<class Real > Array1D<int> JAMA::LU< Real >::getPivot () [inline]`

Return pivot permutation vector

Returns

piv

10.17.3.4 `template<class Real> Array2D<Real> JAMA::LU<Real>::getU () [inline]`

Return upper triangular factor

Returns

U portion of LU factorization.

10.17.3.5 `template<class Real> int JAMA::LU<Real>::isNonsingular () [inline]`

Is the matrix nonsingular?

Returns

1 (true) if upper triangular factor U (and hence A) is nonsingular, 0 otherwise.

10.17.3.6 `template<class Real> Array2D<Real> JAMA::LU<Real>::solve (const Array2D<Real> & B) [inline]`

Solve $A \cdot X = B$

Parameters

B	A Matrix with as many rows as A and any number of columns.
-----	--

Returns

X so that $L \cdot U \cdot X = B(\text{piv}, :)$, if B is nonconformant, returns 0x0 (null) array.

10.17.3.7 `template<class Real> Array1D<Real> JAMA::LU<Real>::solve (const Array1D<Real> & b) [inline]`

Solve $A \cdot x = b$, where x and b are vectors of length equal to the number of rows in A.

Parameters

b	a vector (Array1D) of length equal to the first dimension of A.
-----	---

Returns

x a vector (Array1D) so that $L \cdot U \cdot x = b(\text{piv})$, if B is nonconformant, returns 0x0 (null) array.

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

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-lu.h](#)

10.18 TNT::Matrix< T > Class Template Reference

```
#include <tnt_cmat.h>
```


Public Types

- typedef [Subscript](#) [size_type](#)
- typedef T [value_type](#)
- typedef T [element_type](#)
- typedef T * [pointer](#)
- typedef T * [iterator](#)
- typedef T & [reference](#)
- typedef const T * [const_iterator](#)
- typedef const T & [const_reference](#)

Public Member Functions

- [Subscript](#) [lbound](#) () const
- [operator T **](#) ()
- [operator T **](#) () const
- [Subscript](#) [size](#) () const
- [Matrix](#) ()
- [Matrix](#) (const [Matrix](#)< T > &A)
- [Matrix](#) ([Subscript](#) M, [Subscript](#) N, const T &value=T())
- [Matrix](#) ([Subscript](#) M, [Subscript](#) N, const T *v)
- [Matrix](#) ([Subscript](#) M, [Subscript](#) N, const char *s)
- [~Matrix](#) ()
- [Matrix](#)< T > & [newsize](#) ([Subscript](#) M, [Subscript](#) N)
- [Matrix](#)< T > & [operator=](#) (const [Matrix](#)< T > &A)
- [Matrix](#)< T > & [operator=](#) (const T &scalar)
- [Subscript](#) [dim](#) ([Subscript](#) d) const
- [Subscript](#) [num_rows](#) () const
- [Subscript](#) [num_cols](#) () const
- T * [operator\[\]](#) ([Subscript](#) i)
- const T * [operator\[\]](#) ([Subscript](#) i) const
- [reference](#) [operator\(\)](#) ([Subscript](#) i)
- [const_reference](#) [operator\(\)](#) ([Subscript](#) i) const
- [reference](#) [operator\(\)](#) ([Subscript](#) i, [Subscript](#) j)
- [const_reference](#) [operator\(\)](#) ([Subscript](#) i, [Subscript](#) j) const

Protected Member Functions

- void [initialize](#) ([Subscript](#) M, [Subscript](#) N)
- void [copy](#) (const T *v)
- void [set](#) (const T &val)
- void [destroy](#) ()

Protected Attributes

- [Subscript](#) [m_](#)
- [Subscript](#) [n_](#)
- [Subscript](#) [mn_](#)
- T * [v_](#)
- T ** [row_](#)
- T * [vm1_](#)
- T ** [rowm1_](#)

10.18.1 Member Typedef Documentation

10.18.1.1 `template<class T> typedef const T* TNT::Matrix< T >::const_iterator`

10.18.1.2 `template<class T> typedef const T& TNT::Matrix< T >::const_reference`

10.18.1.3 `template<class T> typedef T TNT::Matrix< T >::element_type`

10.18.1.4 `template<class T> typedef T* TNT::Matrix< T >::iterator`

10.18.1.5 `template<class T> typedef T* TNT::Matrix< T >::pointer`

10.18.1.6 `template<class T> typedef T& TNT::Matrix< T >::reference`

10.18.1.7 `template<class T> typedef Subscript TNT::Matrix< T >::size_type`

10.18.1.8 `template<class T> typedef T TNT::Matrix< T >::value_type`

10.18.2 Constructor & Destructor Documentation

10.18.2.1 `template<class T> TNT::Matrix< T >::Matrix () [inline]`

10.18.2.2 `template<class T> TNT::Matrix< T >::Matrix (const Matrix< T > & A) [inline]`

10.18.2.3 `template<class T> TNT::Matrix< T >::Matrix (Subscript M, Subscript N, const T & value = T ()) [inline]`

10.18.2.4 `template<class T> TNT::Matrix< T >::Matrix (Subscript M, Subscript N, const T * v) [inline]`

10.18.2.5 `template<class T> TNT::Matrix< T >::Matrix (Subscript M, Subscript N, const char * s) [inline]`

10.18.2.6 `template<class T> TNT::Matrix< T >::~~Matrix () [inline]`

10.18.3 Member Function Documentation

10.18.3.1 `template<class T> void TNT::Matrix< T >::copy (const T * v) [inline],[protected]`

10.18.3.2 `template<class T> void TNT::Matrix< T >::destroy () [inline],[protected]`

10.18.3.3 `template<class T> Subscript TNT::Matrix< T >::dim (Subscript d) const [inline]`

10.18.3.4 `template<class T> void TNT::Matrix< T >::initialize (Subscript M, Subscript N) [inline],[protected]`

10.18.3.5 `template<class T> Subscript TNT::Matrix< T >::lbound () const [inline]`

10.18.3.6 `template<class T> Matrix<T>& TNT::Matrix< T >::newsize (Subscript M, Subscript N) [inline]`

10.18.3.7 `template<class T> Subscript TNT::Matrix< T >::num_cols () const [inline]`

10.18.3.8 `template<class T> Subscript TNT::Matrix< T >::num_rows () const [inline]`

10.18.3.9 `template<class T> TNT::Matrix< T >::operator T ** () [inline]`

10.18.3.10 `template<class T> TNT::Matrix< T >::operator T ** () const [inline]`

- 10.18.3.11 `template<class T> reference TNT::Matrix< T >::operator()(Subscript i) [inline]`
- 10.18.3.12 `template<class T> const_reference TNT::Matrix< T >::operator()(Subscript i) const [inline]`
- 10.18.3.13 `template<class T> reference TNT::Matrix< T >::operator()(Subscript i, Subscript j) [inline]`
- 10.18.3.14 `template<class T> const_reference TNT::Matrix< T >::operator()(Subscript i, Subscript j) const [inline]`
- 10.18.3.15 `template<class T> Matrix<T>& TNT::Matrix< T >::operator= (const Matrix< T > & A) [inline]`
- 10.18.3.16 `template<class T> Matrix<T>& TNT::Matrix< T >::operator= (const T & scalar) [inline]`
- 10.18.3.17 `template<class T> T* TNT::Matrix< T >::operator[](Subscript i) [inline]`
- 10.18.3.18 `template<class T> const T* TNT::Matrix< T >::operator[](Subscript i) const [inline]`
- 10.18.3.19 `template<class T> void TNT::Matrix< T >::set (const T & val) [inline], [protected]`
- 10.18.3.20 `template<class T> Subscript TNT::Matrix< T >::size () const [inline]`

10.18.4 Member Data Documentation

- 10.18.4.1 `template<class T> Subscript TNT::Matrix< T >::m_ [protected]`
- 10.18.4.2 `template<class T> Subscript TNT::Matrix< T >::mn_ [protected]`
- 10.18.4.3 `template<class T> Subscript TNT::Matrix< T >::n_ [protected]`
- 10.18.4.4 `template<class T> T** TNT::Matrix< T >::row_ [protected]`
- 10.18.4.5 `template<class T> T** TNT::Matrix< T >::rowm1_ [protected]`
- 10.18.4.6 `template<class T> T* TNT::Matrix< T >::v_ [protected]`
- 10.18.4.7 `template<class T> T* TNT::Matrix< T >::vm1_ [protected]`

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-_cmat.h](#)

10.19 pthread_barrier Struct Reference

Public Attributes

- `pthread_mutex_t` [mutex](#)
- `pthread_cond_t` [cond](#)
- `int` [count](#)
- `int` [trip_count](#)

10.19.1 Member Data Documentation

10.19.1.1 `pthread_cond_t pthread_barrier::cond`

10.19.1.2 `int pthread_barrier::count`

10.19.1.3 `pthread_mutex_t pthread_barrier::mutex`

10.19.1.4 `int pthread_barrier::trip_count`

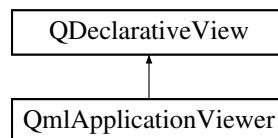
The documentation for this struct was generated from the following file:

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/osx/hid.-c](#)

10.20 QmlApplicationViewer Class Reference

```
#include <qmlapplicationviewer.h>
```

Inheritance diagram for QmlApplicationViewer:



Public Types

- enum `ScreenOrientation` { `ScreenOrientationLockPortrait`, `ScreenOrientationLockLandscape`, `ScreenOrientationAuto` }

Public Member Functions

- `QmlApplicationViewer` (`QWidget *parent=0`)
- virtual `~QmlApplicationViewer` ()
- void `setMainQmlFile` (`const QString &file`)
- void `addImportPath` (`const QString &path`)
- void `setOrientation` (`ScreenOrientation orientation`)
- void `showExpanded` ()

Static Public Member Functions

- static `QmlApplicationViewer * create` ()

10.20.1 Member Enumeration Documentation

10.20.1.1 enum `QmlApplicationViewer::ScreenOrientation`

Enumerator

`ScreenOrientationLockPortrait`
`ScreenOrientationLockLandscape`
`ScreenOrientationAuto`

10.20.2 Constructor & Destructor Documentation

10.20.2.1 `QmlApplicationViewer::QmlApplicationViewer (QWidget * parent = 0)` `[explicit]`

10.20.2.2 `QmlApplicationViewer::~~QmlApplicationViewer ()` `[virtual]`

10.20.3 Member Function Documentation

10.20.3.1 `void QmlApplicationViewer::addImportPath (const QString & path)`

10.20.3.2 `QmlApplicationViewer * QmlApplicationViewer::create ()` `[static]`

10.20.3.3 `void QmlApplicationViewer::setMainQmlFile (const QString & file)`

10.20.3.4 `void QmlApplicationViewer::setOrientation (ScreenOrientation orientation)`

10.20.3.5 `void QmlApplicationViewer::showExpanded ()`

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/qmlapplicationviewer/qmlapplicationviewer.h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/qmlapplicationviewer/qmlapplicationviewer.cpp](#)

10.21 QmlApplicationViewerPrivate Class Reference

Friends

- class [QmlApplicationViewer](#)

10.21.1 Friends And Related Function Documentation

10.21.1.1 `friend class QmlApplicationViewer` `[friend]`

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/qmlapplicationviewer/qmlapplicationviewer.cpp](#)

10.22 JAMA::QR< Real > Class Template Reference

```
#include <jama_qr.h>
```

Public Member Functions

- [QR](#) (const [TNT::Array2D](#)< Real > &A)
- int [isFullRank](#) () const
- [TNT::Array2D](#)< Real > [getHouseholder](#) (void) const
- [TNT::Array2D](#)< Real > [getR](#) () const
- [TNT::Array2D](#)< Real > [getQ](#) () const
- [TNT::Array1D](#)< Real > [solve](#) (const [TNT::Array1D](#)< Real > &b) const
- [TNT::Array2D](#)< Real > [solve](#) (const [TNT::Array2D](#)< Real > &B) const

10.22.1 Detailed Description

```
template<class Real>class JAMA::QR< Real >
```

Classical **QR** Decomposition: for an m-by-n matrix A with $m \geq n$, the **QR** decomposition is an m-by-n orthogonal matrix Q and an n-by-n upper triangular matrix R so that $A = Q \cdot R$.

The **QR** decomposition always exists, even if the matrix does not have full rank, so the constructor will never fail. The primary use of the **QR** decomposition is in the least squares solution of nonsquare systems of simultaneous linear equations. This will fail if `isFullRank()` returns 0 (false).

The Q and R factors can be retrieved via the `getQ()` and `getR()` methods. Furthermore, a `solve()` method is provided to find the least squares solution of $Ax=b$ using the **QR** factors.

(Adapted from **JAMA**, a Java Matrix Library, developed by jointly by the Mathworks and NIST; see <http://math.nist.gov/javanumerics/jama>).

10.22.2 Constructor & Destructor Documentation

```
10.22.2.1 template<class Real > JAMA::QR< Real >::QR ( const TNT::Array2D< Real > & A ) [inline]
```

Create a **QR** factorization object for A.

Parameters

A	rectangular ($m \geq n$) matrix.
---	------------------------------------

10.22.3 Member Function Documentation

```
10.22.3.1 template<class Real > TNT::Array2D<Real> JAMA::QR< Real >::getHouseholder ( void ) const [inline]
```

Retrieve the Householder vectors from **QR** factorization

Returns

lower trapezoidal matrix whose columns define the reflections

```
10.22.3.2 template<class Real > TNT::Array2D<Real> JAMA::QR< Real >::getQ ( ) const [inline]
```

Generate and return the (economy-sized) orthogonal factor

Parameters

Q	the (economy-sized) orthogonal factor ($Q \cdot R = A$).
---	--

```
10.22.3.3 template<class Real > TNT::Array2D<Real> JAMA::QR< Real >::getR ( ) const [inline]
```

Return the upper triangular factor, R, of the **QR** factorization

Returns

R

10.22.3.4 `template<class Real> int JAMA::QR<Real>::isFullRank () const [inline]`

Flag to denote the matrix is of full rank.

Returns

1 if matrix is full rank, 0 otherwise.

10.22.3.5 `template<class Real> TNT::Array1D<Real> JAMA::QR<Real>::solve (const TNT::Array1D<Real> & b) const [inline]`

Least squares solution of $A \cdot x = b$

Parameters

B	m-length array (vector).
-----	--------------------------

Returns

x n-length array (vector) that minimizes the two norm of $Q \cdot R \cdot X - B$. If B is non-conformant, or if [QR.isFullRank\(\)](#) is false, the routine returns a null (0-length) vector.

10.22.3.6 `template<class Real> TNT::Array2D<Real> JAMA::QR<Real>::solve (const TNT::Array2D<Real> & B) const [inline]`

Least squares solution of $A \cdot X = B$

Parameters

B	m x k Array (must conform).
-----	-----------------------------

Returns

X n x k Array that minimizes the two norm of $Q \cdot R \cdot X - B$. If B is non-conformant, or if [QR.isFullRank\(\)](#) is false, the routine returns a null (0x0) array.

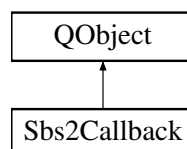
The documentation for this class was generated from the following file:

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-qr.h](#)

10.23 Sbs2Callback Class Reference

```
#include <sbs2callback.h>
```

Inheritance diagram for Sbs2Callback:



Public Slots

- void [startRecording](#) (QString user, QString description)
- void [stopRecording](#) ()
- void [insertIntoMetaFile](#) (QString event)
- void [turnFilterOn](#) (int fbandLow_, int fbandHigh_, int filterOrder_)
- void [turnFilterOff](#) ()
- void [turnChannelSpectrogramOn](#) (int spectrogramChannelSamples_=128, int spectrogramChannelLength_=128, int spectrogramChannelDelta_=0)
- void [turnChannelSpectrogramOff](#) ()
- void [setWindowType](#) ([Sbs2Spectrogram::WindowType](#) windowType)
- void [setWindowType](#) (int windowType)
- void [turnOnSourceReconstructionLoreta](#) (int sourceReconstructionSamples_, int sourceReconstructionDelta_, int sourceReconstructionModelUpdateLength_, int sourceReconstructionModelUpdateDelta_, QString hardware_"emotiv")
- void [turnOnSourceReconstructioSparse](#) (int sourceReconstructionSamples, QVector< double > lambdas, QString hardware="emotiv")
- void [spectrogramUpdatedSlot](#) ()
- void [setHardware](#) (QString hardware)
- void [turnSendRawDataOn](#) (QString rawDataServerAddress_, int rawDataPort_, int rawDataSize_=32, int rawDataQueueLength_=8)
- void [turnSendRawDataOff](#) ()
- void [addRawDataHost](#) (QString address, int port)
- void [removeRawDataHost](#) (QString address, int port)
- void [sendMessage](#) (QString message, QString address, int port)
- void [sendMessage](#) (QString message)
- void [addMessageUdpOutputHost](#) (QString address, int port)
- void [removeMessageUdpOutputHost](#) (QString address)
- void [clearMessageUdpOutputHosts](#) ()
- void [turnReceiveMessageOn](#) (QString address, int port)
- void [turnReceiveMessageOff](#) ()
- void [readMessage](#) (QString data, QString sender, int senderPort)
- void [getNetworkAddresses](#) ()
- void [deviceFound](#) (QMap< QString, QVariant > params_)

Signals

- void [timeTick10](#) ()
- void [timeTick2](#) ()
- void [timeTick4](#) ()
- void [timeTick8](#) ()
- void [timeTick0](#) ()
- void [timeTick16](#) ()
- void [setWindowTypeSignal](#) ([Sbs2Spectrogram::WindowType](#) windowType)
- void [batteryValue](#) (QVariant value)
- void [cqValues](#) (QVariant channel, QVariant value)
- void [cqValue](#) (QString channel, double value)
- void [spectrogramUpdated](#) ()
- void [udpMessageReceived](#) (QString data, QString sender, int port)
- void [networkAddresses](#) (QVariant data)
- void [hardwareChanged](#) (QString hardware)
- void [deviceFoundSignal](#) (QMap< QString, QVariant > [params](#))

Public Member Functions

- [Sbs2Callback](#) (QObject *parent=0)
- virtual void [getData](#) (Sbs2Packet *packet)
- QString [getRawFilename](#) ()

Static Public Member Functions

- static int [getCurrentPacketCounter](#) ()
- static int [getCurrentPacket](#) ()

Protected Member Functions

- void [setPacket](#) (Sbs2Packet *packet)
- void [setSbs2DataHandler](#) (Sbs2DataHandler *sbs2DataHandler_)

Protected Attributes

- Sbs2DataHandler * [sbs2DataHandler](#)
- int [samplesCollected](#)
- Sbs2Packet * [thisPacket](#)
- Sbs2Region * [sbs2Region](#)
- int [isRecording](#)
- QMap< QString, QVariant > [params](#)
- int [devicePresent](#)

Static Protected Attributes

- static int [currentPacketCounter](#) = 0
- static int [currentPacket](#) = 0

10.23.1 Constructor & Destructor Documentation

10.23.1.1 [Sbs2Callback::Sbs2Callback](#) (QObject * *parent* = 0) [explicit]

10.23.2 Member Function Documentation

10.23.2.1 void [Sbs2Callback::addMessageUdpOutputHost](#) (QString *address*, int *port*) [slot]

10.23.2.2 void [Sbs2Callback::addRawDataHost](#) (QString *address*, int *port*) [slot]

10.23.2.3 void [Sbs2Callback::batteryValue](#) (QVariant *value*) [signal]

10.23.2.4 void [Sbs2Callback::clearMessageUdpOutputHosts](#) () [slot]

10.23.2.5 void [Sbs2Callback::cqValue](#) (QString *channel*, double *value*) [signal]

10.23.2.6 void [Sbs2Callback::cqValues](#) (QVariant *channel*, QVariant *value*) [signal]

10.23.2.7 void [Sbs2Callback::deviceFound](#) (QMap< QString, QVariant > *params_*) [slot]

10.23.2.8 void [Sbs2Callback::deviceFoundSignal](#) (QMap< QString, QVariant > *params*) [signal]

- 10.23.2.9 `int Sbs2Callback::getCurrentPacket () [static]`
- 10.23.2.10 `int Sbs2Callback::getCurrentPacketCounter () [static]`
- 10.23.2.11 `virtual void Sbs2Callback::getData (Sbs2Packet * packet) [inline],[virtual]`
- 10.23.2.12 `void Sbs2Callback::getNetworkAddresses () [slot]`
- 10.23.2.13 `QString Sbs2Callback::getRawFilename ()`
- 10.23.2.14 `void Sbs2Callback::hardwareChanged (QString hardware) [signal]`
- 10.23.2.15 `void Sbs2Callback::insertIntoMetaFile (QString event) [slot]`
- 10.23.2.16 `void Sbs2Callback::networkAddresses (QVariant data) [signal]`
- 10.23.2.17 `void Sbs2Callback::readMessage (QString data, QString sender, int senderPort) [slot]`
- 10.23.2.18 `void Sbs2Callback::removeMessageUdpOutputHost (QString address) [slot]`
- 10.23.2.19 `void Sbs2Callback::removeRawDataHost (QString address, int port) [slot]`
- 10.23.2.20 `void Sbs2Callback::sendMessage (QString message, QString address, int port) [slot]`
- 10.23.2.21 `void Sbs2Callback::sendMessage (QString message) [slot]`
- 10.23.2.22 `void Sbs2Callback::setHardware (QString hardware) [slot]`
- 10.23.2.23 `void Sbs2Callback::setPacket (Sbs2Packet * packet) [protected]`
- 10.23.2.24 `void Sbs2Callback::setSbs2DataHandler (Sbs2DataHandler * sbs2DataHandler_) [protected]`

used for setting custom data handlers

- 10.23.2.25 `void Sbs2Callback::setWindowType (Sbs2Spectrogram::WindowType windowType) [slot]`
- 10.23.2.26 `void Sbs2Callback::setWindowType (int windowType) [slot]`
- 10.23.2.27 `void Sbs2Callback::setWindowTypeSignal (Sbs2Spectrogram::WindowType windowType) [signal]`
- 10.23.2.28 `void Sbs2Callback::spectrogramUpdated () [signal]`
- 10.23.2.29 `void Sbs2Callback::spectrogramUpdatedSlot () [slot]`
- 10.23.2.30 `void Sbs2Callback::startRecording (QString user, QString description) [slot]`
- 10.23.2.31 `void Sbs2Callback::stopRecording () [slot]`
- 10.23.2.32 `void Sbs2Callback::timeTick0 () [signal]`
- 10.23.2.33 `void Sbs2Callback::timeTick10 () [signal]`
- 10.23.2.34 `void Sbs2Callback::timeTick16 () [signal]`
- 10.23.2.35 `void Sbs2Callback::timeTick2 () [signal]`

- 10.23.2.36 void Sbs2Callback::timeTick4 () [signal]
- 10.23.2.37 void Sbs2Callback::timeTick8 () [signal]
- 10.23.2.38 void Sbs2Callback::turnChannelSpectrogramOff () [slot]
- 10.23.2.39 void Sbs2Callback::turnChannelSpectrogramOn (int *spectrogramChannelSamples_* = 128, int *spectrogramChannelLength_* = 128, int *spectrogramChannelDelta_* = 0) [slot]
- 10.23.2.40 void Sbs2Callback::turnFilterOff () [slot]
- 10.23.2.41 void Sbs2Callback::turnFilterOn (int *fbandLow_*, int *fbandHigh_*, int *filterOrder_*) [slot]
- 10.23.2.42 void Sbs2Callback::turnOnSourceReconstructionLoreta (int *sourceReconstructionSamples_*, int *sourceReconstructionDelta_*, int *sourceReconstructionModelUpdateLength_*, int *sourceReconstructionModelUpdateDelta_*, QString *hardware_* = "emotiv") [slot]
- 10.23.2.43 void Sbs2Callback::turnOnSourceReconstructioSparse (int *sourceReconstructionSamples*, QVector< double > *lambdas*, QString *hardware* = "emotiv") [slot]
- 10.23.2.44 void Sbs2Callback::turnReceiveMessageOff () [slot]
- 10.23.2.45 void Sbs2Callback::turnReceiveMessageOn (QString *address*, int *port*) [slot]
- 10.23.2.46 void Sbs2Callback::turnSendRawDataOff () [slot]
- 10.23.2.47 void Sbs2Callback::turnSendRawDataOn (QString *rawDataServerAddress_*, int *rawDataPort_*, int *rawDataSize_* = 32, int *rawDataQueueLength_* = 8) [slot]
- 10.23.2.48 void Sbs2Callback::udpMessageReceived (QString *data*, QString *sender*, int *port*) [signal]

10.23.3 Member Data Documentation

- 10.23.3.1 int Sbs2Callback::currentPacket = 0 [static],[protected]
- 10.23.3.2 int Sbs2Callback::currentPacketCounter = 0 [static],[protected]
- 10.23.3.3 int Sbs2Callback::devicePresent [protected]
- 10.23.3.4 int Sbs2Callback::isRecording [protected]
- 10.23.3.5 QMap<QString, QVariant> Sbs2Callback::params [protected]
- 10.23.3.6 int Sbs2Callback::samplesCollected [protected]
- 10.23.3.7 Sbs2DataHandler* Sbs2Callback::sbs2DataHandler [protected]
- 10.23.3.8 Sbs2Region* Sbs2Callback::sbs2Region [protected]
- 10.23.3.9 Sbs2Packet* Sbs2Callback::thisPacket [protected]

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

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/[sbs2callback.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/[sbs2callback.cpp](#)

10.24 Sbs2Common Class Reference

```
#include <sbs2common.h>
```

Static Public Member Functions

- static QMap< QString, QVector< int > > * [getChannels](#) ()
- static QVector< QString > * [getChannelNames](#) ()
- static QMap< QString, int > * [getCqs](#) ()
- static QVector< QString > * [getCqsMapping](#) ()
- static int [normalize](#) (int value)
- static QString [setRootAppPath](#) (QString rootAppPath_)
- static QString [getRootAppPath](#) ()
- static QString [setCatalogPath](#) (QString catalogPath_)
- static QString [getCatalogPath](#) ()
- static QString [setDefaultRootAppPath](#) ()
- static QString [setDefaultCatalogPath](#) ()
- static int [channelsNo](#) ()
- static int [samplingRate](#) ()
- static int [verticesNo](#) ()
- static void [setHardware](#) (QString hardware_)
- static QString [getCurrentHardware](#) ()
- static int [rawDataSize](#) ()

10.24.1 Member Function Documentation

10.24.1.1 int Sbs2Common::channelsNo () [static]

10.24.1.2 QString Sbs2Common::getCatalogPath () [static]

10.24.1.3 QVector< QString > * Sbs2Common::getChannelNames () [static]

10.24.1.4 QMap< QString, QVector< int > > * Sbs2Common::getChannels () [static]

10.24.1.5 QMap< QString, int > * Sbs2Common::getCqs () [static]

10.24.1.6 QVector< QString > * Sbs2Common::getCqsMapping () [static]

10.24.1.7 QString Sbs2Common::getCurrentHardware () [static]

10.24.1.8 QString Sbs2Common::getRootAppPath () [static]

10.24.1.9 int Sbs2Common::normalize (int *value*) [static]

10.24.1.10 int Sbs2Common::rawDataSize () [static]

10.24.1.11 int Sbs2Common::samplingRate () [static]

10.24.1.12 QString Sbs2Common::setCatalogPath (QString *catalogPath_*) [static]

10.24.1.13 QString Sbs2Common::setDefaultCatalogPath () [static]

10.24.1.14 QString Sbs2Common::setDefaultRootAppPath () [static]

10.24.1.15 void Sbs2Common::setHardware (QString *hardware_*) [static]

10.24.1.16 QString Sbs2Common::setRootAppPath (QString *rootAppPath_*) [static]

10.24.1.17 int Sbs2Common::verticesNo () [static]

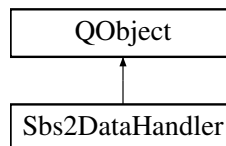
The documentation for this class was generated from the following files:

- /media/philiphj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/[sbs2common.h](#)
- /media/philiphj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/[sbs2common.cpp](#)

10.25 Sbs2DataHandler Class Reference

```
#include <sbs2datahandler.h>
```

Inheritance diagram for Sbs2DataHandler:



Public Slots

- void [setThisPacket](#) (Sbs2Packet *thisPacket_)
- void [turnFilterOn](#) (int fbandLow_, int fbandHigh_, int filterOrder_)
- void [turnFilterOff](#) ()
- void [startRecording](#) (QString user, QString description)
- void [stopRecording](#) ()
- void [insertIntoMetaFile](#) (QString event)
- void [turnChannelSpectrogramOn](#) (int spectrogramChannelSamples_=128, int spectrogramChannelLength_-=128, int spectrogramChannelDelta_=0)
- void [turnChannelSpectrogramOff](#) ()
- void [setWindowType](#) (Sbs2Spectrogram::WindowType windowType)
- void [setSourceReconstructionVerticesToExtract](#) (QVector< int > *verticesToExtract)
- void [turnOnSourceReconstructionLoreta](#) (int sourceReconstructionSamples_, int sourceReconstructionDelta_, int sourceReconstructionModelUpdateLength_, int sourceReconstructionModelUpdateDelta_, QString hardware_)
- void [turnOnSourceReconstructionSparse](#) (int sourceReconstructionSamples_, QVector< double > lambdas, QString hardware_)
- void [doSourceReconstruction](#) ()
- *Sbs2DataHandler::sourceReconstruction.*
- void [doSourceReconstructionSpectrogram](#) ()
- void [turnOffSourceReconstruction](#) ()
- void [setVerticesToExtract](#) (QVector< int > *verticesToExtract)
- void [turnSendRawDataOn](#) (QString rawDataServerAddress_, int rawDataPort_, int rawDataSize_=32, int rawDataQueueLength_=8)
- void [turnSendRawDataOff](#) ()
- void [addRawDataHost](#) (QString address, int port)
- void [removeRawDataHost](#) (QString address, int port)

- void [sendMessage](#) (QString message, QString address, int port)
- void [sendMessage](#) (QString message)
- void [addMessageUdpOutputHost](#) (QString address, int port)
- void [removeMessageUdpOutputHost](#) (QString address)
- void [clearMessageUdpOutputHosts](#) ()
- void [turnReceiveMessageOn](#) (QString address, int port)
- void [turnReceiveMessageOff](#) ()
- void [readMessage](#) (QString data, QString sender, int senderPort)

Signals

- void [spectrogramUpdated](#) ()
- void [sourceReconstructionReady](#) ()
- void [sourceReconstructionSpectrogramReady](#) ()
- void [setWindowTypeSignal](#) (Sbs2Spectrogram::WindowType windowType)
- void [udpMessageReceived](#) (QString data, QString sender, int port)

Public Member Functions

- [Sbs2DataHandler](#) (QObject *parent=0)
- [~Sbs2DataHandler](#) ()
- virtual void [filter](#) ()
- virtual void [record](#) ()
- virtual QString [getRawFilename](#) ()
- virtual void [spectrogramChannel](#) ()
- virtual void [sendRawData](#) ()
- DTU::DtArray2D< double > * [getPowerValues](#) ()
- DTU::DtArray2D< double > * [getSourceReconstructionSpectrogramValues](#) ()
- DTU::DtArray2D< double > * [getSourceReconstructionMeanValues](#) ()
- int [getPacketZero](#) ()

Protected Member Functions

- virtual void [reset](#) ()

Protected Attributes

- int [samplesCollected](#)
- [Sbs2Packet](#) * [thisPacket](#)
- int [filterOn](#)
- int [filterOrder](#)
- int [fbandLow](#)
- int [fbandHigh](#)
- [Sbs2Filter](#) * [sbs2Filter](#)
- DTU::DtArray2D< double > * [toFilterValues](#)
- DTU::DtArray2D< double > * [filterResultValues](#)
- int [recording](#)
- [Sbs2FileHandler](#) * [sbs2FileHandler](#)
- int [spectrogramChannelOn](#)
- int [spectrogramChannelSamples](#)
- int [spectrogramChannelLength](#)
- int [spectrogramChannelDelta](#)
- int [spectrogramChannelDeltaCollected](#)

- [Sbs2Spectrogram](#) * [sbs2Spectrogram](#)
- [DTU::DtuArray2D< double >](#) * [toSpectrogramValues](#)
- [DTU::DtuArray2D< double >](#) * [spectrogramValues](#)
- [DTU::DtuArray2D< double >](#) * [powerValues](#)
- [QString](#) [sourceReconstructionMethod](#)
- [int](#) [sourceReconstructionOn](#)
- [int](#) [isSourceReconstructionReady](#)
- [int](#) [sourceReconstructionSamples](#)
- [int](#) [sourceReconstructionDelta](#)
- [int](#) [sourceReconstructionDeltaCollected](#)
- [int](#) [sourceReconstructionModelUpdateLength](#)
- [int](#) [sourceReconstructionModelUpdateDelta](#)
- [int](#) [readyToReconstruct](#)
- [QString](#) [hardware](#)
- [Sbs2SourceReconstruction](#) * [sbs2SourceReconstruction](#)
- [DTU::DtuArray2D< double >](#) * [toSourceReconstructionValues](#)
- [DTU::DtuArray2D< double >](#) * [sourceReconstructionValues](#)
- [DTU::DtuArray2D< double >](#) * [sourceReconstructionSpectrogramValues](#)
- [int](#) [networkSendRawDataOn](#)
- [Sbs2NetworkHandler](#) * [sbs2NetworkHandler](#)
- [int](#) [packetsSeen](#)

10.25.1 Constructor & Destructor Documentation

10.25.1.1 [Sbs2DataHandler::Sbs2DataHandler \(QObject * *parent* = 0 \)](#) [explicit]

10.25.1.2 [Sbs2DataHandler::~~Sbs2DataHandler \(\)](#)

10.25.2 Member Function Documentation

10.25.2.1 [void Sbs2DataHandler::addMessageUdpOutputHost \(QString *address*, int *port* \)](#) [slot]

10.25.2.2 [void Sbs2DataHandler::addRawDataHost \(QString *address*, int *port* \)](#) [slot]

10.25.2.3 [void Sbs2DataHandler::clearMessageUdpOutputHosts \(\)](#) [slot]

10.25.2.4 [void Sbs2DataHandler::doSourceReconstruction \(\)](#) [slot]

[Sbs2DataHandler::sourceReconstruction.](#)

Runs the source reconstruction with [Sbs2SourceReconstruction::doRec](#). This only happens if 'sourceReconstruction-On' is turned on. This variable is controlled by [Sbs2DataHandler::turnSourceReconstructionOn](#)

10.25.2.5 [void Sbs2DataHandler::doSourceReconstructionSpectrogram \(\)](#) [slot]

10.25.2.6 [void Sbs2DataHandler::filter \(\)](#) [virtual]

10.25.2.7 [int Sbs2DataHandler::getPacketZero \(\)](#)

10.25.2.8 [DTU::DtuArray2D< double > * Sbs2DataHandler::getPowerValues \(\)](#)

10.25.2.9 [QString Sbs2DataHandler::getRawFilename \(\)](#) [virtual]

10.25.2.10 [DTU::DtuArray2D< double > * Sbs2DataHandler::getSourceReconstructionMeanValues \(\)](#)

- 10.25.2.11 **DTU::DtuArray2D< double > * Sbs2DataHandler::getSourceReconstructionSpectrogramValues ()**
- 10.25.2.12 **void Sbs2DataHandler::insertIntoMetaFile (QString *event*) [slot]**
- 10.25.2.13 **void Sbs2DataHandler::readMessage (QString *data*, QString *sender*, int *senderPort*) [slot]**
- 10.25.2.14 **void Sbs2DataHandler::record () [virtual]**
- 10.25.2.15 **void Sbs2DataHandler::removeMessageUdpOutputHost (QString *address*) [slot]**
- 10.25.2.16 **void Sbs2DataHandler::removeRawDataHost (QString *address*, int *port*) [slot]**
- 10.25.2.17 **void Sbs2DataHandler::reset () [protected],[virtual]**
- 10.25.2.18 **void Sbs2DataHandler::sendMessage (QString *message*, QString *address*, int *port*) [slot]**
- 10.25.2.19 **void Sbs2DataHandler::sendMessage (QString *message*) [slot]**
- 10.25.2.20 **void Sbs2DataHandler::sendRawData () [virtual]**
- 10.25.2.21 **void Sbs2DataHandler::setSourceReconstructionVerticesToExtract (QVector< int > * *verticesToExtract*) [slot]**
- 10.25.2.22 **void Sbs2DataHandler::setThisPacket (Sbs2Packet * *thisPacket_*) [slot]**
- 10.25.2.23 **void Sbs2DataHandler::setVerticesToExtract (QVector< int > * *verticesToExtract*) [slot]**
- 10.25.2.24 **void Sbs2DataHandler::setWindowType (Sbs2Spectrogram::WindowType *windowType*) [slot]**
- 10.25.2.25 **void Sbs2DataHandler::setWindowTypeSignal (Sbs2Spectrogram::WindowType *windowType*) [signal]**
- 10.25.2.26 **void Sbs2DataHandler::sourceReconstructionReady () [signal]**
- 10.25.2.27 **void Sbs2DataHandler::sourceReconstructionSpectrogramReady () [signal]**
- 10.25.2.28 **void Sbs2DataHandler::spectrogramChannel () [virtual]**
- 10.25.2.29 **void Sbs2DataHandler::spectrogramUpdated () [signal]**
- 10.25.2.30 **void Sbs2DataHandler::startRecording (QString *user*, QString *description*) [slot]**
- 10.25.2.31 **void Sbs2DataHandler::stopRecording () [slot]**
- 10.25.2.32 **void Sbs2DataHandler::turnChannelSpectrogramOff () [slot]**
- 10.25.2.33 **void Sbs2DataHandler::turnChannelSpectrogramOn (int *spectrogramChannelSamples_* = 128, int *spectrogramChannelLength_* = 128, int *spectrogramChannelDelta_* = 0) [slot]**
- 10.25.2.34 **void Sbs2DataHandler::turnFilterOff () [slot]**
- 10.25.2.35 **void Sbs2DataHandler::turnFilterOn (int *fbandLow_*, int *fbandHigh_*, int *filterOrder_*) [slot]**
- 10.25.2.36 **void Sbs2DataHandler::turnOffSourceReconstruction () [slot]**

- 10.25.2.37 void Sbs2DataHandler::turnOnSourceReconstructionLoreta (int *sourceReconstructionSamples_*, int *sourceReconstructionDelta_*, int *sourceReconstructionModelUpdateLength_*, int *sourceReconstructionModelUpdateDelta_*, QString *hardware_*) [slot]
- 10.25.2.38 void Sbs2DataHandler::turnOnSourceReconstructionSparse (int *sourceReconstructionSamples_*, QVector< double > *lambdas*, QString *hardware_*) [slot]
- 10.25.2.39 void Sbs2DataHandler::turnReceiveMessageOff () [slot]
- 10.25.2.40 void Sbs2DataHandler::turnReceiveMessageOn (QString *address*, int *port*) [slot]
- 10.25.2.41 void Sbs2DataHandler::turnSendRawDataOff () [slot]
- 10.25.2.42 void Sbs2DataHandler::turnSendRawDataOn (QString *rawDataServerAddress_*, int *rawDataPort_*, int *rawDataSize_* = 32, int *rawDataQueueLength_* = 8) [slot]
- 10.25.2.43 void Sbs2DataHandler::udpMessageReceived (QString *data*, QString *sender*, int *port*) [signal]

10.25.3 Member Data Documentation

- 10.25.3.1 int Sbs2DataHandler::fbandHigh [protected]
- 10.25.3.2 int Sbs2DataHandler::fbandLow [protected]
- 10.25.3.3 int Sbs2DataHandler::filterOn [protected]
- 10.25.3.4 int Sbs2DataHandler::filterOrder [protected]
- 10.25.3.5 DTU::DtuArray2D<double>* Sbs2DataHandler::filterResultValues [protected]
- 10.25.3.6 QString Sbs2DataHandler::hardware [protected]
- 10.25.3.7 int Sbs2DataHandler::isSourceReconstructionReady [protected]
- 10.25.3.8 int Sbs2DataHandler::networkSendRawDataOn [protected]
- 10.25.3.9 int Sbs2DataHandler::packetsSeen [protected]
- 10.25.3.10 DTU::DtuArray2D<double>* Sbs2DataHandler::powerValues [protected]
- 10.25.3.11 int Sbs2DataHandler::readyToReconstruct [protected]
- 10.25.3.12 int Sbs2DataHandler::recording [protected]
- 10.25.3.13 int Sbs2DataHandler::samplesCollected [protected]
- 10.25.3.14 Sbs2FileHandler* Sbs2DataHandler::sbs2FileHandler [protected]
- 10.25.3.15 Sbs2Filter* Sbs2DataHandler::sbs2Filter [protected]
- 10.25.3.16 Sbs2NetworkHandler* Sbs2DataHandler::sbs2NetworkHandler [protected]
- 10.25.3.17 Sbs2SourceReconstruction* Sbs2DataHandler::sbs2SourceReconstruction [protected]
- 10.25.3.18 Sbs2Spectrogram* Sbs2DataHandler::sbs2Spectrogram [protected]

- 10.25.3.19 `int Sbs2DataHandler::sourceReconstructionDelta` [protected]
- 10.25.3.20 `int Sbs2DataHandler::sourceReconstructionDeltaCollected` [protected]
- 10.25.3.21 `QString Sbs2DataHandler::sourceReconstructionMethod` [protected]
- 10.25.3.22 `int Sbs2DataHandler::sourceReconstructionModelUpdateDelta` [protected]
- 10.25.3.23 `int Sbs2DataHandler::sourceReconstructionModelUpdateLength` [protected]
- 10.25.3.24 `int Sbs2DataHandler::sourceReconstructionOn` [protected]
- 10.25.3.25 `int Sbs2DataHandler::sourceReconstructionSamples` [protected]
- 10.25.3.26 `DTU::DtuArray2D<double>* Sbs2DataHandler::sourceReconstructionSpectrogramValues` [protected]
- 10.25.3.27 `DTU::DtuArray2D<double>* Sbs2DataHandler::sourceReconstructionValues` [protected]
- 10.25.3.28 `int Sbs2DataHandler::spectrogramChannelDelta` [protected]
- 10.25.3.29 `int Sbs2DataHandler::spectrogramChannelDeltaCollected` [protected]
- 10.25.3.30 `int Sbs2DataHandler::spectrogramChannelLength` [protected]
- 10.25.3.31 `int Sbs2DataHandler::spectrogramChannelOn` [protected]
- 10.25.3.32 `int Sbs2DataHandler::spectrogramChannelSamples` [protected]
- 10.25.3.33 `DTU::DtuArray2D<double>* Sbs2DataHandler::spectrogramValues` [protected]
- 10.25.3.34 `Sbs2Packet* Sbs2DataHandler::thisPacket` [protected]
- 10.25.3.35 `DTU::DtuArray2D<double>* Sbs2DataHandler::toFilterValues` [protected]
- 10.25.3.36 `DTU::DtuArray2D<double>* Sbs2DataHandler::toSourceReconstructionValues` [protected]
- 10.25.3.37 `DTU::DtuArray2D<double>* Sbs2DataHandler::toSpectrogramValues` [protected]

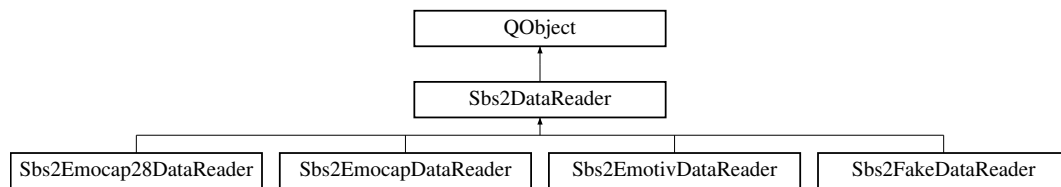
The documentation for this class was generated from the following files:

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.cpp](#)

10.26 Sbs2DataReader Class Reference

```
#include <sbs2datareader.h>
```

Inheritance diagram for Sbs2DataReader:



Public Slots

- virtual void [deviceFound](#) (QMap< QString, QVariant > params)
- virtual void [deviceLost](#) ()
- virtual void [aboutToQuit](#) ()
- virtual void [udpDataReceived](#) (QVector< char * > *data, int counter)
- virtual void [udpDataReceived](#) (QUdpSocket *rawDataUdpInputSocket)
- virtual void [turnReceiveUdpDataOn](#) (QString address, int port)
- virtual void [turnReceiveUdpDataOff](#) ()

Signals

- void [deviceFoundSignal](#) (QMap< QString, QVariant > params)

Public Member Functions

- [~Sbs2DataReader](#) ()

Protected Member Functions

- [Sbs2DataReader](#) ([Sbs2Callback](#) *sbs2Callback_, int readOnlyFromNetwork_=0, QObject *parent=0)
- virtual void [execute](#) ()

Protected Attributes

- int [framesRead](#)
- int [currentIndex](#)
- int [bufferIndex](#)
- int [bufferSize](#)
- int [running](#)
- [Sbs2Callback](#) * [sbs2Callback](#)
- int [testDummyRead](#)
- [Sbs2NetworkHandler](#) * [sbs2NetworkHandler](#)
- int [readOnlyFromNetwork](#)
- int [lastReceiveRawDataCounter](#)

10.26.1 Constructor & Destructor Documentation

10.26.1.1 Sbs2DataReader::~~Sbs2DataReader ()

10.26.1.2 Sbs2DataReader::Sbs2DataReader ([Sbs2Callback](#) * *sbs2Callback_*, int *readOnlyFromNetwork_* = 0, QObject * *parent* = 0) [protected]

10.26.2 Member Function Documentation

- 10.26.2.1 void Sbs2DataReader::aboutToQuit () [virtual],[slot]
- 10.26.2.2 void Sbs2DataReader::deviceFound (QMap< QString, QVariant > *params*) [virtual],[slot]
- 10.26.2.3 void Sbs2DataReader::deviceFoundSignal (QMap< QString, QVariant > *params*) [signal]
- 10.26.2.4 void Sbs2DataReader::deviceLost () [virtual],[slot]
- 10.26.2.5 void Sbs2DataReader::execute () [protected],[virtual]
- 10.26.2.6 void Sbs2DataReader::turnReceiveUdpDataOff () [virtual],[slot]
- 10.26.2.7 void Sbs2DataReader::turnReceiveUdpDataOn (QString *address*, int *port*) [virtual],[slot]
- 10.26.2.8 void Sbs2DataReader::udpDataReceived (QVector< char * > * *data*, int *counter*) [virtual],[slot]
- 10.26.2.9 void Sbs2DataReader::udpDataReceived (QUdpSocket * *rawDataUdpInputSocket*) [virtual],[slot]

10.26.3 Member Data Documentation

- 10.26.3.1 int Sbs2DataReader::bufferIndex [protected]
- 10.26.3.2 int Sbs2DataReader::bufferSize [protected]
- 10.26.3.3 int Sbs2DataReader::currentIndex [protected]
- 10.26.3.4 int Sbs2DataReader::framesRead [protected]
- 10.26.3.5 int Sbs2DataReader::lastReceiveRawDataCounter [protected]
- 10.26.3.6 int Sbs2DataReader::readOnlyFromNetwork [protected]
- 10.26.3.7 int Sbs2DataReader::running [protected]
- 10.26.3.8 Sbs2Callback* Sbs2DataReader::sbs2Callback [protected]
- 10.26.3.9 Sbs2NetworkHandler* Sbs2DataReader::sbs2NetworkHandler [protected]
- 10.26.3.10 int Sbs2DataReader::testDummyRead [protected]

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

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/[sbs2datareader.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/[sbs2datareader.cpp](#)

10.27 Sbs2Emocap28DataContainer Class Reference

```
#include <sbs2emocap28datareader.h>
```

Public Member Functions

- [Sbs2Emocap28DataContainer](#) ()
- void [update](#) (char *data_, int counter_)

Public Attributes

- char * [data](#)
- int [counter](#)

10.27.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.27.2 Constructor & Destructor Documentation

10.27.2.1 Sbs2Emocap28DataContainer::Sbs2Emocap28DataContainer () [\[inline\]](#)

10.27.3 Member Function Documentation

10.27.3.1 void Sbs2Emocap28DataContainer::update (char * [data_](#), int [counter_](#))

10.27.4 Member Data Documentation

10.27.4.1 int Sbs2Emocap28DataContainer::counter

10.27.4.2 char* Sbs2Emocap28DataContainer::data

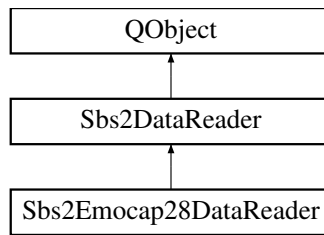
The documentation for this class was generated from the following files:

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28datacontainer.h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28datacontainer.cpp](#)

10.28 Sbs2Emocap28DataReader Class Reference

```
#include <sbs2emocap28datareader.h>
```

Inheritance diagram for Sbs2Emocap28DataReader:



Public Slots

- void [deviceFound](#) (QMap< QString, QVariant > params)
- void [deviceLost](#) ()
- void [aboutToQuit](#) ()
- void [udpDataReceived](#) (QVector< char * > *data, int counter)
- void [udpDataReceived](#) (QUdpSocket *rawDataUdpInputSocket)
- void [turnReceiveUdpDataOn](#) (QString address, int port)
- void [turnReceiveUdpDataOff](#) ()

Signals

- void [amp1FoundSignal](#) (QVariant number, QVariant path, QVariant serialNumber)
- void [amp2FoundSignal](#) (QVariant number, QVariant path, QVariant serialNumber)
- void [readyForData](#) ()
- void [inMappingSignal](#) ()
- void [mappingSuccessful](#) (int mapping)
- void [mappingFailed](#) ()
- void [alignedSignal](#) (int mapping, int mappingAlignment, int mappingCorr)

Public Member Functions

- [~Sbs2Emocap28DataReader](#) ()

Static Public Member Functions

- static [Sbs2Emocap28DataReader](#) * [New](#) ([Sbs2Callback](#) *sbs2Callback_, int readOnlyFromNetwork_=0, QObject *parent=0)

Additional Inherited Members

10.28.1 Constructor & Destructor Documentation

- 10.28.1.1 [Sbs2Emocap28DataReader::~~Sbs2Emocap28DataReader](#) ()

10.28.2 Member Function Documentation

- 10.28.2.1 void [Sbs2Emocap28DataReader::aboutToQuit](#) () [slot]

- 10.28.2.2 void [Sbs2Emocap28DataReader::alignedSignal](#) (int *mapping*, int *mappingAlignment*, int *mappingCorr*) [signal]

- 10.28.2.3 void Sbs2Emocap28DataReader::amp1FoundSignal (QVariant *number*, QVariant *path*, QVariant *serialNumber*)
[signal]
- 10.28.2.4 void Sbs2Emocap28DataReader::amp2FoundSignal (QVariant *number*, QVariant *path*, QVariant *serialNumber*)
[signal]
- 10.28.2.5 void Sbs2Emocap28DataReader::deviceFound (QMap< QString, QVariant > *params*) [slot]
- 10.28.2.6 void Sbs2Emocap28DataReader::deviceLost () [slot]
- 10.28.2.7 void Sbs2Emocap28DataReader::inMappingSignal () [signal]
- 10.28.2.8 void Sbs2Emocap28DataReader::mappingFailed () [signal]
- 10.28.2.9 void Sbs2Emocap28DataReader::mappingSuccessful (int *mapping*) [signal]
- 10.28.2.10 Sbs2Emocap28DataReader * Sbs2Emocap28DataReader::New (Sbs2Callback * *sbs2Callback_*, int
readOnlyFromNetwork_ = 0, QObject * *parent* = 0) [static]
- 10.28.2.11 void Sbs2Emocap28DataReader::readyForData () [signal]
- 10.28.2.12 void Sbs2Emocap28DataReader::turnReceiveUdpDataOff () [slot]
- 10.28.2.13 void Sbs2Emocap28DataReader::turnReceiveUdpDataOn (QString *address*, int *port*) [slot]
- 10.28.2.14 void Sbs2Emocap28DataReader::udpDataReceived (QVector< char * > * *data*, int *counter*) [slot]
- 10.28.2.15 void Sbs2Emocap28DataReader::udpDataReceived (QUdpSocket * *rawDataUdpInputSocket*) [slot]

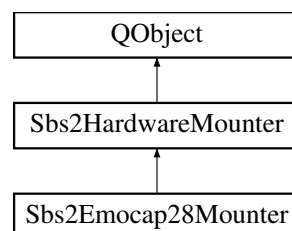
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/[sbs2emocap28mounter.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/[sbs2emocap28mounter.cpp](#)

10.29 Sbs2Emocap28Mounter Class Reference

```
#include <sbs2emocap28mounter.h>
```

Inheritance diagram for Sbs2Emocap28Mounter:



Public Slots

- void [start](#) ()
- void [stop](#) ()
- void [invalidate](#) ()

Public Member Functions

- [~Sbs2Emocap28Mounter](#) ()

Static Public Member Functions

- static [Sbs2Emocap28Mounter](#) * [New](#) (QObject *parent=0)

Additional Inherited Members

10.29.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU](#) Informatics, Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. On OSX we use hidapi to access raw data.

10.29.2 Constructor & Destructor Documentation

10.29.2.1 [Sbs2Emocap28Mounter::~Sbs2Emocap28Mounter](#) ()

10.29.3 Member Function Documentation

10.29.3.1 [void Sbs2Emocap28Mounter::invalidate](#) () [slot]

10.29.3.2 [Sbs2Emocap28Mounter * Sbs2Emocap28Mounter::New](#) (QObject * *parent* = 0) [static]

10.29.3.3 [void Sbs2Emocap28Mounter::start](#) () [slot]

10.29.3.4 [void Sbs2Emocap28Mounter::stop](#) () [slot]

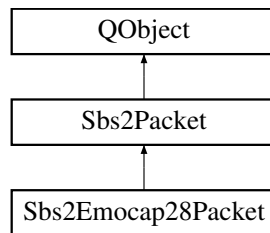
The documentation for this class was generated from the following files:

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28Mounter.h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28Mounter.cpp](#)

10.30 Sbs2Emocap28Packet Class Reference

```
#include <sbs2emocap28packet.h>
```

Inheritance diagram for Sbs2Emocap28Packet:



Public Member Functions

- [Sbs2Emocap28Packet](#) (QObject *parent)
- void [update](#) (char *data)
Method for updating data in the packet. To avoid continuous creation and destruction of objects, certain number of empty packets is constructed in initialization and then updated with wrap-around. Packets should see the raw data delivered by [Sbs2DataReader](#) and form themselves.
- void [update](#) (char *data1, char *data2)
- int [getCounter](#) (char *data)
- int [getValue](#) (char *data)

Additional Inherited Members

10.30.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. [http://code.-google.com/p/smartphonebrainscanner2](http://code.google.com/p/smartphonebrainscanner2)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.30.2 Constructor & Destructor Documentation

10.30.2.1 [Sbs2Emocap28Packet::Sbs2Emocap28Packet \(QObject * parent \)](#)

10.30.3 Member Function Documentation

10.30.3.1 `int Sbs2Emocap28Packet::getCounter (char * data)`

10.30.3.2 `int Sbs2Emocap28Packet::getValue (char * data)`

10.30.3.3 `void Sbs2Emocap28Packet::update (char * data)` [virtual]

Method for updating data in the packet. To avoid continuous creation and destruction of objects, certain number of empty packets is constructed in initialization and then updated with wrap-around. Packets should see the raw data delivered by [Sbs2DataReader](#) and form themselves.

Parameters

<i>data</i>	Pointer to raw data.
-------------	----------------------

Reimplemented from [Sbs2Packet](#).

10.30.3.4 `void Sbs2Emocap28Packet::update (char * data1, char * data2)`

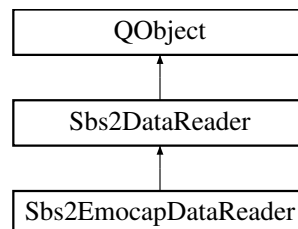
The documentation for this class was generated from the following files:

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28packet.h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28packet.cpp](#)

10.31 Sbs2EmocapDataReader Class Reference

```
#include <sbs2emocapdatareader.h>
```

Inheritance diagram for Sbs2EmocapDataReader:



Public Slots

- void [deviceFound](#) (QMap< QString, QVariant > params)
- void [deviceLost](#) ()
- void [aboutToQuit](#) ()
- void [udpDataReceived](#) (QVector< char * > *data, int counter)
- void [udpDataReceived](#) (QUdpSocket *rawDataUdpInputSocket)
- void [turnReceiveUdpDataOn](#) (QString address, int port)
- void [turnReceiveUdpDataOff](#) ()

Public Member Functions

- [~Sbs2EmocapDataReader](#) ()

Static Public Member Functions

- static [Sbs2EmocapDataReader](#) * [New](#) ([Sbs2Callback](#) *sbs2Callback_, int readOnlyFromNetwork_=0, QObject *parent=0)

Additional Inherited Members

10.31.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.31.2 Constructor & Destructor Documentation

10.31.2.1 [Sbs2EmocapDataReader::~Sbs2EmocapDataReader \(\)](#)

10.31.3 Member Function Documentation

10.31.3.1 [void Sbs2EmocapDataReader::aboutToQuit \(\) \[slot\]](#)

10.31.3.2 [void Sbs2EmocapDataReader::deviceFound \(QMap< QString, QVariant > *params* \) \[slot\]](#)

10.31.3.3 [void Sbs2EmocapDataReader::deviceLost \(\) \[slot\]](#)

10.31.3.4 [Sbs2EmocapDataReader * Sbs2EmocapDataReader::New \(Sbs2Callback * *sbs2Callback_*, int *readOnlyFromNetwork_* = 0, QObject * *parent* = 0 \) \[static\]](#)

10.31.3.5 [void Sbs2EmocapDataReader::turnReceiveUdpDataOff \(\) \[slot\]](#)

10.31.3.6 [void Sbs2EmocapDataReader::turnReceiveUdpDataOn \(QString *address*, int *port* \) \[slot\]](#)

10.31.3.7 [void Sbs2EmocapDataReader::udpDataReceived \(QVector< char * > * *data*, int *counter* \) \[slot\]](#)

10.31.3.8 [void Sbs2EmocapDataReader::udpDataReceived \(QUdpSocket * *rawDataUdpInputSocket* \) \[slot\]](#)

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

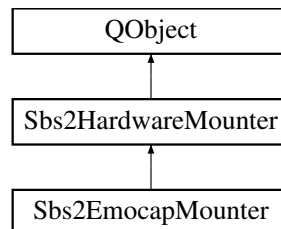
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2h](#)

- [/media/philipjhj/Data/OneDrive/Studie/Studentprogrammer/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocapmounter.cpp](#)

10.32 Sbs2EmocapMounter Class Reference

```
#include <sbs2emocapmounter.h>
```

Inheritance diagram for Sbs2EmocapMounter:



Public Slots

- void [start](#) ()
- void [stop](#) ()
- void [invalidate](#) ()

Public Member Functions

- [~Sbs2EmocapMounter](#) ()

Static Public Member Functions

- static [Sbs2EmocapMounter](#) * [New](#) (QObject *parent=0)

Additional Inherited Members

10.32.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. On OSX we use hidapi to access raw data.

10.32.2 Constructor & Destructor Documentation

10.32.2.1 `Sbs2EmocapMounter::~~Sbs2EmocapMounter ()`

10.32.3 Member Function Documentation

10.32.3.1 `void Sbs2EmocapMounter::invalidate ()` [slot]

10.32.3.2 `Sbs2EmocapMounter * Sbs2EmocapMounter::New (QObject * parent = 0)` [static]

10.32.3.3 `void Sbs2EmocapMounter::start ()` [slot]

10.32.3.4 `void Sbs2EmocapMounter::stop ()` [slot]

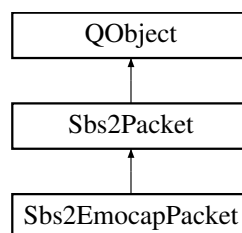
The documentation for this class was generated from the following files:

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2cpp](#)

10.33 Sbs2EmocapPacket Class Reference

```
#include <sbs2emocappacket.h>
```

Inheritance diagram for Sbs2EmocapPacket:



Public Member Functions

- [Sbs2EmocapPacket](#) (QObject *parent)
- void [update](#) (char *data)

Method for updating data in the packet. To avoid continuous creation and destruction of objects, certain number of empty packets is constructed in initialization and then updated with wrap-around. Packets should see the raw data delivered by [Sbs2DataReader](#) and form themselves.

Additional Inherited Members

10.33.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.33.2 Constructor & Destructor Documentation

10.33.2.1 `Sbs2EmocapPacket::Sbs2EmocapPacket (QObject * parent)`

10.33.3 Member Function Documentation

10.33.3.1 `void Sbs2EmocapPacket::update (char * data) [virtual]`

Method for updating data in the packet. To avoid continuous creation and destruction of objects, certain number of empty packets is constructed in initialization and then updated with wrap-around. Packets should see the raw data delivered by [Sbs2DataReader](#) and form themselves.

Parameters

<i>data</i>	Pointer to raw data.
-------------	----------------------

Reimplemented from [Sbs2Packet](#).

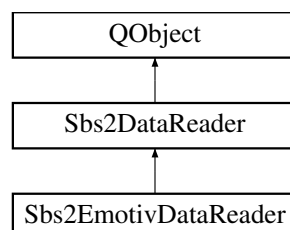
The documentation for this class was generated from the following files:

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2.h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2.cpp](#)

10.34 Sbs2EmotivDataReader Class Reference

```
#include <sbs2emotivdatareader.h>
```

Inheritance diagram for Sbs2EmotivDataReader:



Public Slots

- void [deviceFound](#) (QMap< QString, QVariant > params)
- void [deviceLost](#) ()

- void [aboutToQuit](#) ()
- void [udpDataReceived](#) (QVector< char * > *data, int counter)
- void [udpDataReceived](#) (QUdpSocket *rawDataUdpInputSocket)
- void [turnReceiveUdpDataOn](#) (QString address, int port)
- void [turnReceiveUdpDataOff](#) ()

Public Member Functions

- [~Sbs2EmotivDataReader](#) ()

Static Public Member Functions

- static [Sbs2EmotivDataReader](#) * [New](#) ([Sbs2Callback](#) *sbs2Callback_, int readOnlyFromNetwork_=0, QObject *parent=0)

Additional Inherited Members

10.34.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.34.2 Constructor & Destructor Documentation

10.34.2.1 [Sbs2EmotivDataReader::~~Sbs2EmotivDataReader](#) ()

10.34.3 Member Function Documentation

10.34.3.1 void [Sbs2EmotivDataReader::aboutToQuit](#) () [[slot](#)]

10.34.3.2 void [Sbs2EmotivDataReader::deviceFound](#) (QMap< QString, QVariant > *params*) [[slot](#)]

10.34.3.3 void [Sbs2EmotivDataReader::deviceLost](#) () [[slot](#)]

10.34.3.4 [Sbs2EmotivDataReader](#) * [Sbs2EmotivDataReader::New](#) ([Sbs2Callback](#) * *sbs2Callback_*, int *readOnlyFromNetwork_* = 0, QObject * *parent* = 0) [[static](#)]

10.34.3.5 void Sbs2EmotivDataReader::turnReceiveUdpDataOff () [slot]

10.34.3.6 void Sbs2EmotivDataReader::turnReceiveUdpDataOn (QString *address*, int *port*) [slot]

10.34.3.7 void Sbs2EmotivDataReader::udpDataReceived (QVector< char * > * *data*, int *counter*) [slot]

10.34.3.8 void Sbs2EmotivDataReader::udpDataReceived (QUdpSocket * *rawDataUdpInputSocket*) [slot]

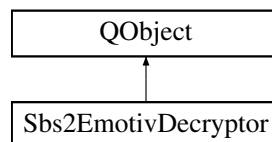
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/[sbs2emotivdatareader.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/[sbs2emotivdatareader.cpp](#)

10.35 Sbs2EmotivDecryptor Class Reference

```
#include <sbs2emotivdecryptor.h>
```

Inheritance diagram for Sbs2EmotivDecryptor:



Public Member Functions

- [Sbs2EmotivDecryptor](#) (QObject *parent=0)
- void [setSerialNumber](#) (char *serialNumber_)
- void [setSerialNumber](#) (QString serialNumber_)
- void [decrypt](#) (char cipher[], char plain[])

10.35.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.35.2 Constructor & Destructor Documentation

10.35.2.1 `Sbs2EmotivDecryptor::Sbs2EmotivDecryptor (QObject * parent = 0)` `[explicit]`

10.35.3 Member Function Documentation

10.35.3.1 `void Sbs2EmotivDecryptor::decrypt (char cipher[], char plain[])`

10.35.3.2 `void Sbs2EmotivDecryptor::setSerialNumber (char * serialNumber_)`

10.35.3.3 `void Sbs2EmotivDecryptor::setSerialNumber (QString serialNumber_)`

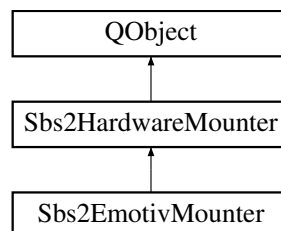
The documentation for this class was generated from the following files:

- `/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivmounter.h`
- `/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivmounter_dummy.cpp`

10.36 Sbs2EmotivMounter Class Reference

```
#include <sbs2emotivmounter.h>
```

Inheritance diagram for Sbs2EmotivMounter:



Public Slots

- `void start ()`
- `void stop ()`
- `void invalidate ()`

Public Member Functions

- `~Sbs2EmotivMounter ()`

Static Public Member Functions

- `static Sbs2EmotivMounter * New (QObject *parent=0)`

Additional Inherited Members

10.36.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics, Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>](http://code.google.com/p/smartphonebrainscanner2)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. On OSX we use hidapi to access raw data.

10.36.2 Constructor & Destructor Documentation

10.36.2.1 `Sbs2EmotivMounter::~~Sbs2EmotivMounter ()`

10.36.3 Member Function Documentation

10.36.3.1 `void Sbs2EmotivMounter::invalidate () [slot]`

10.36.3.2 `Sbs2EmotivMounter * Sbs2EmotivMounter::New (QObject * parent = 0) [static]`

10.36.3.3 `void Sbs2EmotivMounter::start () [slot]`

10.36.3.4 `void Sbs2EmotivMounter::stop () [slot]`

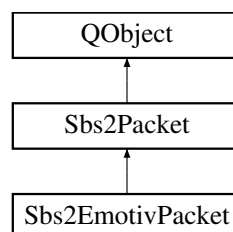
The documentation for this class was generated from the following files:

- `/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivmounter.h`
- `/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivmounter.cpp`

10.37 Sbs2EmotivPacket Class Reference

```
#include <sbs2emotivpacket.h>
```

Inheritance diagram for Sbs2EmotivPacket:



Public Member Functions

- [Sbs2EmotivPacket](#) (QObject *parent)
- void [update](#) (char *data)

Method for updating data in the packet. To avoid continuous creation and destruction of objects, certain number of empty packets is constructed in initialization and then updated with wrap-around. Packets should see the raw data delivered by [Sbs2DataReader](#) and form themselves.

Additional Inherited Members

10.37.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.37.2 Constructor & Destructor Documentation

10.37.2.1 [Sbs2EmotivPacket::Sbs2EmotivPacket \(QObject * parent \)](#)

10.37.3 Member Function Documentation

10.37.3.1 [void Sbs2EmotivPacket::update \(char * data \)](#) [virtual]

Method for updating data in the packet. To avoid continuous creation and destruction of objects, certain number of empty packets is constructed in initialization and then updated with wrap-around. Packets should see the raw data delivered by [Sbs2DataReader](#) and form themselves.

Parameters

<i>data</i>	Pointer to raw data.
-------------	----------------------

Reimplemented from [Sbs2Packet](#).

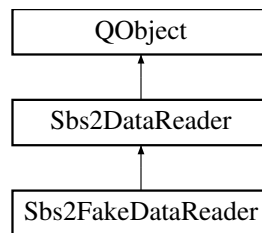
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/[sbs2emotivpacket.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/[sbs2emotivpacket.cpp](#)

10.38 Sbs2FakeDataReader Class Reference

```
#include <sbs2fakedatareader.h>
```

Inheritance diagram for Sbs2FakeDataReader:



Public Slots

- void [start](#) ()
- void [stop](#) ()
- void [setFilename](#) (QString filename_)

Public Member Functions

- [~Sbs2FakeDataReader](#) ()

Static Public Member Functions

- static [Sbs2FakeDataReader](#) * [New](#) ([Sbs2Callback](#) *sbs2Callback_, int readOnlyFromNetwork_=0, QObject *parent=0)

Additional Inherited Members

10.38.1 Constructor & Destructor Documentation

10.38.1.1 [Sbs2FakeDataReader::~~Sbs2FakeDataReader](#) ()

10.38.2 Member Function Documentation

10.38.2.1 [Sbs2FakeDataReader](#) * [Sbs2FakeDataReader::New](#) ([Sbs2Callback](#) * *sbs2Callback_*, int *readOnlyFromNetwork_* = 0, QObject * *parent* = 0) [static]

10.38.2.2 void [Sbs2FakeDataReader::setFilename](#) (QString *filename_*) [inline],[slot]

10.38.2.3 void [Sbs2FakeDataReader::start](#) () [slot]

10.38.2.4 void [Sbs2FakeDataReader::stop](#) () [inline],[slot]

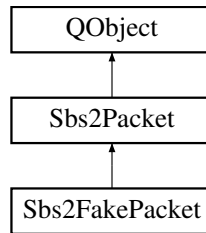
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/[sbs2fakedatareader.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/[sbs2fakedatareader.cpp](#)

10.39 Sbs2FakePacket Class Reference

```
#include <sbs2fakepacket.h>
```

Inheritance diagram for Sbs2FakePacket:



Public Member Functions

- [Sbs2FakePacket](#) (QObject *parent)
- void [update](#) (double *data)

Additional Inherited Members

10.39.1 Constructor & Destructor Documentation

10.39.1.1 Sbs2FakePacket::Sbs2FakePacket (QObject * *parent*)

10.39.2 Member Function Documentation

10.39.2.1 void Sbs2FakePacket::update (double * *data*)

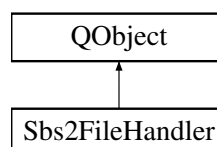
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/[sbs2fakepacket.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/[sbs2fakepacket.cpp](#)

10.40 Sbs2FileHandler Class Reference

```
#include <sbs2filehandler.h>
```

Inheritance diagram for Sbs2FileHandler:



Public Slots

- void [insertIntoMetaFile](#) (QString event)
- void [close](#) ()
- void [createMetaFile](#) (QString user_, QString description_)

Public Member Functions

- void `dumpRawData` (char *rawData)
- `~Sbs2FileHandler` ()
- QString `getRawFilename` ()
- int `getPacketZero` ()

Static Public Member Functions

- static `Sbs2FileHandler * New` (QObject *parent=0)

10.40.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](http://code.google.com/p/smartphonebrainscanner2), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.40.2 Constructor & Destructor Documentation

10.40.2.1 `Sbs2FileHandler::~~Sbs2FileHandler` ()

10.40.3 Member Function Documentation

10.40.3.1 `void Sbs2FileHandler::close` () [slot]

10.40.3.2 `void Sbs2FileHandler::createMetaFile` (QString *user_*, QString *description_*) [slot]

10.40.3.3 `void Sbs2FileHandler::dumpRawData` (char * *rawData*)

10.40.3.4 `int Sbs2FileHandler::getPacketZero` ()

10.40.3.5 `QString Sbs2FileHandler::getRawFilename` ()

10.40.3.6 `void Sbs2FileHandler::insertIntoMetaFile` (QString *event*) [slot]

10.40.3.7 `Sbs2FileHandler * Sbs2FileHandler::New` (QObject * *parent* = 0) [static]

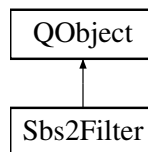
The documentation for this class was generated from the following files:

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filehandler.-h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filehandler.-cpp](#)

10.41 Sbs2Filter Class Reference

```
#include <sbs2filter.h>
```

Inheritance diagram for Sbs2Filter:



Public Member Functions

- void [loadFilter](#) ()
- void [updateFilter](#) (int order_, int fbandLow_, int fbandHigh_)
- void [doFilter](#) (DTU::DtuArray2D< double > *values, DTU::DtuArray2D< double > *returnValues)
- [~Sbs2Filter](#) ()

Static Public Member Functions

- static [Sbs2Filter](#) * [New](#) (int fbandLow_, int fbandHigh_, int order_, QObject *parent=0)

10.41.1 Constructor & Destructor Documentation

10.41.1.1 [Sbs2Filter::~~Sbs2Filter](#) ()

10.41.2 Member Function Documentation

10.41.2.1 void [Sbs2Filter::doFilter](#) (DTU::DtuArray2D< double > * *values*, DTU::DtuArray2D< double > * *returnValues*)

10.41.2.2 void [Sbs2Filter::loadFilter](#) ()

10.41.2.3 [Sbs2Filter](#) * [Sbs2Filter::New](#) (int *fbandLow_*, int *fbandHigh_*, int *order_*, QObject * *parent* = 0) [static]

10.41.2.4 void [Sbs2Filter::updateFilter](#) (int *order_*, int *fbandLow_*, int *fbandHigh_*)

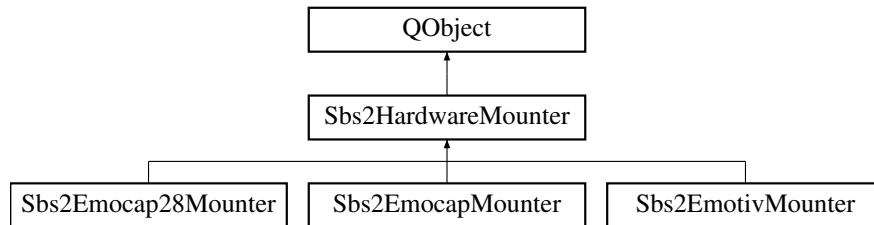
The documentation for this class was generated from the following files:

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filter.-h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filter.-cpp](#)

10.42 Sbs2HardwareMounter Class Reference

```
#include <sbs2hardwaremounter.h>
```

Inheritance diagram for Sbs2HardwareMounter:



Public Slots

- virtual void [start](#) ()=0
- virtual void [stop](#) ()=0
- virtual void [invalidate](#) ()=0

Signals

- void [deviceFound](#) (QMap< QString, QVariant > params)
- void [deviceLost](#) ()

Public Member Functions

- [~Sbs2HardwareMounter](#) ()

Static Public Member Functions

- static QString [getIdentifier](#) ()

Protected Member Functions

- [Sbs2HardwareMounter](#) (QObject *parent=0)
- void [mySleep](#) ()
- virtual void [init](#) ()
- virtual void [mount](#) ()
- virtual void [umount](#) ()
- virtual void [readHardwareParameters](#) ()

Static Protected Attributes

- static QString [mountedHardware](#) = ""
- static QString [identifier](#) = ""

10.42.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics, Cognitive Systems Section](http://code.google.com/p/smartphonebrainscanner2). <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.42.2 Constructor & Destructor Documentation

10.42.2.1 Sbs2HardwareMounter::~~Sbs2HardwareMounter ()

10.42.2.2 Sbs2HardwareMounter::Sbs2HardwareMounter (QObject * *parent* = 0) [protected]

10.42.3 Member Function Documentation

10.42.3.1 void Sbs2HardwareMounter::deviceFound (QMap< QString, QVariant > *params*) [signal]

10.42.3.2 void Sbs2HardwareMounter::deviceLost () [signal]

10.42.3.3 QString Sbs2HardwareMounter::getIdentifier () [static]

10.42.3.4 void Sbs2HardwareMounter::init () [protected],[virtual]

10.42.3.5 virtual void Sbs2HardwareMounter::invalidate () [pure virtual],[slot]

10.42.3.6 void Sbs2HardwareMounter::mount () [protected],[virtual]

10.42.3.7 void Sbs2HardwareMounter::mySleep () [protected]

10.42.3.8 void Sbs2HardwareMounter::readHardwareParameters () [protected],[virtual]

10.42.3.9 void Sbs2HardwareMounter::start () [pure virtual],[slot]

10.42.3.10 virtual void Sbs2HardwareMounter::stop () [pure virtual],[slot]

10.42.3.11 void Sbs2HardwareMounter::umount () [protected],[virtual]

10.42.4 Member Data Documentation

10.42.4.1 QString Sbs2HardwareMounter::identifier = "" [static],[protected]

10.42.4.2 `QString Sbs2HardwareMounter::mountedHardware = ""` `[static], [protected]`

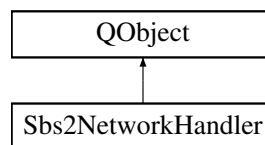
The documentation for this class was generated from the following files:

- [/media/philipjh/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2hardware.h](#)
- [/media/philipjh/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2hardware.cpp](#)

10.43 Sbs2NetworkHandler Class Reference

```
#include <sbs2networkhandler.h>
```

Inheritance diagram for Sbs2NetworkHandler:



Public Slots

- void [turnSendRawDataOn](#) (QString rawDataServerAddress_, int rawDataPort_, int rawDataSize_, int rawDataQueueLength_)
- void [turnSendRawDataOff](#) ()
- void [addRawDataHost](#) (QString address, int port)
- void [removeRawDataHost](#) (QString address, int port)
- void [sendRawData](#) (char *data)
- void [turnReceiveRawDataOn](#) (QString rawDataUdpInputAddress_, int rawDataUdpInputPort_)
- void [turnReceiveRawDataOff](#) ()
- void [readRawData](#) ()
- void [sendMessage](#) (QString message, QString address, int port)
- void [sendMessage](#) (QString message)
- void [addMessageUdpOutputHost](#) (QString address, int port)
- void [removeMessageUdpOutputHost](#) (QString address)
- void [clearMessageUdpOutputHosts](#) ()
- void [turnReceiveMessageOn](#) (QString address, int port)
- void [turnReceiveMessageOff](#) ()
- void [readMessage](#) ()

Signals

- void [rawDataSentSignal](#) ()
- void [rawDataReceived](#) (char *data, int size, QString address, int port)
- void [rawDataReceived](#) (QVector< char * > *data, int counter)
- void [rawDataReceived](#) (QUdpSocket *rawDataUdpInputSocket)
- void [messageReceived](#) (QString data, QString sender, int senderPort)

Public Member Functions

- [Sbs2NetworkHandler](#) (QObject *parent=0)

10.43.1 Constructor & Destructor Documentation

10.43.1.1 Sbs2NetworkHandler::Sbs2NetworkHandler (QObject * *parent* = 0)

10.43.2 Member Function Documentation

10.43.2.1 void Sbs2NetworkHandler::addMessageUdpOutputHost (QString *address*, int *port*) [slot]

10.43.2.2 void Sbs2NetworkHandler::addRawDataHost (QString *address*, int *port*) [slot]

10.43.2.3 void Sbs2NetworkHandler::clearMessageUdpOutputHosts () [slot]

10.43.2.4 void Sbs2NetworkHandler::messageReceived (QString *data*, QString *sender*, int *senderPort*) [signal]

10.43.2.5 void Sbs2NetworkHandler::rawDataReceived (char * *data*, int *size*, QString *address*, int *port*) [signal]

10.43.2.6 void Sbs2NetworkHandler::rawDataReceived (QVector< char * > * *data*, int *counter*) [signal]

10.43.2.7 void Sbs2NetworkHandler::rawDataReceived (QUdpSocket * *rawDataUdpInputSocket*) [signal]

10.43.2.8 void Sbs2NetworkHandler::rawDataSentSignal () [signal]

10.43.2.9 void Sbs2NetworkHandler::readMessage () [slot]

10.43.2.10 void Sbs2NetworkHandler::readRawData () [slot]

10.43.2.11 void Sbs2NetworkHandler::removeMessageUdpOutputHost (QString *address*) [slot]

10.43.2.12 void Sbs2NetworkHandler::removeRawDataHost (QString *address*, int *port*) [slot]

10.43.2.13 void Sbs2NetworkHandler::sendMessage (QString *message*, QString *address*, int *port*) [slot]

10.43.2.14 void Sbs2NetworkHandler::sendMessage (QString *message*) [slot]

10.43.2.15 void Sbs2NetworkHandler::sendRawData (char * *data*) [slot]

10.43.2.16 void Sbs2NetworkHandler::turnReceiveMessageOff () [slot]

10.43.2.17 void Sbs2NetworkHandler::turnReceiveMessageOn (QString *address*, int *port*) [slot]

10.43.2.18 void Sbs2NetworkHandler::turnReceiveRawDataOff () [slot]

10.43.2.19 void Sbs2NetworkHandler::turnReceiveRawDataOn (QString *rawDataUdpInputAddress_*, int *rawDataUdpInputPort_*) [slot]

10.43.2.20 void Sbs2NetworkHandler::turnSendRawDataOff () [slot]

10.43.2.21 void Sbs2NetworkHandler::turnSendRawDataOn (QString *rawDataServerAddress_*, int *rawDataPort_*, int *rawDataSize_*, int *rawDataQueueLength_*) [slot]

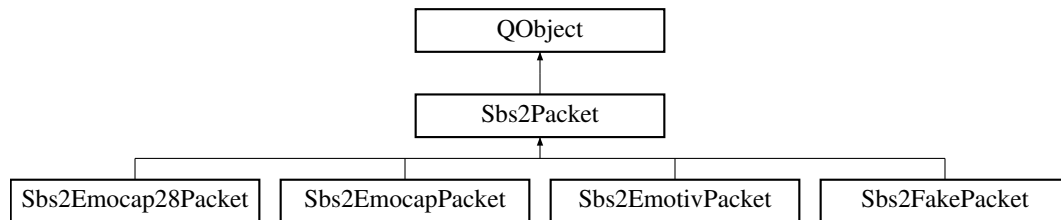
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.h
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.cpp

10.44 Sbs2Packet Class Reference

```
#include <sbs2packet.h>
```

Inheritance diagram for Sbs2Packet:



Public Member Functions

- [Sbs2Packet](#) (QObject *parent=0)
Constructor Constructs empty packet.
- virtual void [update](#) (char *data)
Method for updating data in the packet. To avoid continuous creation and destruction of objects, certain number of empty packets is constructed in initialization and then updated with wrap-around. Packets should see the raw data delivered by [Sbs2DataReader](#) and form themselves.

Public Attributes

- int [counter](#)
- int [gyroX](#)
- int [gyroY](#)
- int [cq](#)
- int [cqIndex](#)
- QString [cqName](#)
- QMap< QString, double > [values](#)
- QMap< QString, double > [filteredValues](#)
- int [battery](#)
- char * [rawData](#)

10.44.1 Constructor & Destructor Documentation

10.44.1.1 [Sbs2Packet::Sbs2Packet \(QObject * parent = 0 \)](#) [explicit]

Constructor Constructs empty packet.

Parameters

<i>parent</i>	Pointer to teh parent QObject object.
---------------	---------------------------------------

10.44.2 Member Function Documentation

10.44.2.1 void [Sbs2Packet::update \(char * data \)](#) [virtual]

Method for updating data in the packet. To avoid continuous creation and destruction of objects, certain number of empty packets is constructed in initialization and then updated with wrap-around. Packets should see the raw data delivered by [Sbs2DataReader](#) and form themselves.

Parameters

<i>data</i>	Pointer to raw data.
-------------	----------------------

Reimplemented in [Sbs2EmocapPacket](#), [Sbs2Emocap28Packet](#), and [Sbs2EmotivPacket](#).

10.44.3 Member Data Documentation

10.44.3.1 int Sbs2Packet::battery

10.44.3.2 int Sbs2Packet::counter

10.44.3.3 int Sbs2Packet::cq

10.44.3.4 int Sbs2Packet::cqIndex

10.44.3.5 QString Sbs2Packet::cqName

10.44.3.6 QMap<QString, double> Sbs2Packet::filteredValues

10.44.3.7 int Sbs2Packet::gyroX

10.44.3.8 int Sbs2Packet::gyroY

10.44.3.9 char* Sbs2Packet::rawData

10.44.3.10 QMap<QString, double> Sbs2Packet::values

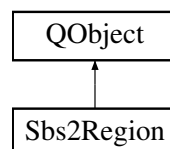
The documentation for this class was generated from the following files:

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2packet.h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2packet.cpp](#)

10.45 Sbs2Region Class Reference

```
#include <sbs2region.h>
```

Inheritance diagram for Sbs2Region:



Public Slots

- void [clearVerticesToExtract](#) ()
- void [addRegion](#) (QString region)
- void [addRegionsIntersection](#) (QString region1, QString region2)

Public Member Functions

- [Sbs2Region](#) (QObject *parent=0)
- QVector< int > * [getVerticesToExtract](#) ()
- QVector< QVector< int > > * [getRegionsToExtract](#) ()

10.45.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.45.2 Constructor & Destructor Documentation

10.45.2.1 `Sbs2Region::Sbs2Region (QObject *parent = 0)` [explicit]

10.45.3 Member Function Documentation

10.45.3.1 `void Sbs2Region::addRegion (QString region)` [slot]

10.45.3.2 `void Sbs2Region::addRegionsIntersection (QString region1, QString region2)` [slot]

10.45.3.3 `void Sbs2Region::clearVerticesToExtract ()` [slot]

10.45.3.4 `QVector< QVector< int > > * Sbs2Region::getRegionsToExtract ()`

10.45.3.5 `QVector< int > * Sbs2Region::getVerticesToExtract ()`

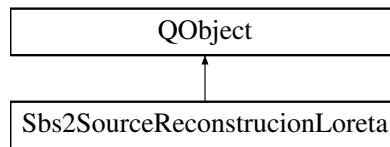
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/[sbs2region.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/[sbs2region.cpp](#)

10.46 Sbs2SourceReconstrucionLoreta Class Reference

```
#include <sbs2sourcereconstruction_loreta.h>
```

Inheritance diagram for Sbs2SourceReconstrucionLoreta:



Public Types

- enum [SumType](#) { [MEAN](#), [POWER](#) }

Public Slots

- void [setSumType](#) ([SumType](#) sumType_)
Sbs2SourceReconstrucion::setSumType.
- void [setMeanExtraction](#) (int enabled)
- void [setAScaling](#) (int scaling)
- void [setVerticesToExtract](#) (QVector< int > *verticesToExtract_)
- void [doRec](#) ([DTU::DtuArray2D](#)< double > *input_, [DTU::DtuArray2D](#)< double > *output_, int *sourceReconstructionReady)
- void [doRecPow](#) ([DTU::DtuArray2D](#)< double > *input_, [DTU::DtuArray2D](#)< double > *output_, int *sourceReconstructionReady)

Public Member Functions

- [Sbs2SourceReconstrucionLoreta](#) (int channels_, int samples_, int samplesDelta_, int vertices_, QString hardware_, QObject *parent_, int modelUpdateLength_=8, int modelUpdateDelta_=24)

Public Attributes

- int [tempModelUpdatedReady](#)

10.46.1 Member Enumeration Documentation

10.46.1.1 enum Sbs2SourceReconstrucionLoreta::SumType

Enumerator

MEAN

POWER

10.46.2 Constructor & Destructor Documentation

- 10.46.2.1 [Sbs2SourceReconstrucionLoreta::Sbs2SourceReconstrucionLoreta](#) (int *channels_*, int *samples_*, int *samplesDelta_*, int *vertices_*, QString *hardware_*, QObject * *parent_*, int *modelUpdateLength_* = 8, int *modelUpdateDelta_* = 24)
[explicit]

10.46.3 Member Function Documentation

- 10.46.3.1 void [Sbs2SourceReconstrucionLoreta::doRec](#) ([DTU::DtuArray2D](#)< double > * *input_*, [DTU::DtuArray2D](#)< double > * *output_*, int * *sourceReconstructionReady*) [slot]

10.46.3.2 void Sbs2SourceReconstrucionLoreta::doRecPow (DTU::DtuArray2D< double > * *input_*, DTU::DtuArray2D< double > * *output_*, int * *sourceReconstructionReady*) [slot]

10.46.3.3 void Sbs2SourceReconstrucionLoreta::setAScaling (int *scaling*) [slot]

10.46.3.4 void Sbs2SourceReconstrucionLoreta::setMeanExtraction (int *enabled*) [slot]

10.46.3.5 void Sbs2SourceReconstrucionLoreta::setSumType (SumType *sumType_*) [slot]

Sbs2SourceReconstrucion::setSumType.

Parameters

<i>sumType_</i>	should be either 'MEAN' or 'SUM'
-----------------	----------------------------------

Set the 'sumType' variable

10.46.3.6 void Sbs2SourceReconstrucionLoreta::setVerticesToExtract (QVector< int > * *verticesToExtract_*) [slot]

10.46.4 Member Data Documentation

10.46.4.1 int Sbs2SourceReconstrucionLoreta::tempModelUpdatedReady

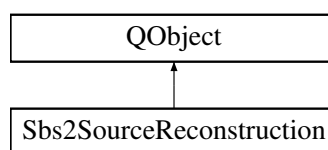
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-reconstruction/loreta/[sbs2sourceconstruction_loreta.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source-reconstruction/loreta/[sbs2sourceconstruction_loreta.cpp](#)

10.47 Sbs2SourceReconstruction Class Reference

```
#include <sbs2sourceconstruction.h>
```

Inheritance diagram for Sbs2SourceReconstruction:



Public Slots

- void [turnOnLoreta](#) (int *sourceReconstructionSamples*, int *sourceReconstructionDelta*, int *sourceReconstructionModelUpdateLength*, int *sourceReconstructionModelUpdateDelta*, QString *hardware*, QString *sourceReconstructionMethod_*)
- void [turnOnSparse](#) (int *sourceReconstructionSamples*, QString *hardware*, QVector< double > *lambdas*, QString *sourceReconstructionMethod_*)
- void [doReconstruction](#) (DTU::DtuArray2D< double > **input_*, DTU::DtuArray2D< double > **output_*, int **sourceReconstructionReady*)
- void [doReconstructionSpectrogram](#) (DTU::DtuArray2D< double > **input_*, DTU::DtuArray2D< double > **output_*, int **sourceReconstructionReady*)
- void [stopReconstruction](#) ()
- void [turnOff](#) ()

Public Member Functions

- [Sbs2SourceReconstruction](#) (QObject *parent=0)

10.47.1 Constructor & Destructor Documentation

10.47.1.1 `Sbs2SourceReconstruction::Sbs2SourceReconstruction (QObject * parent = 0) [explicit]`

10.47.2 Member Function Documentation

10.47.2.1 `void Sbs2SourceReconstruction::doReconstruction (DTU::DtuArray2D< double > * input_, DTU::DtuArray2D< double > * output_, int * sourceReconstructionReady) [slot]`

10.47.2.2 `void Sbs2SourceReconstruction::doReconstructionSpectrogram (DTU::DtuArray2D< double > * input_, DTU::DtuArray2D< double > * output_, int * sourceReconstructionReady) [slot]`

10.47.2.3 `void Sbs2SourceReconstruction::stopReconstruction () [slot]`

10.47.2.4 `void Sbs2SourceReconstruction::turnOff () [slot]`

10.47.2.5 `void Sbs2SourceReconstruction::turnOnLoreta (int sourceReconstructionSamples, int sourceReconstructionDelta, int sourceReconstructionModelUpdateLength, int sourceReconstructionModelUpdateDelta, QString hardware, QString sourceReconstructionMethod_) [slot]`

10.47.2.6 `void Sbs2SourceReconstruction::turnOnSparse (int sourceReconstructionSamples, QString hardware, QVector< double > lambdas, QString sourceReconstructionMethod_) [slot]`

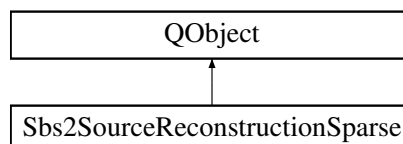
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/[sbs2sourcereconstruction.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/[sbs2sourcereconstruction.cpp](#)

10.48 Sbs2SourceReconstructionSparse Class Reference

```
#include <sbs2sourcereconstruction_sparse.h>
```

Inheritance diagram for Sbs2SourceReconstructionSparse:



Public Member Functions

- [Sbs2SourceReconstructionSparse](#) (int channels_input, int sources_input, int samples_, QVector< double > lambdas_, QString hardware_, QObject *parent=0, int numDatosTrain_input=8, int numDatosTest_input=6, double error_tol_=0.0001)
- `void f_objective_general_group_lasso (DTU::DtuArray2D< double > *A_normalized, DTU::DtuArray2D< double > *S, DTU::DtuArray2D< double > *Y, double lambda, double *out)`

- void `derivative_square_loss_frobenius` (DTU::DtuArray2D< double > *A, DTU::DtuArray2D< double > *Y, DTU::DtuArray2D< double > *S, DTU::DtuArray2D< double > *out)
- void `proximal_operator_standard_group_lasso` (DTU::DtuArray2D< double > *X, double *regularizer_factor, DTU::DtuArray2D< double > *out)
- void `fista_method_group_lasso_v2` (DTU::DtuArray2D< double > *A_normalized, DTU::DtuArray2D< double > *Y, double lambda, double L, DTU::DtuArray2D< double > *estimated_S)
- void `cross_validation_k_channel` (DTU::DtuArray2D< double > *Y_mean_0, DTU::DtuArray2D< double > *estimated_S)
- void `rootMeanSquareError` (DTU::DtuArray2D< double > *Y_mean_0_test, DTU::DtuArray2D< double > *A_normalized_test, DTU::DtuArray2D< double > *estimated_S, double *rmse)
- void `doRec` (DTU::DtuArray2D< double > *Y_input, DTU::DtuArray2D< double > *S_output, int *isSourceReconstructionReady)
- void `doRecPow` (DTU::DtuArray2D< double > *Y_input, DTU::DtuArray2D< double > *S_output, int *isSourceReconstructionReady)
- void `preprocessData` ()
- void `calculateMean` (DTU::DtuArray2D< double > *input, DTU::DtuArray2D< double > *output)
- void `calculatePower` (DTU::DtuArray2D< double > *input, DTU::DtuArray2D< double > *output)

10.48.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, DTU Informatics, Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.48.2 Constructor & Destructor Documentation

- 10.48.2.1 `Sbs2SourceReconstructionSparse::Sbs2SourceReconstructionSparse (int channels_input, int sources_input, int samples_, QVector< double > lambdas_, QString hardware_, QObject * parent = 0, int numDatosTrain_input = 8, int numDatosTest_input = 6, double error_tol_ = 0.0001) [explicit]`

Variables used in the function `fista_method_group_lasso_v2`

10.48.3 Member Function Documentation

- 10.48.3.1 `void Sbs2SourceReconstructionSparse::calculateMean (DTU::DtuArray2D< double > * input, DTU::DtuArray2D< double > * output)`
- 10.48.3.2 `void Sbs2SourceReconstructionSparse::calculatePower (DTU::DtuArray2D< double > * input, DTU::DtuArray2D< double > * output)`

- 10.48.3.3 void Sbs2SourceReconstructionSparse::cross_validation_k_channel (DTU::DtuArray2D< double > * *Y_mean_0*, DTU::DtuArray2D< double > * *estimated_S*)
- 10.48.3.4 void Sbs2SourceReconstructionSparse::derivative_square_loss_frobenius (DTU::DtuArray2D< double > * *A*, DTU::DtuArray2D< double > * *Y*, DTU::DtuArray2D< double > * *S*, DTU::DtuArray2D< double > * *out*)
- 10.48.3.5 void Sbs2SourceReconstructionSparse::doRec (DTU::DtuArray2D< double > * *Y_input*, DTU::DtuArray2D< double > * *S_output*, int * *isSourceReconstructionReady*)
- 10.48.3.6 void Sbs2SourceReconstructionSparse::doRecPow (DTU::DtuArray2D< double > * *Y_input*, DTU::DtuArray2D< double > * *S_output*, int * *isSourceReconstructionReady*)
- 10.48.3.7 void Sbs2SourceReconstructionSparse::f_objective_general_group_lasso (DTU::DtuArray2D< double > * *A_normalized*, DTU::DtuArray2D< double > * *S*, DTU::DtuArray2D< double > * *Y*, double *lambda*, double * *out*)
- 10.48.3.8 void Sbs2SourceReconstructionSparse::fista_method_group_lasso_v2 (DTU::DtuArray2D< double > * *A_normalized*, DTU::DtuArray2D< double > * *Y*, double *lambda*, double *L*, DTU::DtuArray2D< double > * *estimated_S*)
- 10.48.3.9 void Sbs2SourceReconstructionSparse::preprocessData ()
- 10.48.3.10 void Sbs2SourceReconstructionSparse::proximal_operator_standard_group_lasso (DTU::DtuArray2D< double > * *X*, double * *regularizer_factor*, DTU::DtuArray2D< double > * *out*)
- 10.48.3.11 void Sbs2SourceReconstructionSparse::rootMeanSquareError (DTU::DtuArray2D< double > * *Y_mean_0_test*, DTU::DtuArray2D< double > * *A_normalized_test*, DTU::DtuArray2D< double > * *estimated_S*, double * *rmse*)

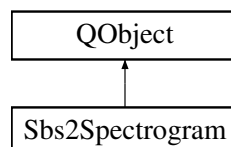
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/sbs2sourcereconstruction_sparse.h
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/sbs2sourcereconstruction_sparse.cpp

10.49 Sbs2Spectrogram Class Reference

```
#include <sbs2spectrogram.h>
```

Inheritance diagram for Sbs2Spectrogram:



Public Types

- enum [WindowType](#) { [RECT](#), [HANN](#), [HAMMING](#) }

Public Slots

- void [setWindowType](#) ([Sbs2Spectrogram::WindowType](#) *windowType_*)

Public Member Functions

- [Sbs2Spectrogram](#) (int *length_*, QObject **parent*=0)
- void [doSpectrogram](#) ([DTU::DtuArray2D](#)< double > **input*, [DTU::DtuArray2D](#)< double > **output*)
- [~Sbs2Spectrogram](#) ()
- [WindowType](#) [getWindowType](#) ()

10.49.1 Member Enumeration Documentation

10.49.1.1 enum [Sbs2Spectrogram::WindowType](#)

Enumerator

RECT

HANN

HAMMING

10.49.2 Constructor & Destructor Documentation

10.49.2.1 [Sbs2Spectrogram::Sbs2Spectrogram](#) (int *length_*, QObject **parent* = 0) [explicit]

10.49.2.2 [Sbs2Spectrogram::~~Sbs2Spectrogram](#) ()

10.49.3 Member Function Documentation

10.49.3.1 void [Sbs2Spectrogram::doSpectrogram](#) ([DTU::DtuArray2D](#)< double > **input*, [DTU::DtuArray2D](#)< double > **output*)

10.49.3.2 [Sbs2Spectrogram::WindowType](#) [Sbs2Spectrogram::getWindowType](#) ()

10.49.3.3 void [Sbs2Spectrogram::setWindowType](#) ([Sbs2Spectrogram::WindowType](#) *windowType_*) [slot]

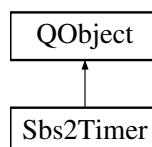
The documentation for this class was generated from the following files:

- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/[sbs2spectrogram.h](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/[sbs2spectrogram.cpp](#)

10.50 Sbs2Timer Class Reference

```
#include <sbs2timer.h>
```

Inheritance diagram for [Sbs2Timer](#):



Public Member Functions

- [Sbs2Timer](#) (QObject *parent=0)

Static Public Member Functions

- static void [tic](#) ()
- static void [tic](#) (QString label_)
- static void [toc](#) ()

Static Public Attributes

- static qint64 [tic_time](#) = 0

10.50.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU](#) Informatics, Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.50.2 Constructor & Destructor Documentation

10.50.2.1 [Sbs2Timer::Sbs2Timer](#) (QObject * *parent* = 0) [explicit]

10.50.3 Member Function Documentation

10.50.3.1 void [Sbs2Timer::tic](#) () [static]

10.50.3.2 void [Sbs2Timer::tic](#) (QString *label_*) [static]

10.50.3.3 void [Sbs2Timer::toc](#) () [static]

10.50.4 Member Data Documentation

10.50.4.1 qint64 [Sbs2Timer::tic_time](#) = 0 [static]

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utils/sbs2timer.h](#)
- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utils/sbs2timer.cpp](#)

10.51 TNT::Sparse_Matrix_CompRow< T > Class Template Reference

```
#include <tnt_sparse_matrix_csr.h>
```

Public Member Functions

- [Sparse_Matrix_CompRow](#) (const [Sparse_Matrix_CompRow](#) &S)
- [Sparse_Matrix_CompRow](#) (int M, int N, int nz, const T **val*, const int **r*, const int **c*)
- const T & *val* (int i) const
- const int & *row_ptr* (int i) const
- const int & *col_ind* (int i) const
- int *dim1* () const
- int *dim2* () const
- int *NumNonzeros* () const
- [Sparse_Matrix_CompRow](#) & *operator=* (const [Sparse_Matrix_CompRow](#) &R)

10.51.1 Detailed Description

```
template<class T>class TNT::Sparse_Matrix_CompRow< T >
```

Read-only view of a sparse matrix in compressed-row storage format. Neither array elements (nonzeros) nor sparsity structure can be modified. If modifications are required, create a new view.

Index values begin at 0.

Storage requirements: An (m x n) matrix with nz nonzeros requires no more than ((T+I)*nz + M*I) bytes, where T is the size of data elements and I is the size of integers.

10.51.2 Constructor & Destructor Documentation

10.51.2.1 `template<class T> TNT::Sparse_Matrix_CompRow< T >::Sparse_Matrix_CompRow (const Sparse_Matrix_CompRow< T > & S)`

10.51.2.2 `template<class T> TNT::Sparse_Matrix_CompRow< T >::Sparse_Matrix_CompRow (int M, int N, int nz, const T * val, const int * r, const int * c)`

Construct a read-only view of existing sparse matrix in compressed-row storage format.

Parameters

<i>M</i>	the number of rows of sparse matrix
<i>N</i>	the number of columns of sparse matrix
<i>nz</i>	the number of nonzeros
<i>val</i>	a contiguous list of nonzero values
<i>r</i>	row-pointers: <i>r</i> [<i>i</i>] denotes the beginning position of row <i>i</i> (i.e. the <i>i</i> th row begins at <i>val</i> [<i>row</i> [<i>i</i>]]).

<code>c</code>	column-indices: <code>c[i]</code> denotes the column location of <code>val[i]</code>
----------------	--

10.51.3 Member Function Documentation

10.51.3.1 `template<class T> const int& TNT::Sparse_Matrix_CompRow< T>::col_ind (int i) const [inline]`

10.51.3.2 `template<class T> int TNT::Sparse_Matrix_CompRow< T>::dim1 () const [inline]`

10.51.3.3 `template<class T> int TNT::Sparse_Matrix_CompRow< T>::dim2 () const [inline]`

10.51.3.4 `template<class T> int TNT::Sparse_Matrix_CompRow< T>::NumNonzeros () const [inline]`

10.51.3.5 `template<class T> Sparse_Matrix_CompRow& TNT::Sparse_Matrix_CompRow< T>::operator= (const Sparse_Matrix_CompRow< T> & R)`

10.51.3.6 `template<class T> const int& TNT::Sparse_Matrix_CompRow< T>::row_ptr (int i) const [inline]`

10.51.3.7 `template<class T> const T& TNT::Sparse_Matrix_CompRow< T>::val (int i) const [inline]`

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

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-_sparse_matrix_csr.h](#)

10.52 TNT::Stopwatch Class Reference

```
#include <tnt_stopwatch.h>
```

Public Member Functions

- [Stopwatch](#) ()
- void [start](#) ()
- double [stop](#) ()
- double [read](#) ()
- void [resume](#) ()
- int [running](#) ()

10.52.1 Constructor & Destructor Documentation

10.52.1.1 `TNT::Stopwatch::Stopwatch () [inline]`

10.52.2 Member Function Documentation

10.52.2.1 `double TNT::Stopwatch::read () [inline]`

10.52.2.2 `void TNT::Stopwatch::resume () [inline]`

10.52.2.3 `int TNT::Stopwatch::running () [inline]`

10.52.2.4 `void TNT::Stopwatch::start () [inline]`

10.52.2.5 `double TNT::Stopwatch::stop () [inline]`

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

- [/media/philipjhj/Data/OneDrive/Studie/Studententerprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-_stopwatch.h](#)

10.53 JAMA::SVD< Real > Class Template Reference

```
#include <jama_svd.h>
```

Public Member Functions

- [SVD](#) (const [Array2D](#)< Real > &Arg)
- void [getU](#) ([Array2D](#)< Real > &A)
- void [getV](#) ([Array2D](#)< Real > &A)
- void [getSingularValues](#) ([Array1D](#)< Real > &x)
- void [getS](#) ([Array2D](#)< Real > &A)
- Real [norm2](#) ()
- Real [cond](#) ()
- int [rank](#) ()

10.53.1 Detailed Description

```
template<class Real>class JAMA::SVD< Real >
```

Singular Value Decomposition.

For an m-by-n matrix A with $m \geq n$, the singular value decomposition is an m-by-n orthogonal matrix U, an n-by-n diagonal matrix S, and an n-by-n orthogonal matrix V so that $A = U \cdot S \cdot V'$.

The singular values, $\sigma[k] = S[k][k]$, are ordered so that $\sigma[0] \geq \sigma[1] \geq \dots \geq \sigma[n-1]$.

The singular value decomposition always exists, so the constructor will never fail. The matrix condition number and the effective numerical rank can be computed from this decomposition.

(Adapted from [JAMA](#), a Java Matrix Library, developed by jointly by the Mathworks and NIST; see <http://math.nist.gov/javanumerics/jama>).

10.53.2 Constructor & Destructor Documentation

10.53.2.1 `template<class Real> JAMA::SVD< Real >::SVD (const Array2D< Real > & Arg) [inline]`

10.53.3 Member Function Documentation

10.53.3.1 `template<class Real> Real JAMA::SVD< Real >::cond () [inline]`

Two norm of condition number ($\max(S)/\min(S)$)

10.53.3.2 `template<class Real> void JAMA::SVD< Real >::getS (Array2D< Real > & A) [inline]`

Return the diagonal matrix of singular values

Returns

S

10.53.3.3 `template<class Real> void JAMA::SVD< Real >::getSingularValues (Array1D< Real > & x) [inline]`

Return the one-dimensional array of singular values

10.53.3.4 `template<class Real> void JAMA::SVD< Real >::getU (Array2D< Real > & A) [inline]`

10.53.3.5 `template<class Real> void JAMA::SVD< Real >::getV (Array2D< Real > & A) [inline]`

10.53.3.6 `template<class Real> Real JAMA::SVD< Real >::norm2 () [inline]`

Two norm (max(S))

10.53.3.7 `template<class Real> int JAMA::SVD< Real >::rank () [inline]`

Effective numerical matrix rank

Returns

Number of nonnegligible singular values.

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

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama-svd.h](#)

10.54 TNT::Vector< T > Class Template Reference

```
#include <tnt_vec.h>
```

Public Types

- typedef [Subscript](#) `size_type`
- typedef T `value_type`
- typedef T `element_type`
- typedef T * `pointer`
- typedef T * `iterator`
- typedef T & `reference`
- typedef const T * `const_iterator`
- typedef const T & `const_reference`

Public Member Functions

- [Subscript](#) `lbound () const`
- [iterator](#) `begin ()`
- [iterator](#) `end ()`
- const [iterator](#) `begin () const`
- const [iterator](#) `end () const`
- [~Vector](#) `()`

- [Vector](#) ()
- [Vector](#) (const [Vector](#)< T > &A)
- [Vector](#) ([Subscript](#) N, const T &value=T())
- [Vector](#) ([Subscript](#) N, const T *v)
- [Vector](#) ([Subscript](#) N, char *s)
- [Vector](#)< T > & [newsize](#) ([Subscript](#) N)
- [Vector](#)< T > & [operator=](#) (const [Vector](#)< T > &A)
- [Vector](#)< T > & [operator=](#) (const T &scalar)
- [Subscript](#) dim () const
- [Subscript](#) size () const
- [reference operator](#)() ([Subscript](#) i)
- [const_reference operator](#)() ([Subscript](#) i) const
- [reference operator](#)[] ([Subscript](#) i)
- [const_reference operator](#)[] ([Subscript](#) i) const

Protected Member Functions

- void [initialize](#) ([Subscript](#) N)
- void [copy](#) (const T *v)
- void [set](#) (const T &val)
- void [destroy](#) ()

Protected Attributes

- T * [v_](#)
- T * [vm1_](#)
- [Subscript](#) [n_](#)

10.54.1 Detailed Description

template<class T>class TNT::Vector< T >

[Deprecated] Value-based vector class from pre-1.0 [TNT](#) version. Kept here for backward compatibility, but should use the newer [TNT::Array1D](#) classes instead.

10.54.2 Member Typedef Documentation

10.54.2.1 template<class T> typedef const T* TNT::Vector< T >::const_iterator

10.54.2.2 template<class T> typedef const T& TNT::Vector< T >::const_reference

10.54.2.3 template<class T> typedef T TNT::Vector< T >::element_type

10.54.2.4 template<class T> typedef T* TNT::Vector< T >::iterator

10.54.2.5 template<class T> typedef T* TNT::Vector< T >::pointer

10.54.2.6 template<class T> typedef T& TNT::Vector< T >::reference

10.54.2.7 template<class T> typedef [Subscript](#) TNT::Vector< T >::size_type

10.54.2.8 template<class T> typedef T TNT::Vector< T >::value_type

10.54.3 Constructor & Destructor Documentation

10.54.3.1 `template<class T> TNT::Vector< T >::~~Vector () [inline]`

10.54.3.2 `template<class T> TNT::Vector< T >::Vector () [inline]`

10.54.3.3 `template<class T> TNT::Vector< T >::Vector (const Vector< T > & A) [inline]`

10.54.3.4 `template<class T> TNT::Vector< T >::Vector (Subscript N, const T & value = T()) [inline]`

10.54.3.5 `template<class T> TNT::Vector< T >::Vector (Subscript N, const T * v) [inline]`

10.54.3.6 `template<class T> TNT::Vector< T >::Vector (Subscript N, char * s) [inline]`

10.54.4 Member Function Documentation

10.54.4.1 `template<class T> iterator TNT::Vector< T >::begin () [inline]`

10.54.4.2 `template<class T> const iterator TNT::Vector< T >::begin () const [inline]`

10.54.4.3 `template<class T> void TNT::Vector< T >::copy (const T * v) [inline], [protected]`

10.54.4.4 `template<class T> void TNT::Vector< T >::destroy () [inline], [protected]`

10.54.4.5 `template<class T> Subscript TNT::Vector< T >::dim () const [inline]`

10.54.4.6 `template<class T> iterator TNT::Vector< T >::end () [inline]`

10.54.4.7 `template<class T> const iterator TNT::Vector< T >::end () const [inline]`

10.54.4.8 `template<class T> void TNT::Vector< T >::initialize (Subscript N) [inline], [protected]`

10.54.4.9 `template<class T> Subscript TNT::Vector< T >::lbound () const [inline]`

10.54.4.10 `template<class T> Vector< T > & TNT::Vector< T >::newsize (Subscript N) [inline]`

10.54.4.11 `template<class T> reference TNT::Vector< T >::operator() (Subscript i) [inline]`

10.54.4.12 `template<class T> const_reference TNT::Vector< T >::operator() (Subscript i) const [inline]`

10.54.4.13 `template<class T> Vector< T > & TNT::Vector< T >::operator= (const Vector< T > & A) [inline]`

10.54.4.14 `template<class T> Vector< T > & TNT::Vector< T >::operator= (const T & scalar) [inline]`

10.54.4.15 `template<class T> reference TNT::Vector< T >::operator[] (Subscript i) [inline]`

10.54.4.16 `template<class T> const_reference TNT::Vector< T >::operator[] (Subscript i) const [inline]`

10.54.4.17 `template<class T> void TNT::Vector< T >::set (const T & val) [inline], [protected]`

10.54.4.18 `template<class T> Subscript TNT::Vector< T >::size () const [inline]`

10.54.5 Member Data Documentation

10.54.5.1 `template<class T> Subscript TNT::Vector< T >::n_ [protected]`

10.54.5.2 `template<class T> T* TNT::Vector< T >::v_` [protected]

10.54.5.3 `template<class T> T* TNT::Vector< T >::vm1_` [protected]

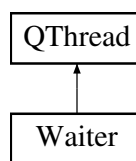
The documentation for this class was generated from the following file:

- [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt-_vec.h](#)

10.55 Waiter Class Reference

```
#include <waiter.h>
```

Inheritance diagram for Waiter:



Public Member Functions

- [Waiter](#) (long msec)
- void [run](#) ()

10.55.1 Detailed Description

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.55.2 Constructor & Destructor Documentation

10.55.2.1 `Waiter::Waiter (long msec)` [inline]

10.55.3 Member Function Documentation

10.55.3.1 void Waiter::run () [inline]

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

- [/media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/utils/waiter.-h](#)

Chapter 11

File Documentation

11.1 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/documentation_static.cpp File Reference](#)

11.2 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/dtu_array_2d.h File Reference](#)

```
#include <math.h>
#include "jama125/tnt_array2d.h"
#include "jama125/tnt_array2d_utils.h"
#include "jama125/jama_svd.h"
```

Classes

- class [DTU::DtuArray2D< T >](#)

Namespaces

- [DTU](#)

11.3 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/FFTReal.cpp File Reference](#)

```
#include "FFTReal.h"
#include <cassert>
#include <cmath>
```

11.4 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/FFTReal.h File Reference](#)

Classes

- class [FFTReal](#)

Macros

- `#define FFTReal_CURRENT_HEADER`
- `#define FFTReal_HEADER_INCLUDED`

Typedefs

- `typedef double flt_t`

11.4.1 Macro Definition Documentation

11.4.1.1 `#define FFTReal_CURRENT_HEADER`

11.4.1.2 `#define FFTReal_HEADER_INCLUDED`

11.4.2 Typedef Documentation

11.4.2.1 `typedef double flt_t`

11.5 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocapdatareader.cpp File Reference

```
#include <hardware/emocap/sbs2emocapdatareader.h>
```

11.6 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocapdatareader.h File Reference

```
#include <hardware/sbs2datareader.h>
#include <hardware/emocap/sbs2emocapmounter.h>
#include <hardware/emotiv/sbs2emotivdecryptor.h>
#include <hardware/emocap/sbs2emocappacket.h>
```

Classes

- class [Sbs2EmocapDataReader](#)

11.7 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocapmounter.cpp File Reference

```
#include <hardware/emocap/sbs2emocapmounter.h>
```


11.8 /media/philipjhj/Data/OneDrive/Studie/Studentprogramør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocapmounter.h File Reference

```
#include <hardware/sbs2hardwaremounter.h>
```

Classes

- class [Sbs2EmocapMounter](#)

11.9 /media/philipjhj/Data/OneDrive/Studie/Studentprogramør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocappacket.cpp File Reference

```
#include <hardware/emocap/sbs2emocappacket.h>
```

11.10 /media/philipjhj/Data/OneDrive/Studie/Studentprogramør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs2emocappacket.h File Reference

```
#include <hardware/sbs2packet.h>
```

Classes

- class [Sbs2EmocapPacket](#)

11.11 /media/philipjhj/Data/OneDrive/Studie/Studentprogramør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28datareader.cpp File Reference

```
#include <hardware/emocap28/sbs2emocap28datareader.h>
```

Functions

- int [mod](#) (int *x*, int *m*)
- bool [lessThan](#) (const QVector< double > &*s1*, const QVector< double > &*s2*)

11.11.1 Function Documentation

11.11.1.1 bool [lessThan](#) (const QVector< double > & *s1*, const QVector< double > & *s2*)

11.11.1.2 int [mod](#) (int *x*, int *m*)

11.12 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28datareader.h File Reference

```
#include <hardware/sbs2datareader.h>
#include <hardware/emocap28/sbs2emocap28mounter.h>
#include <hardware/emotiv/sbs2emotivdecryptor.h>
#include <hardware/emocap28/sbs2emocap28packet.h>
```

Classes

- class [Sbs2Emocap28DataContainer](#)
- class [Sbs2Emocap28DataReader](#)

11.13 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28mounter.cpp File Reference

```
#include <hardware/emocap28/sbs2emocap28mounter.h>
```

11.14 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28mounter.h File Reference

```
#include <hardware/sbs2hardwaremounter.h>
```

Classes

- class [Sbs2Emocap28Mounter](#)

11.15 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28packet.cpp File Reference

```
#include <hardware/emocap28/sbs2emocap28packet.h>
```

11.16 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/sbs2emocap28packet.h File Reference

```
#include <hardware/sbs2packet.h>
```

Classes

- class [Sbs2Emocap28Packet](#)

11.17 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivdatareader.cpp File

Reference

127

11.17 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivdatareader.cpp File Reference

```
#include <hardware/emotiv/sbs2emotivdatareader.h>
```

11.18 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivdatareader.h File Reference

```
#include <hardware/sbs2datareader.h>
#include <hardware/emotiv/sbs2emotivmounter.h>
#include <hardware/emotiv/sbs2emotivdecryptor.h>
#include <hardware/emotiv/sbs2emotivpacket.h>
```

Classes

- class [Sbs2EmotivDataReader](#)

11.19 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivdecryptor.h File Reference

```
#include <QObject>
#include <utils/Rijndael.h>
#include <QDebug>
```

Classes

- class [Sbs2EmotivDecryptor](#)

11.20 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivdecryptor_dummy.cpp File Reference

```
#include "sbs2emotivdecryptor.h"
```

11.21 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivmounter.cpp File Reference

```
#include <hardware/emotiv/sbs2emotivmounter.h>
#include "QFile"
```

11.22 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivmounter.h File Reference

```
#include <hardware/sbs2hardwaremounter.h>
```

Classes

- class [Sbs2EmotivMounter](#)

11.23 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivpacket.cpp File Reference

```
#include <hardware/emotiv/sbs2emotivpacket.h>
```

11.24 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/emotiv/sbs2emotivpacket.h File Reference

```
#include <hardware/sbs2packet.h>
```

Classes

- class [Sbs2EmotivPacket](#)

11.25 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakedatareader.cpp File Reference

```
#include "sbs2fakedatareader.h"  
#include <QtCore>
```

11.26 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakedatareader.h File Reference

```
#include <hardware/sbs2datareader.h>  
#include <hardware/fake/sbs2fakepacket.h>
```

Classes

- class [Sbs2FakeDataReader](#)

11.27 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakepacket.cpp File

Reference

129

11.27 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakepacket.cpp File Reference

```
#include "sbs2fakepacket.h"
```

11.28 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/fake/sbs2fakepacket.h File Reference

```
#include <hardware/sbs2packet.h>
```

Classes

- class [Sbs2FakePacket](#)

11.29 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2datareader.cpp File Reference

```
#include "sbs2datareader.h"
```

11.30 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2datareader.h File Reference

```
#include <QObject>
#include <fstream>
#include <sbs2callback.h>
#include <sbs2networkhandler.h>
```

Classes

- class [Sbs2DataReader](#)

11.31 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2hardwaremounter.cpp File Reference

```
#include <hardware/sbs2hardwaremounter.h>
```

11.32 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2hardwaremounter.h File Reference

```
#include <QObject>
#include <QTimer>
#include <QtCore>
#include <sbs2common.h>
#include <utils/waiter.h>
#include <QString>
```

Classes

- class [Sbs2HardwareMounter](#)

11.33 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2packet.cpp File Reference

```
#include "sbs2packet.h"
```

11.34 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/hardware/sbs2packet.h File Reference

```
#include <QObject>
#include <QMap>
#include <sbs2common.h>
```

Classes

- class [Sbs2Packet](#)

11.35 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama_cholesky.h File Reference

```
#include "math.h"
```

Classes

- class [JAMA::Cholesky< Real >](#)

Namespaces

- [JAMA](#)

11.36 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2- core/src/jama125/jama_eig.h File Reference

```
#include "tnt_array1d.h"  
#include "tnt_array2d.h"  
#include "tnt_math_utils.h"  
#include <algorithm>  
#include <cmath>
```

Classes

- class [JAMA::Eigenvalue< Real >](#)

Namespaces

- [JAMA](#)

11.37 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2- core/src/jama125/jama_lu.h File Reference

```
#include "tnt.h"  
#include <algorithm>
```

Classes

- class [JAMA::LU< Real >](#)

Namespaces

- [JAMA](#)

11.38 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2- core/src/jama125/jama_qr.h File Reference

```
#include "tnt_array1d.h"  
#include "tnt_array2d.h"  
#include "tnt_math_utils.h"
```

Classes

- class [JAMA::QR< Real >](#)

Namespaces

- [JAMA](#)

11.39 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/jama_svd.h File Reference

```
#include "tnt_array1d.h"
#include "tnt_array1d_utils.h"
#include "tnt_array2d.h"
#include "tnt_array2d_utils.h"
#include "tnt_math_utils.h"
#include <algorithm>
#include <cmath>
```

Classes

- class [JAMA::SVD](#) < [Real](#) >

Namespaces

- [JAMA](#)

11.40 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt.h File Reference

```
#include "tnt_version.h"
#include "tnt_math_utils.h"
#include "tnt_array1d.h"
#include "tnt_array2d.h"
#include "tnt_array3d.h"
#include "tnt_array1d_utils.h"
#include "tnt_array2d_utils.h"
#include "tnt_array3d_utils.h"
#include "tnt_fortran_array1d.h"
#include "tnt_fortran_array2d.h"
#include "tnt_fortran_array3d.h"
#include "tnt_fortran_array1d_utils.h"
#include "tnt_fortran_array2d_utils.h"
#include "tnt_fortran_array3d_utils.h"
#include "tnt_sparse_matrix_csr.h"
#include "tnt_stopwatch.h"
#include "tnt_subscript.h"
#include "tnt_vec.h"
#include "tnt_cmat.h"
```

11.41 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array1d.h File Reference

```
#include <iostream>
#include "tnt_i_refvec.h"
```


- class [TNT::Array1D< T >](#)

Namespaces

- [TNT](#)

11.42 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array1d_utils.h File Reference

```
#include <cstdlib>
#include <cassert>
```

Namespaces

- [TNT](#)

Functions

- template<class T >
std::ostream & [TNT::operator<<](#) (std::ostream &s, const Array1D< T > &A)
- template<class T >
std::istream & [TNT::operator>>](#) (std::istream &s, Array1D< T > &A)
- template<class T >
Array1D< T > [TNT::operator+](#) (const Array1D< T > &A, const Array1D< T > &B)
- template<class T >
Array1D< T > [TNT::operator-](#) (const Array1D< T > &A, const Array1D< T > &B)
- template<class T >
Array1D< T > [TNT::operator*](#) (const Array1D< T > &A, const Array1D< T > &B)
- template<class T >
Array1D< T > [TNT::operator/](#) (const Array1D< T > &A, const Array1D< T > &B)
- template<class T >
Array1D< T > & [TNT::operator+=](#) (Array1D< T > &A, const Array1D< T > &B)
- template<class T >
Array1D< T > & [TNT::operator-=](#) (Array1D< T > &A, const Array1D< T > &B)
- template<class T >
Array1D< T > & [TNT::operator*=](#) (Array1D< T > &A, const Array1D< T > &B)
- template<class T >
Array1D< T > & [TNT::operator/=](#) (Array1D< T > &A, const Array1D< T > &B)

11.43 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array2d.h File Reference

```
#include <cstdlib>
#include <iostream>
#include "tnt_array1d.h"
```

Classes

- class [TNT::Array2D< T >](#)

Namespaces

- [TNT](#)

11.44 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array2d_utils.h File Reference

```
#include <cstdlib>
#include <cassert>
```

Namespaces

- [TNT](#)

Functions

- template<class T >
std::ostream & [TNT::operator<<](#) (std::ostream &s, const Array2D< T > &A)
- template<class T >
std::istream & [TNT::operator>>](#) (std::istream &s, Array2D< T > &A)
- template<class T >
Array2D< T > [TNT::operator+](#) (const Array2D< T > &A, const Array2D< T > &B)
- template<class T >
Array2D< T > [TNT::operator-](#) (const Array2D< T > &A, const Array2D< T > &B)
- template<class T >
Array2D< T > [TNT::operator*](#) (const Array2D< T > &A, const Array2D< T > &B)
- template<class T >
Array2D< T > [TNT::operator/](#) (const Array2D< T > &A, const Array2D< T > &B)
- template<class T >
Array2D< T > & [TNT::operator+=](#) (Array2D< T > &A, const Array2D< T > &B)
- template<class T >
Array2D< T > & [TNT::operator-=](#) (Array2D< T > &A, const Array2D< T > &B)
- template<class T >
Array2D< T > & [TNT::operator*=](#) (Array2D< T > &A, const Array2D< T > &B)
- template<class T >
Array2D< T > & [TNT::operator/=](#) (Array2D< T > &A, const Array2D< T > &B)
- template<class T >
Array2D< T > [TNT::matmult](#) (const Array2D< T > &A, const Array2D< T > &B)

11.45 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array3d.h File Reference

```
#include <cstdlib>
#include <iostream>
#include "tnt_array1d.h"
#include "tnt_array2d.h"
```

- class [TNT::Array3D< T >](#)

Namespaces

- [TNT](#)

11.46 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_array3d_utils.h File Reference

```
#include <cstdlib>
#include <cassert>
```

Namespaces

- [TNT](#)

Functions

- template<class T >
std::ostream & [TNT::operator<<](#) (std::ostream &s, const Array3D< T > &A)
- template<class T >
std::istream & [TNT::operator>>](#) (std::istream &s, Array3D< T > &A)
- template<class T >
Array3D< T > [TNT::operator+](#) (const Array3D< T > &A, const Array3D< T > &B)
- template<class T >
Array3D< T > [TNT::operator-](#) (const Array3D< T > &A, const Array3D< T > &B)
- template<class T >
Array3D< T > [TNT::operator*](#) (const Array3D< T > &A, const Array3D< T > &B)
- template<class T >
Array3D< T > [TNT::operator/](#) (const Array3D< T > &A, const Array3D< T > &B)
- template<class T >
Array3D< T > & [TNT::operator+=](#) (Array3D< T > &A, const Array3D< T > &B)
- template<class T >
Array3D< T > & [TNT::operator-=](#) (Array3D< T > &A, const Array3D< T > &B)
- template<class T >
Array3D< T > & [TNT::operator*=](#) (Array3D< T > &A, const Array3D< T > &B)
- template<class T >
Array3D< T > & [TNT::operator/=](#) (Array3D< T > &A, const Array3D< T > &B)

11.47 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_cmat.h File Reference

```
#include "tnt_subscript.h"
#include "tnt_vec.h"
#include <cstdlib>
#include <cassert>
#include <iostream>
#include <sstream>
```

Classes

- class [TNT::Matrix< T >](#)

Namespaces

- [TNT](#)

Functions

- template<class T >
std::ostream & [TNT::operator<<](#) (std::ostream &s, const Matrix< T > &A)
- template<class T >
std::istream & [TNT::operator>>](#) (std::istream &s, Matrix< T > &A)
- template<class T >
Matrix< T > [TNT::operator+](#) (const Matrix< T > &A, const Matrix< T > &B)
- template<class T >
Matrix< T > [TNT::operator-](#) (const Matrix< T > &A, const Matrix< T > &B)
- template<class T >
Matrix< T > [TNT::mult_element](#) (const Matrix< T > &A, const Matrix< T > &B)
- template<class T >
Matrix< T > [TNT::transpose](#) (const Matrix< T > &A)
- template<class T >
Matrix< T > [TNT::matmult](#) (const Matrix< T > &A, const Matrix< T > &B)
- template<class T >
Matrix< T > [TNT::operator*](#) (const Matrix< T > &A, const Matrix< T > &B)
- template<class T >
int [TNT::matmult](#) (Matrix< T > &C, const Matrix< T > &A, const Matrix< T > &B)
- template<class T >
Vector< T > [TNT::matmult](#) (const Matrix< T > &A, const Vector< T > &x)
- template<class T >
Vector< T > [TNT::operator*](#) (const Matrix< T > &A, const Vector< T > &x)

11.48 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array1d.h File Reference

```
#include <cstdlib>
#include <iostream>
#include "tnt_i_refvec.h"
```

Classes

- class [TNT::Fortran_Array1D< T >](#)

Namespaces

- [TNT](#)

11.49 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array1d_utils.h File Reference

```
#include <iostream>
```

Namespaces

- [TNT](#)

Functions

- `template<class T >`
`std::ostream & TNT::operator<< (std::ostream &s, const Fortran_Array1D< T > &A)`
- `template<class T >`
`std::istream & TNT::operator>> (std::istream &s, Fortran_Array1D< T > &A)`
- `template<class T >`
`Fortran_Array1D< T > TNT::operator+ (const Fortran_Array1D< T > &A, const Fortran_Array1D< T > &B)`
- `template<class T >`
`Fortran_Array1D< T > TNT::operator- (const Fortran_Array1D< T > &A, const Fortran_Array1D< T > &B)`
- `template<class T >`
`Fortran_Array1D< T > TNT::operator* (const Fortran_Array1D< T > &A, const Fortran_Array1D< T > &B)`
- `template<class T >`
`Fortran_Array1D< T > TNT::operator/ (const Fortran_Array1D< T > &A, const Fortran_Array1D< T > &B)`
- `template<class T >`
`Fortran_Array1D< T > & TNT::operator+= (Fortran_Array1D< T > &A, const Fortran_Array1D< T > &B)`
- `template<class T >`
`Fortran_Array1D< T > & TNT::operator-= (Fortran_Array1D< T > &A, const Fortran_Array1D< T > &B)`
- `template<class T >`
`Fortran_Array1D< T > & TNT::operator*= (Fortran_Array1D< T > &A, const Fortran_Array1D< T > &B)`
- `template<class T >`
`Fortran_Array1D< T > & TNT::operator/= (Fortran_Array1D< T > &A, const Fortran_Array1D< T > &B)`

11.50 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array2d.h File Reference

```
#include <cstdlib>
#include <iostream>
#include "tnt_i_refvec.h"
```

Classes

- class [TNT::Fortran_Array2D< T >](#)

Namespaces

- [TNT](#)

11.51 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array2d_utils.h File Reference

```
#include <iostream>
```

Namespaces

- [TNT](#)

Functions

- `template<class T >`
`std::ostream & TNT::operator<< (std::ostream &s, const Fortran_Array2D< T > &A)`
- `template<class T >`
`std::istream & TNT::operator>> (std::istream &s, Fortran_Array2D< T > &A)`
- `template<class T >`
`Fortran_Array2D< T > TNT::operator+ (const Fortran_Array2D< T > &A, const Fortran_Array2D< T > &B)`
- `template<class T >`
`Fortran_Array2D< T > TNT::operator- (const Fortran_Array2D< T > &A, const Fortran_Array2D< T > &B)`
- `template<class T >`
`Fortran_Array2D< T > TNT::operator* (const Fortran_Array2D< T > &A, const Fortran_Array2D< T > &B)`
- `template<class T >`
`Fortran_Array2D< T > TNT::operator/ (const Fortran_Array2D< T > &A, const Fortran_Array2D< T > &B)`
- `template<class T >`
`Fortran_Array2D< T > & TNT::operator+= (Fortran_Array2D< T > &A, const Fortran_Array2D< T > &B)`
- `template<class T >`
`Fortran_Array2D< T > & TNT::operator-= (Fortran_Array2D< T > &A, const Fortran_Array2D< T > &B)`
- `template<class T >`
`Fortran_Array2D< T > & TNT::operator*= (Fortran_Array2D< T > &A, const Fortran_Array2D< T > &B)`
- `template<class T >`
`Fortran_Array2D< T > & TNT::operator/= (Fortran_Array2D< T > &A, const Fortran_Array2D< T > &B)`

11.52 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array3d.h File Reference

```
#include <cstdlib>
#include <iostream>
#include "tnt_i_refvec.h"
```

Classes

- class [TNT::Fortran_Array3D< T >](#)

Namespaces

- [TNT](#)

11.53 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_fortran_array3d_utils.h File Reference

```
#include <cstdlib>
#include <cassert>
```

Namespaces

- [TNT](#)

Functions

- `template<class T >`
`std::ostream & TNT::operator<< (std::ostream &s, const Fortran_Array3D< T > &A)`
- `template<class T >`
`std::istream & TNT::operator>> (std::istream &s, Fortran_Array3D< T > &A)`
- `template<class T >`
`Fortran_Array3D< T > TNT::operator+ (const Fortran_Array3D< T > &A, const Fortran_Array3D< T > &B)`
- `template<class T >`
`Fortran_Array3D< T > TNT::operator- (const Fortran_Array3D< T > &A, const Fortran_Array3D< T > &B)`
- `template<class T >`
`Fortran_Array3D< T > TNT::operator* (const Fortran_Array3D< T > &A, const Fortran_Array3D< T > &B)`
- `template<class T >`
`Fortran_Array3D< T > TNT::operator/ (const Fortran_Array3D< T > &A, const Fortran_Array3D< T > &B)`
- `template<class T >`
`Fortran_Array3D< T > & TNT::operator+= (Fortran_Array3D< T > &A, const Fortran_Array3D< T > &B)`
- `template<class T >`
`Fortran_Array3D< T > & TNT::operator-= (Fortran_Array3D< T > &A, const Fortran_Array3D< T > &B)`
- `template<class T >`
`Fortran_Array3D< T > & TNT::operator*= (Fortran_Array3D< T > &A, const Fortran_Array3D< T > &B)`
- `template<class T >`
`Fortran_Array3D< T > & TNT::operator/= (Fortran_Array3D< T > &A, const Fortran_Array3D< T > &B)`

11.54 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_i_refvec.h File Reference

```
#include <cstdlib>
#include <iostream>
```

Classes

- class [TNT::i_refvec< T >](#)

Namespaces

- [TNT](#)

Macros

- `#define` [NULL](#) 0

11.54.1 Macro Definition Documentation

11.54.1.1 `#define` [NULL](#) 0

11.55 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_math_utils.h](#) File Reference

```
#include <cmath>
```

Namespaces

- [TNT](#)

Functions

- `template<class Real >`
`Real` [TNT::hypot](#) (`const Real &a`, `const Real &b`)

11.56 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_sparse_matrix_csr.h](#) File Reference

```
#include "tnt_array1d.h"
```

Classes

- `class` [TNT::Sparse_Matrix_CompRow](#)< `T` >

Namespaces

- [TNT](#)

11.57 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_stopwatch.h](#) File Reference

```
#include <time.h>
```

Classes

- `class` [TNT::Stopwatch](#)

Namespaces

- [TNT](#)

11.58 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_subscript.h File Reference

Namespaces

- [TNT](#)

Macros

- `#define TNT_SUBSCRIPT_TYPE int`
- `#define TNT_BASE_OFFSET (1)`

Typedefs

- `typedef TNT_SUBSCRIPT_TYPE TNT::Subscript`

11.58.1 Macro Definition Documentation

11.58.1.1 `#define TNT_BASE_OFFSET (1)`

11.58.1.2 `#define TNT_SUBSCRIPT_TYPE int`

11.59 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_vec.h File Reference

```
#include "tnt_subscript.h"
#include <cstdlib>
#include <cassert>
#include <iostream>
#include <sstream>
```

Classes

- class [TNT::Vector< T >](#)

Namespaces

- [TNT](#)

Functions

- `template<class T >`
`std::ostream & TNT::operator<< (std::ostream &s, const Vector< T > &A)`
- `template<class T >`
`std::istream & TNT::operator>> (std::istream &s, Vector< T > &A)`

- `template<class T >`
`Vector< T > TNT::operator+ (const Vector< T > &A, const Vector< T > &B)`
- `template<class T >`
`Vector< T > TNT::operator- (const Vector< T > &A, const Vector< T > &B)`
- `template<class T >`
`Vector< T > TNT::operator* (const Vector< T > &A, const Vector< T > &B)`
- `template<class T >`
`T TNT::dot_prod (const Vector< T > &A, const Vector< T > &B)`

11.60 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/jama125/tnt_version.h File Reference

Macros

- `#define TNT_MAJOR_VERSION '1'`
- `#define TNT_MINOR_VERSION '2'`
- `#define TNT_SUBMINOR_VERSION '6'`
- `#define TNT_VERSION_STRING "1.2.6"`

11.60.1 Macro Definition Documentation

11.60.1.1 `#define TNT_MAJOR_VERSION '1'`

11.60.1.2 `#define TNT_MINOR_VERSION '2'`

11.60.1.3 `#define TNT_SUBMINOR_VERSION '6'`

11.60.1.4 `#define TNT_VERSION_STRING "1.2.6"`

11.61 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/linux/hid.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <locale.h>
#include <errno.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <sys/utsname.h>
#include <fcntl.h>
#include <poll.h>
#include <linux/hidraw.h>
#include <linux/version.h>
#include <linux/input.h>
#include <libudev.h>
#include "hidapi.h"
```

- struct [hid_device_](#)

Macros

- #define [HIDIOCSFEATURE](#)(len) _IOC(_IOC_WRITE|_IOC_READ, 'H', 0x06, len)
- #define [HIDIOCGFEATURE](#)(len) _IOC(_IOC_WRITE|_IOC_READ, 'H', 0x07, len)

Enumerations

- enum [device_string_id](#) { [DEVICE_STRING_MANUFACTURER](#), [DEVICE_STRING_PRODUCT](#), [DEVICE_STRING_SERIAL](#), [DEVICE_STRING_COUNT](#) }

Functions

- [hid_device](#) * [new_hid_device](#) ()
- int [parse_uevent_info](#) (const char *uevent, int *bus_type, unsigned short *vendor_id, unsigned short *product_id, char **serial_number_utf8, char **product_name_utf8)
- int [HID_API_EXPORT](#) [hid_init](#) (void)
Initialize the HIDAPI library.
- int [HID_API_EXPORT](#) [hid_exit](#) (void)
Finalize the HIDAPI library.
- struct [hid_device_info](#)
[HID_API_EXPORT](#) * [hid_enumerate](#) (unsigned short vendor_id, unsigned short product_id)
Enumerate the HID Devices.
- void [HID_API_EXPORT](#) [hid_free_enumeration](#) (struct [hid_device_info](#) *devs)
Free an enumeration Linked List.
- [hid_device](#) * [hid_open](#) (unsigned short vendor_id, unsigned short product_id, const wchar_t *serial_number)
Open a HID device using a Vendor ID (VID), Product ID (PID) and optionally a serial number.
- [hid_device](#) * [HID_API_EXPORT](#) [hid_open_path](#) (const char *path)
Open a HID device by its path name.
- int [HID_API_EXPORT](#) [hid_write](#) ([hid_device](#) *dev, const unsigned char *data, size_t length)
Write an Output report to a HID device.
- int [HID_API_EXPORT](#) [hid_read_timeout](#) ([hid_device](#) *dev, unsigned char *data, size_t length, int milliseconds)
Read an Input report from a HID device with timeout.
- int [HID_API_EXPORT](#) [hid_read](#) ([hid_device](#) *dev, unsigned char *data, size_t length)
Read an Input report from a HID device.
- int [HID_API_EXPORT](#) [hid_set_nonblocking](#) ([hid_device](#) *dev, int nonblock)
Set the device handle to be non-blocking.
- int [HID_API_EXPORT](#) [hid_send_feature_report](#) ([hid_device](#) *dev, const unsigned char *data, size_t length)
Send a Feature report to the device.
- int [HID_API_EXPORT](#) [hid_get_feature_report](#) ([hid_device](#) *dev, unsigned char *data, size_t length)
Get a feature report from a HID device.
- void [HID_API_EXPORT](#) [hid_close](#) ([hid_device](#) *dev)
Close a HID device.
- int [HID_API_EXPORT_CALL](#) [hid_get_manufacturer_string](#) ([hid_device](#) *dev, wchar_t *string, size_t maxlen)
Get The Manufacturer String from a HID device.
- int [HID_API_EXPORT_CALL](#) [hid_get_product_string](#) ([hid_device](#) *dev, wchar_t *string, size_t maxlen)
Get The Product String from a HID device.

- int `HID_API_EXPORT_CALL hid_get_serial_number_string` (`hid_device` *dev, `wchar_t` *string, `size_t` maxlen)
Get The Serial Number String from a HID device.
- int `HID_API_EXPORT_CALL hid_get_indexed_string` (`hid_device` *dev, int string_index, `wchar_t` *string, `size_t` maxlen)
Get a string from a HID device, based on its string index.
- `HID_API_EXPORT` const `wchar_t`
*`HID_API_CALL hid_error` (`hid_device` *dev)
Get a string describing the last error which occurred.

Variables

- const char * `device_string_names` []

11.61.1 Macro Definition Documentation

11.61.1.1 `#define HIDIOCGFEATURE(len) _IOC(_IOC_WRITE|_IOC_READ, 'H', 0x07, len)`

11.61.1.2 `#define HIDIOCSFEATURE(len) _IOC(_IOC_WRITE|_IOC_READ, 'H', 0x06, len)`

11.61.2 Enumeration Type Documentation

11.61.2.1 enum `device_string_id`

Enumerator

DEVICE_STRING_MANUFACTURER

DEVICE_STRING_PRODUCT

DEVICE_STRING_SERIAL

DEVICE_STRING_COUNT

11.61.3 Function Documentation

11.61.3.1 `hid_device*` `new_hid_device` (void)

11.61.3.2 int `parse_uevent_info` (const char * *uevent*, int * *bus_type*, unsigned short * *vendor_id*, unsigned short * *product_id*, char ** *serial_number_utf8*, char ** *product_name_utf8*)

type vendor product

HID_ID=0003:000005AC:00008242

11.61.4 Variable Documentation

11.61.4.1 const char* `device_string_names`[]

Initial value:

```
= {
    "manufacturer",
    "product",
    "serial",
}
```

11.62 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammer/SBS3/smartphonebrainscanner2-core/src/platform/osx/hid.c File Reference

```
#include <IOKit/hid/IOHIDManager.h>
#include <IOKit/hid/IOHIDKeys.h>
#include <CoreFoundation/CoreFoundation.h>
#include <wchar.h>
#include <locale.h>
#include <pthread.h>
#include <sys/time.h>
#include <unistd.h>
#include "hidapi.h"
```

Classes

- struct [pthread_barrier](#)
- struct [input_report](#)
- struct [hid_device_](#)

Macros

- #define [BUF_LEN](#) 256

Typedefs

- typedef int [pthread_barrierattr_t](#)
- typedef struct [pthread_barrier](#) [pthread_barrier_t](#)

Functions

- int [HID_API_EXPORT hid_init](#) (void)
Initialize the HIDAPI library.
- int [HID_API_EXPORT hid_exit](#) (void)
Finalize the HIDAPI library.
- struct [hid_device_info](#)
[HID_API_EXPORT * hid_enumerate](#) (unsigned short vendor_id, unsigned short product_id)
Enumerate the HID Devices.
- void [HID_API_EXPORT hid_free_enumeration](#) (struct [hid_device_info](#) *devs)
Free an enumeration Linked List.
- [hid_device](#) *[HID_API_EXPORT hid_open](#) (unsigned short vendor_id, unsigned short product_id, wchar_t *serial_number)
Open a HID device using a Vendor ID (VID), Product ID (PID) and optionally a serial number.
- [hid_device](#) *[HID_API_EXPORT hid_open_path](#) (const char *path)
Open a HID device by its path name.
- int [HID_API_EXPORT hid_write](#) ([hid_device](#) *dev, const unsigned char *data, size_t length)
Write an Output report to a HID device.
- int [HID_API_EXPORT hid_read_timeout](#) ([hid_device](#) *dev, unsigned char *data, size_t length, int milliseconds)
Read an Input report from a HID device with timeout.
- int [HID_API_EXPORT hid_read](#) ([hid_device](#) *dev, unsigned char *data, size_t length)

- Read an Input report from a HID device.*

 - int [HID_API_EXPORT hid_set_nonblocking](#) ([hid_device](#) *dev, int nonblock)

Set the device handle to be non-blocking.
- int [HID_API_EXPORT hid_send_feature_report](#) ([hid_device](#) *dev, const unsigned char *data, size_t length)

Send a Feature report to the device.
- int [HID_API_EXPORT hid_get_feature_report](#) ([hid_device](#) *dev, unsigned char *data, size_t length)

Get a feature report from a HID device.
- void [HID_API_EXPORT hid_close](#) ([hid_device](#) *dev)

Close a HID device.
- int [HID_API_EXPORT_CALL hid_get_manufacturer_string](#) ([hid_device](#) *dev, wchar_t *string, size_t maxlen)

Get The Manufacturer String from a HID device.
- int [HID_API_EXPORT_CALL hid_get_product_string](#) ([hid_device](#) *dev, wchar_t *string, size_t maxlen)

Get The Product String from a HID device.
- int [HID_API_EXPORT_CALL hid_get_serial_number_string](#) ([hid_device](#) *dev, wchar_t *string, size_t maxlen)

Get The Serial Number String from a HID device.
- int [HID_API_EXPORT_CALL hid_get_indexed_string](#) ([hid_device](#) *dev, int string_index, wchar_t *string, size_t maxlen)

Get a string from a HID device, based on its string index.
- [HID_API_EXPORT](#) const wchar_t *[HID_API_CALL hid_error](#) ([hid_device](#) *dev)

Get a string describing the last error which occurred.

11.62.1 Macro Definition Documentation

11.62.1.1 `#define BUF_LEN 256`

11.62.2 Typedef Documentation

11.62.2.1 `typedef struct pthread_barrier pthread_barrier_t`

11.62.2.2 `typedef int pthread_barrierattr_t`

11.63 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/linux/hidapi.h File Reference

```
#include <wchar.h>
```

Classes

- struct [hid_device_info](#)

Macros

- `#define HID_API_EXPORT`
- `#define HID_API_CALL`
- `#define HID_API_EXPORT_CALL HID_API_EXPORT HID_API_CALL`

- typedef struct [hid_device](#) [hid_device](#)

Functions

- int [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_init](#) (void)
Initialize the HIDAPI library.
- int [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_exit](#) (void)
Finalize the HIDAPI library.
- struct [hid_device_info](#)
[HID_API_EXPORT](#) *[HID_API_CALL](#) [hid_enumerate](#) (unsigned short vendor_id, unsigned short product_id)
Enumerate the HID Devices.
- void [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_free_enumeration](#) (struct [hid_device_info](#) *devs)
Free an enumeration Linked List.
- [HID_API_EXPORT](#) [hid_device](#)
*[HID_API_CALL](#) [hid_open](#) (unsigned short vendor_id, unsigned short product_id, const wchar_t *serial_number)
Open a HID device using a Vendor ID (VID), Product ID (PID) and optionally a serial number.
- [HID_API_EXPORT](#) [hid_device](#)
*[HID_API_CALL](#) [hid_open_path](#) (const char *path)
Open a HID device by its path name.
- int [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_write](#) ([hid_device](#) *device, const unsigned char *data, size_t length)
Write an Output report to a HID device.
- int [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_read_timeout](#) ([hid_device](#) *dev, unsigned char *data, size_t length, int milliseconds)
Read an Input report from a HID device with timeout.
- int [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_read](#) ([hid_device](#) *device, unsigned char *data, size_t length)
Read an Input report from a HID device.
- int [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_set_nonblocking](#) ([hid_device](#) *device, int nonblock)
Set the device handle to be non-blocking.
- int [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_send_feature_report](#) ([hid_device](#) *device, const unsigned char *data, size_t length)
Send a Feature report to the device.
- int [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_get_feature_report](#) ([hid_device](#) *device, unsigned char *data, size_t length)
Get a feature report from a HID device.
- void [HID_API_EXPORT](#) [HID_API_CALL](#) [hid_close](#) ([hid_device](#) *device)
Close a HID device.
- int [HID_API_EXPORT_CALL](#) [hid_get_manufacturer_string](#) ([hid_device](#) *device, wchar_t *string, size_t maxlen)
Get The Manufacturer String from a HID device.
- int [HID_API_EXPORT_CALL](#) [hid_get_product_string](#) ([hid_device](#) *device, wchar_t *string, size_t maxlen)
Get The Product String from a HID device.
- int [HID_API_EXPORT_CALL](#) [hid_get_serial_number_string](#) ([hid_device](#) *device, wchar_t *string, size_t maxlen)
Get The Serial Number String from a HID device.
- int [HID_API_EXPORT_CALL](#) [hid_get_indexed_string](#) ([hid_device](#) *device, int string_index, wchar_t *string, size_t maxlen)
Get a string from a HID device, based on its string index.
- [HID_API_EXPORT](#) const wchar_t
*[HID_API_CALL](#) [hid_error](#) ([hid_device](#) *device)
Get a string describing the last error which occurred.

11.63.1 Macro Definition Documentation

11.63.1.1 `#define HID_API_CALL`

API call macro

11.63.1.2 `#define HID_API_EXPORT`

API export macro

11.63.1.3 `#define HID_API_EXPORT_CALL HID_API_EXPORT HID_API_CALL`

API export and call macro

11.63.2 Typedef Documentation

11.63.2.1 `typedef struct hid_device_ hid_device`

opaque hidapi structure

11.64 `/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/platform/osx/hidapi.h` File Reference

```
#include <wchar.h>
```

Classes

- struct [hid_device_info](#)

Macros

- `#define HID_API_EXPORT`
- `#define HID_API_CALL`
- `#define HID_API_EXPORT_CALL HID_API_EXPORT HID_API_CALL`

Typedefs

- `typedef struct hid_device_ hid_device`

Functions

- int [HID_API_EXPORT HID_API_CALL hid_init](#) (void)
Initialize the HIDAPI library.
- int [HID_API_EXPORT HID_API_CALL hid_exit](#) (void)
Finalize the HIDAPI library.
- struct [hid_device_info](#)
[HID_API_EXPORT *HID_API_CALL hid_enumerate](#) (unsigned short vendor_id, unsigned short product_id)
Enumerate the HID Devices.

-
- void [HID_API_EXPORT HID_API_CALL hid_free_enumeration](#) (struct [hid_device_info](#) *devs)
Free an enumeration Linked List.
 - [HID_API_EXPORT hid_device](#)
[*HID_API_CALL hid_open](#) (unsigned short vendor_id, unsigned short product_id, wchar_t *serial_number)
Open a HID device using a Vendor ID (VID), Product ID (PID) and optionally a serial number.
 - [HID_API_EXPORT hid_device](#)
[*HID_API_CALL hid_open_path](#) (const char *path)
Open a HID device by its path name.
 - int [HID_API_EXPORT HID_API_CALL hid_write](#) ([hid_device](#) *device, const unsigned char *data, size_t length)
Write an Output report to a HID device.
 - int [HID_API_EXPORT HID_API_CALL hid_read_timeout](#) ([hid_device](#) *dev, unsigned char *data, size_t length, int milliseconds)
Read an Input report from a HID device with timeout.
 - int [HID_API_EXPORT HID_API_CALL hid_read](#) ([hid_device](#) *device, unsigned char *data, size_t length)
Read an Input report from a HID device.
 - int [HID_API_EXPORT HID_API_CALL hid_set_nonblocking](#) ([hid_device](#) *device, int nonblock)
Set the device handle to be non-blocking.
 - int [HID_API_EXPORT HID_API_CALL hid_send_feature_report](#) ([hid_device](#) *device, const unsigned char *data, size_t length)
Send a Feature report to the device.
 - int [HID_API_EXPORT HID_API_CALL hid_get_feature_report](#) ([hid_device](#) *device, unsigned char *data, size_t length)
Get a feature report from a HID device.
 - void [HID_API_EXPORT HID_API_CALL hid_close](#) ([hid_device](#) *device)
Close a HID device.
 - int [HID_API_EXPORT_CALL hid_get_manufacturer_string](#) ([hid_device](#) *device, wchar_t *string, size_t maxlen)
Get The Manufacturer String from a HID device.
 - int [HID_API_EXPORT_CALL hid_get_product_string](#) ([hid_device](#) *device, wchar_t *string, size_t maxlen)
Get The Product String from a HID device.
 - int [HID_API_EXPORT_CALL hid_get_serial_number_string](#) ([hid_device](#) *device, wchar_t *string, size_t maxlen)
Get The Serial Number String from a HID device.
 - int [HID_API_EXPORT_CALL hid_get_indexed_string](#) ([hid_device](#) *device, int string_index, wchar_t *string, size_t maxlen)
Get a string from a HID device, based on its string index.
 - [HID_API_EXPORT](#) const wchar_t
[*HID_API_CALL hid_error](#) ([hid_device](#) *device)
Get a string describing the last error which occurred.

11.64.1 Macro Definition Documentation

11.64.1.1 #define HID_API_CALL

API call macro

11.64.1.2 #define HID_API_EXPORT

API export macro

11.64.1.3 `#define HID_API_EXPORT_CALL HID_API_EXPORT HID_API_CALL`

API export and call macro

11.64.2 Typedef Documentation

11.64.2.1 `typedef struct hid_device_ hid_device`

opaque hidapi structure

11.65 `/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/qmlapplicationviewer/qmlapplicationviewer.cpp` File Reference

```
#include "qmlapplicationviewer.h"
#include <QtCore/QDir>
#include <QtCore/QFileInfo>
#include <QtDeclarative/QDeclarativeComponent>
#include <QtDeclarative/QDeclarativeEngine>
#include <QtDeclarative/QDeclarativeContext>
#include <QtGui/QApplication>
#include <qplatformdefs.h>
```

Classes

- class [QmlApplicationViewerPrivate](#)

Functions

- `QApplication *` [createApplication](#) (`int &argc`, `char **argv`)

11.65.1 Function Documentation

11.65.1.1 `QApplication* createApplication (int & argc, char ** argv)`

11.66 `/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/qmlapplicationviewer/qmlapplicationviewer.h` File Reference

```
#include <QtDeclarative/QDeclarativeView>
```

Classes

- class [QmlApplicationViewer](#)

Functions

- `QApplication *` [createApplication](#) (`int &argc`, `char **argv`)

11.66.1 Function Documentation

11.66.1.1 `QApplication* createApplication (int & argc, char ** argv)`

11.67 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/README.md File Reference

11.68 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2callback.cpp File Reference

```
#include <sbs2callback.h>
#include <sbs2datahandler.h>
```

11.69 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2callback.h File Reference

```
#include <QObject>
#include <QNetworkInterface>
#include <QVariant>
#include <hardware/sbs2packet.h>
#include <sbs2spectrogram.h>
#include <sbs2region.h>
```

Classes

- class [Sbs2Callback](#)

11.70 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2common.cpp File Reference

```
#include <sbs2common.h>
```

11.71 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2common.h File Reference

```
#include <QMap>
#include <QVector>
#include <QString>
#include <QDebug>
#include <qplatformdefs.h>
#include <QDir>
```

Classes

- class [Sbs2Common](#)

Macros

- #define [DEPLOYMENT](#) 0

11.71.1 Macro Definition Documentation

11.71.1.1 #define DEPLOYMENT 0

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU](#) Informatics, Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Todo Loading hardware configuration from a file.

11.72 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.cpp File Reference

```
#include "sbs2datahandler.h"
```

11.73 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.h File Reference

```
#include <QObject>
#include <sbs2networkhandler.h>
#include <sbs2filter.h>
#include <hardware/sbs2packet.h>
#include <sbs2common.h>
#include <sbs2filehandler.h>
#include <sbs2spectrogram.h>
#include <source_reconstruction/sbs2sourcereconstruction.h>
#include <QtCore>
```

- class [Sbs2DataHandler](#)

11.74 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filehandler.cpp File Reference

```
#include "sbs2filehandler.h"
```

11.75 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filehandler.h File Reference

```
#include <QObject>
#include <sbs2common.h>
#include <fstream>
#include <QFile>
#include <stdlib.h>
#include <QDateTime>
#include <hardware/sbs2hardwaremounter.h>
#include <sbs2callback.h>
```

Classes

- class [Sbs2FileHandler](#)

11.76 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filter.cpp File Reference

```
#include "sbs2filter.h"
```

11.77 /media/philipjhj/Data/OneDrive/Studie/Studentprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2filter.h File Reference

```
#include <QObject>
#include <sbs2common.h>
#include <QFile>
#include <QStringList>
#include <dtu_array_2d.h>
```

Classes

- class [Sbs2Filter](#)

11.78 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.cpp File Reference

```
#include <sbs2networkhandler.h>
```

11.79 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.h File Reference

```
#include <QString>
#include <QUdpSocket>
#include <vector>
#include <iostream>
#include <QTcpSocket>
#include <cmath>
```

Classes

- class [Sbs2NetworkHandler](#)

Macros

- #define [MAX_BUFFER_SIZE](#) 512

11.79.1 Macro Definition Documentation

11.79.1.1 #define MAX_BUFFER_SIZE 512

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

11.80 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2region.cpp File Reference

```
#include "sbs2region.h"
```

11.81 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2region.h File Reference

```
#include <QObject>
#include <sbs2common.h>
#include <QMap>
#include <QFile>
#include <QString>
#include <QStringList>
#include <dtu_array_2d.h>
#include <QtAlgorithms>
```

Classes

- class [Sbs2Region](#)

11.82 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2spectrogram.cpp File Reference

```
#include "sbs2spectrogram.h"
```

11.83 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/sbs2spectrogram.h File Reference

```
#include <QObject>
#include <FFTReal.h>
#include <dtu_array_2d.h>
```

Classes

- class [Sbs2Spectrogram](#)

Macros

- #define [PI](#) 3.14159265

11.83.1 Macro Definition Documentation

11.83.1.1 #define PI 3.14159265

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics](#), Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

11.84 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/loreta/sbs2sourcereconstruction_loreta.cpp File Reference

```
#include "sbs2sourcereconstruction_loreta.h"
```

11.85 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/loreta/sbs2sourcereconstruction_loreta.h File Reference

```
#include <QObject>
#include <dtu_array_2d.h>
#include <sbs2common.h>
#include <QFile>
#include <QStringList>
#include <QDateTime>
#include <QtCore>
#include <utils/sbs2timer.h>
#include <complex>
#include <sbs2spectrogram.h>
```

Classes

- class [Sbs2SourceReconstrucionLoreta](#)

- #define [PI](#) 3.14159265

11.85.1 Macro Definition Documentation

11.85.1.1 #define [PI](#) 3.14159265

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU](#) Informatics, Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

11.86 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sbs2sourcereconstruction.cpp File Reference

```
#include "sbs2sourcereconstruction.h"
```

11.87 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sbs2sourcereconstruction.h File Reference

```
#include <QObject>
#include <source_reconstruction/sparse/sbs2sourcereconstruction_sparse.h>
#include <source_reconstruction/loreta/sbs2sourcereconstruction_loreta.h>
#include <sbs2common.h>
```

Classes

- class [Sbs2SourceReconstruction](#)

11.88 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/math_utilities.cpp File Reference

```
#include <source_reconstruction/sparse/math_utilities.h>
```

```
#include <cmath>
#include <iostream>
#include <cstdio>
#include <QStringList>
#include <QFile>
#include <QTextStream>
#include <QDebug>
```

Functions

- void `getMean` (`DTU::DtuArray2D< double > *matrix`, `DTU::DtuArray2D< double > *matrixMean`)
- void `loadData` (`QString pathFile`, `DTU::DtuArray2D< double > *matrix`)
- void `loadData` (`QString pathFile`, `double *scalar`)
- void `printMatrix` (`DTU::DtuArray2D< double > *matrix`)
- void `copyMatrix` (`DTU::DtuArray2D< double > *matrix_source`, `DTU::DtuArray2D< double > *matrix_destiny`)
- void `matrixMultiplicationComponentWise` (`DTU::DtuArray2D< double > *matrix_A`, `DTU::DtuArray2D< double > *matrix_B`, `DTU::DtuArray2D< double > *out`)
- void `vectorOuterProduct` (`vector< double > &vector1`, `vector< double > &vector2`, `DTU::DtuArray2D< double > *out`)
- void `thresholding_insitu` (`vector< double > &x_vector`)
- void `scalarMinusVector_insitu` (`double *scalar`, `vector< double > &x_vector`)
- void `scalarDividedbyVectorComponentWise_insitu` (`double *scalar`, `vector< double > &x_vector`)
- void `printVector` (`vector< double > &out`)
- void `matrixL21NormEachRow` (`DTU::DtuArray2D< double > *matrix`, `vector< double > &out`)
- void `matrixL21Norm` (`DTU::DtuArray2D< double > *matrix`, `double *out`)
- void `matrixFrobNorm` (`DTU::DtuArray2D< double > *matrix`, `double *out`)

11.88.1 Function Documentation

11.88.1.1 void `copyMatrix` (`DTU::DtuArray2D< double > * matrix_source`, `DTU::DtuArray2D< double > * matrix_destiny`)

11.88.1.2 void `getMean` (`DTU::DtuArray2D< double > * matrix`, `DTU::DtuArray2D< double > * matrixMean`)

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, DTU Informatics, Cognitive Systems Section. <http://code.google.com/p/smartphonebrainscanner2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-
- 11.88.1.3 void loadData (QString pathFile, DTU::DtuArray2D< double > * matrix)
 - 11.88.1.4 void loadData (QString pathFile, double * scalar)
 - 11.88.1.5 void matrixFrobNorm (DTU::DtuArray2D< double > * matrix, double * out)
 - 11.88.1.6 void matrixL21Norm (DTU::DtuArray2D< double > * matrix, double * out)
 - 11.88.1.7 void matrixL21NormEachRow (DTU::DtuArray2D< double > * matrix, vector< double > & out)
 - 11.88.1.8 void matrixMultiplicationComponentWise (DTU::DtuArray2D< double > * matrix_A, DTU::DtuArray2D< double > * matrix_B, DTU::DtuArray2D< double > * out)
 - 11.88.1.9 void printMatrix (DTU::DtuArray2D< double > * matrix)
 - 11.88.1.10 void printVector (vector< double > & out)
 - 11.88.1.11 void scalarDividedbyVectorComponentWise_insitu (double * scalar, vector< double > & x_vector)
 - 11.88.1.12 void scalarMinusVector_insitu (double * scalar, vector< double > & x_vector)
 - 11.88.1.13 void thresholding_insitu (vector< double > & x_vector)
 - 11.88.1.14 void vectorOuterProduct (vector< double > & vector1, vector< double > & vector2, DTU::DtuArray2D< double > * out)

11.89 /media/philipjhj/Data/OneDrive/Studie/Studentprogramør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/math_utilities.h File Reference

```
#include <dtu_array_2d.h>
#include <vector>
#include <QString>
```

Functions

- void [getMean](#) (DTU::DtuArray2D< double > *matrix, DTU::DtuArray2D< double > *matrixMean)
- void [loadData](#) (QString pathFile, DTU::DtuArray2D< double > *matrix)
- void [loadData](#) (QString pathFile, double *scalar)
- void [matrixL21Norm](#) (DTU::DtuArray2D< double > *matrix, double *out)
- void [matrixFrobNorm](#) (DTU::DtuArray2D< double > *matrix, double *out)
- void [matrixL21NormEachRow](#) (DTU::DtuArray2D< double > *matrix, vector< double > &out_vector)
- void [printVector](#) (vector< double > &out_vector)
- void [printMatrix](#) (DTU::DtuArray2D< double > *matrix)
- void [scalarDividedbyVectorComponentWise_insitu](#) (double *scalar, vector< double > &x_vector)
- void [scalarMinusVector_insitu](#) (double *scalar, vector< double > &x_vector)
- void [thresholding_insitu](#) (vector< double > &x_vector)
- void [vectorOuterProduct](#) (vector< double > &vector1, vector< double > &vector2, DTU::DtuArray2D< double > *out)
- void [matrixMultiplicationComponentWise](#) (DTU::DtuArray2D< double > *matrix_A, DTU::DtuArray2D< double > *matrix_B, DTU::DtuArray2D< double > *out)
- void [copyMatrix](#) (DTU::DtuArray2D< double > *matrix_source, DTU::DtuArray2D< double > *matrix_destiny)

11.89.1 Function Documentation

11.89.1.1 `void copyMatrix (DTU::DtuArray2D< double > * matrix_source, DTU::DtuArray2D< double > * matrix_destiny)`

11.89.1.2 `void getMean (DTU::DtuArray2D< double > * matrix, DTU::DtuArray2D< double > * matrixMean)`

Smartphone Brain Scanner 2 License Agreement (MIT License)

Copyright (c) 2012 Arkadiusz Stopczynski, Jakob Eg Larsen, Carsten Stahlhut, Michael Kai Petersen, Lars Kai Hansen. Technical University of Denmark, [DTU Informatics, Cognitive Systems Section. http://code.google.com/p/smartphonebrainscanner2](http://code.google.com/p/smartphonebrainscanner2)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

11.89.1.3 `void loadData (QString pathFile, DTU::DtuArray2D< double > * matrix)`

11.89.1.4 `void loadData (QString pathFile, double * scalar)`

11.89.1.5 `void matrixFrobNorm (DTU::DtuArray2D< double > * matrix, double * out)`

11.89.1.6 `void matrixL21Norm (DTU::DtuArray2D< double > * matrix, double * out)`

11.89.1.7 `void matrixL21NormEachRow (DTU::DtuArray2D< double > * matrix, vector< double > & out_vector)`

11.89.1.8 `void matrixMultiplicationComponentWise (DTU::DtuArray2D< double > * matrix_A, DTU::DtuArray2D< double > * matrix_B, DTU::DtuArray2D< double > * out)`

11.89.1.9 `void printMatrix (DTU::DtuArray2D< double > * matrix)`

11.89.1.10 `void printVector (vector< double > & out_vector)`

11.89.1.11 `void scalarDividedbyVectorComponentWise_insitu (double * scalar, vector< double > & x_vector)`

11.89.1.12 `void scalarMinusVector_insitu (double * scalar, vector< double > & x_vector)`

11.89.1.13 `void thresholding_insitu (vector< double > & x_vector)`

11.89.1.14 `void vectorOuterProduct (vector< double > & vector1, vector< double > & vector2, DTU::DtuArray2D< double > * out)`

11.90 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/sbs2sourcereconstruction_sparse.cpp](#) File

Reference

161

11.90 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/sbs2sourcereconstruction_sparse.cpp](#) File
Reference

```
#include "sbs2sourcereconstruction_sparse.h"
#include <cassert>
#include <iostream>
#include <sbs2common.h>
```

11.91 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/source_reconstruction/sparse/sbs2sourcereconstruction_sparse.h](#) File
Reference

```
#include <QObject>
#include <dtu_array_2d.h>
#include <vector>
#include <QString>
#include <source_reconstruction/sparse/math_utilities.h>
#include <QVector>
#include <sbs2spectrogram.h>
```

Classes

- class [Sbs2SourceReconstructionSparse](#)

11.92 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utls/Rijndael.cpp](#) File Reference

```
#include <cstring>
#include <exception>
#include "Rijndael.h"
```

11.93 [/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utls/Rijndael.h](#) File Reference

```
#include <exception>
#include <cstring>
```

Classes

- class [CRijndael](#)

11.94 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utils/sbs2timer.cpp File Reference

```
#include "sbs2timer.h"
```

11.95 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utils/sbs2timer.h File Reference

```
#include <QObject>  
#include <QDateTime>  
#include <QDebug>
```

Classes

- class [Sbs2Timer](#)

11.96 /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/SBS3/smartphonebrainscanner2-core/src/utils/waiter.h File Reference

```
#include <QThread>
```

Classes

- class [Waiter](#)

Index

- ~Array1D
 - TNT::Array1D, [36](#)
- ~Array2D
 - TNT::Array2D, [37](#)
- ~Array3D
 - TNT::Array3D, [39](#)
- ~CRijndael
 - CRijndael, [44](#)
- ~FFTReal
 - FFTReal, [49](#)
- ~Fortran_Array1D
 - TNT::Fortran_Array1D, [50](#)
- ~Fortran_Array2D
 - TNT::Fortran_Array2D, [52](#)
- ~Fortran_Array3D
 - TNT::Fortran_Array3D, [53](#)
- ~Matrix
 - TNT::Matrix, [62](#)
- ~QmlApplicationViewer
 - QmlApplicationViewer, [65](#)
- ~Sbs2DataHandler
 - Sbs2DataHandler, [75](#)
- ~Sbs2DataReader
 - Sbs2DataReader, [79](#)
- ~Sbs2Emocap28DataReader
 - Sbs2Emocap28DataReader, [82](#)
- ~Sbs2Emocap28Mounter
 - Sbs2Emocap28Mounter, [84](#)
- ~Sbs2EmocapDataReader
 - Sbs2EmocapDataReader, [87](#)
- ~Sbs2EmocapMounter
 - Sbs2EmocapMounter, [89](#)
- ~Sbs2EmotivDataReader
 - Sbs2EmotivDataReader, [91](#)
- ~Sbs2EmotivMounter
 - Sbs2EmotivMounter, [94](#)
- ~Sbs2FakeDataReader
 - Sbs2FakeDataReader, [96](#)
- ~Sbs2FileHandler
 - Sbs2FileHandler, [98](#)
- ~Sbs2Filter
 - Sbs2Filter, [99](#)
- ~Sbs2HardwareMounter
 - Sbs2HardwareMounter, [101](#)
- ~Sbs2Spectrogram
 - Sbs2Spectrogram, [112](#)
- ~Vector
 - TNT::Vector, [119](#)
- ~i_refvec

- TNT::i_refvec, [57](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/FF-TRReal.cpp, [123](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/FF-TRReal.h, [123](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/R-EADME.md, [151](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/documentation-_static.cpp, [123](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/dtu-_array_2d.h, [123](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs.cpp, [124](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs.h, [124](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs.cpp, [124](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs.h, [125](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs.cpp, [125](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap/sbs.h, [125](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/s.cpp, [125](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/s.h, [126](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/s.cpp, [126](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/s.h, [126](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/hardware/emocap28/s.cpp, [126](#)

[_i_refvec.h, 139](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/jama125/tnt-h, 153](#)
[_math_utils.h, 140](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/jama125/tnt-cpp, 153](#)
[_sparse_matrix_csr.h, 140](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2filter.-h, 153](#)
[_stopwatch.h, 140](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.-cpp, 154](#)
[_subscript.h, 141](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2networkhandler.-h, 154](#)
[_vec.h, 141](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2region.-cpp, 155](#)
[_version.h, 142](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2region.-c, 142](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2spectrogram.-cpp, 155](#)
[h, 146](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2spectrogram.-h, 155](#)
[c, 145](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/source-reconstruction/loreta/sbs2sourcereconstruction-h, 148](#)
[_loreta.cpp, 156](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/source-reconstruction/loreta/sbs2sourcereconstruction-loreta.h, 156](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/source-reconstruction/sbs2sourcereconstruction.-cpp, 157](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/source-reconstruction/sbs2sourcereconstruction.h, 157](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2common.-cpp, 151](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2common.-_reconstruction/sparse/math_utilities.cpp, 157](#)
[h, 151](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2common.-_reconstruction/sparse/math_utilities.h, 159](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.-cpp, 152](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/sbs2datahandler.-_sparse.cpp, 161](#)
[h, 152](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-SBS3/smartphonebrainscanner2-core/src/source-reconstruction/sparse/sbs2sourcereconstruction-sparse.h, 161](#)
[cpp, 153](#)
[/media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-](#)

- SBS3/smartphonebrainscanner2-core/src/utils/-
 - Rijndael.cpp, [161](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-
 - SBS3/smartphonebrainscanner2-core/src/utils/-
 - Rijndael.h, [161](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-
 - SBS3/smartphonebrainscanner2-core/src/utils/sbs2timer.h, [162](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-
 - SBS3/smartphonebrainscanner2-core/src/utils/sbs2timer.h, [162](#)
- /media/philipjhj/Data/OneDrive/Studie/Studenterprogrammør/-
 - SBS3/smartphonebrainscanner2-core/src/utils/waiter.h, [162](#)
- aboutToQuit
 - Sbs2DataReader, [79](#)
 - Sbs2Emocap28DataReader, [82](#)
 - Sbs2EmocapDataReader, [87](#)
 - Sbs2EmotivDataReader, [91](#)
- add
 - DTU::DtuArray2D, [45](#)
- addImportPath
 - QmlApplicationViewer, [65](#)
- addMessageUdpOutputHost
 - Sbs2Callback, [69](#)
 - Sbs2DataHandler, [75](#)
 - Sbs2NetworkHandler, [103](#)
- addRawDataHost
 - Sbs2Callback, [69](#)
 - Sbs2DataHandler, [75](#)
 - Sbs2NetworkHandler, [103](#)
- addRegion
 - Sbs2Region, [106](#)
- addRegionsIntersection
 - Sbs2Region, [106](#)
- alignedSignal
 - Sbs2Emocap28DataReader, [82](#)
- amp1FoundSignal
 - Sbs2Emocap28DataReader, [82](#)
- amp2FoundSignal
 - Sbs2Emocap28DataReader, [83](#)
- Array1D
 - TNT::Array1D, [35](#), [36](#)
- Array2D
 - TNT::Array2D, [37](#)
- Array3D
 - TNT::Array3D, [39](#)
- BUF_LEN
 - osx/hid.c, [146](#)
- barrier
 - hid_device_, [54](#)
- battery
 - Sbs2Packet, [105](#)
- batteryValue
 - Sbs2Callback, [69](#)
- begin
 - TNT::i_refvec, [57](#)
- TNT::Vector, [119](#)
- blocking
 - hid_device_, [54](#)
- bufferIndex
 - Sbs2DataReader, [80](#)
- bufferSize
 - Sbs2DataReader, [80](#)
- CBC
 - CRijndael, [43](#)
- CFB
 - CRijndael, [43](#)
- CRijndael
 - CBC, [43](#)
 - CFB, [43](#)
 - ECB, [43](#)
 - CRijndael, [43](#)
 - ~CRijndael, [44](#)
 - CRijndael, [44](#)
 - CRijndael, [44](#)
 - Decrypt, [44](#)
 - DecryptBlock, [44](#)
 - Encrypt, [44](#)
 - EncryptBlock, [44](#)
 - GetBlockSize, [44](#)
 - GetKeyLength, [44](#)
 - GetRounds, [44](#)
 - MakeKey, [44](#)
 - ResetChain, [44](#)
 - sm_chain0, [44](#)
- calculateMean
 - Sbs2SourceReconstructionSparse, [110](#)
- calculatePower
 - Sbs2SourceReconstructionSparse, [110](#)
- channelsNo
 - Sbs2Common, [72](#)
- Cholesky
 - JAMA::Cholesky, [41](#)
- Class, [42](#)
- clearMessageUdpOutputHosts
 - Sbs2Callback, [69](#)
 - Sbs2DataHandler, [75](#)
 - Sbs2NetworkHandler, [103](#)
- clearVerticesToExtract
 - Sbs2Region, [106](#)
- close
 - Sbs2FileHandler, [98](#)
- col_ind
 - TNT::Sparse_Matrix_CompRow, [115](#)
- cond
 - JAMA::SVD, [116](#)
 - pthread_barrier, [63](#)
- condition
 - hid_device_, [54](#)
- const_iterator
 - TNT::Matrix, [62](#)
 - TNT::Vector, [118](#)
- const_reference
 - TNT::Matrix, [62](#)

- TNT::Vector, 118
- copy
 - TNT::Array1D, 36
 - TNT::Array2D, 38
 - TNT::Array3D, 39
 - TNT::Fortran_Array1D, 50
 - TNT::Fortran_Array2D, 52
 - TNT::Fortran_Array3D, 53
 - TNT::Matrix, 62
 - TNT::Vector, 119
- copy_
 - TNT::i_refvec, 57
- copyMatrix
 - math_utilities.cpp, 158
 - math_utilities.h, 160
- count
 - pthread_barrier, 64
- counter
 - Sbs2Emocap28DataContainer, 81
 - Sbs2Packet, 105
- cq
 - Sbs2Packet, 105
- cqIndex
 - Sbs2Packet, 105
- cqName
 - Sbs2Packet, 105
- cqValue
 - Sbs2Callback, 69
- cqValues
 - Sbs2Callback, 69
- create
 - QmlApplicationViewer, 65
- createApplication
 - qmlapplicationviewer.cpp, 150
 - qmlapplicationviewer.h, 151
- createMetaFile
 - Sbs2FileHandler, 98
- cross_validation_k_channel
 - Sbs2SourceReconstructionSparse, 110
- currentIndex
 - Sbs2DataReader, 80
- currentPacket
 - Sbs2Callback, 71
- currentPacketCounter
 - Sbs2Callback, 71
- DEVICE_STRING_COUNT
 - linux/hid.c, 144
- DEVICE_STRING_MANUFACTURER
 - linux/hid.c, 144
- DEVICE_STRING_PRODUCT
 - linux/hid.c, 144
- DEVICE_STRING_SERIAL
 - linux/hid.c, 144
- DEPLOYMENT
 - sbs2common.h, 152
- DTU, 25
- DTU::DtuArray2D
 - add, 45
 - dim1, 45
 - dim2, 46
 - DtuArray2D, 45
 - getSVD, 46
 - multiply, 46
 - multiplyR, 46
 - operator=, 46
 - pinv, 46
 - print, 46
 - subtract, 46
 - toIdentityMatrix, 46
 - toTntArray2D, 46
 - trace, 46
 - transpose, 46
 - transpose_insitu, 46
 - value_type, 45
- DTU::DtuArray2D< T >, 44
- data
 - input_report, 58
 - Sbs2Emocap28DataContainer, 81
- data_
 - TNT::Array2D, 38
- Decrypt
 - CRijndael, 44
- decrypt
 - Sbs2EmotivDecryptor, 93
- DecryptBlock
 - CRijndael, 44
- derivative_square_loss_frobenius
 - Sbs2SourceReconstructionSparse, 111
- destroy
 - TNT::i_refvec, 57
 - TNT::Matrix, 62
 - TNT::Vector, 119
- det
 - JAMA::LU, 59
- device_handle
 - hid_device_, 54, 55
- device_string_id
 - linux/hid.c, 144
- device_string_names
 - linux/hid.c, 144
- deviceFound
 - Sbs2Callback, 69
 - Sbs2DataReader, 80
 - Sbs2Emocap28DataReader, 83
 - Sbs2EmocapDataReader, 87
 - Sbs2EmotivDataReader, 91
 - Sbs2HardwareMounter, 101
- deviceFoundSignal
 - Sbs2Callback, 69
 - Sbs2DataReader, 80
- deviceLost
 - Sbs2DataReader, 80
 - Sbs2Emocap28DataReader, 83
 - Sbs2EmocapDataReader, 87
 - Sbs2EmotivDataReader, 91
 - Sbs2HardwareMounter, 101

- devicePresent
 - Sbs2Callback, 71
- dim
 - TNT::Array1D, 36
 - TNT::Fortran_Array1D, 50
 - TNT::Matrix, 62
 - TNT::Vector, 119
- dim1
 - DTU::DtuArray2D, 45
 - TNT::Array1D, 36
 - TNT::Array2D, 38
 - TNT::Array3D, 39
 - TNT::Fortran_Array1D, 50
 - TNT::Fortran_Array2D, 52
 - TNT::Fortran_Array3D, 53
 - TNT::Sparse_Matrix_CompRow, 115
- dim2
 - DTU::DtuArray2D, 46
 - TNT::Array2D, 38
 - TNT::Array3D, 39
 - TNT::Fortran_Array2D, 52
 - TNT::Fortran_Array3D, 53
 - TNT::Sparse_Matrix_CompRow, 115
- dim3
 - TNT::Array3D, 39
 - TNT::Fortran_Array3D, 53
- disconnected
 - hid_device_, 55
- do_fft
 - FFTReal, 49
- do_ifft
 - FFTReal, 49
- doFilter
 - Sbs2Filter, 99
- doRec
 - Sbs2SourceReconstrucionLoreta, 107
 - Sbs2SourceReconstructionSparse, 111
- doRecPow
 - Sbs2SourceReconstrucionLoreta, 107
 - Sbs2SourceReconstructionSparse, 111
- doReconstruction
 - Sbs2SourceReconstruction, 109
- doReconstructionSpectrogram
 - Sbs2SourceReconstruction, 109
- doSourceReconstruction
 - Sbs2DataHandler, 75
- doSourceReconstructionSpectrogram
 - Sbs2DataHandler, 75
- doSpectrogram
 - Sbs2Spectrogram, 112
- dot_prod
 - TNT, 29
- DtuArray2D
 - DTU::DtuArray2D, 45
- dumpRawData
 - Sbs2FileHandler, 98
- ECB
 - CRijndael, 43
- Eigenvalue
 - JAMA::Eigenvalue, 47
- element_type
 - TNT::Matrix, 62
 - TNT::Vector, 118
- Encrypt
 - CRijndael, 44
- EncryptBlock
 - CRijndael, 44
- end
 - TNT::Vector, 119
- execute
 - Sbs2DataReader, 80
- f_objective_general_group_lasso
 - Sbs2SourceReconstructionSparse, 111
- FFTReal, 49
 - ~FFTReal, 49
 - do_fft, 49
 - do_ifft, 49
 - FFTReal, 49
 - FFTReal, 49
 - flt_t, 49
 - rescale, 49
- FFTReal.h
 - flt_t, 124
- fbandHigh
 - Sbs2DataHandler, 77
- fbandLow
 - Sbs2DataHandler, 77
- filter
 - Sbs2DataHandler, 75
- filterOn
 - Sbs2DataHandler, 77
- filterOrder
 - Sbs2DataHandler, 77
- filterResultValues
 - Sbs2DataHandler, 77
- filteredValues
 - Sbs2Packet, 105
- fista_method_group_lasso_v2
 - Sbs2SourceReconstructionSparse, 111
- flt_t
 - FFTReal, 49
 - FFTReal.h, 124
- Fortran_Array1D
 - TNT::Fortran_Array1D, 50
- Fortran_Array2D
 - TNT::Fortran_Array2D, 52
- Fortran_Array3D
 - TNT::Fortran_Array3D, 53
- framesRead
 - Sbs2DataReader, 80
- GetBlockSize
 - CRijndael, 44
- getCatalogPath
 - Sbs2Common, 72
- getChannelNames

- Sbs2Common, 72
- getChannels
 - Sbs2Common, 72
- getCounter
 - Sbs2Emocap28Packet, 85
- getCqs
 - Sbs2Common, 72
- getCqsMapping
 - Sbs2Common, 72
- getCurrentHardware
 - Sbs2Common, 72
- getCurrentPacket
 - Sbs2Callback, 69
- getCurrentPacketCounter
 - Sbs2Callback, 70
- getD
 - JAMA::Eigenvalue, 48
- getData
 - Sbs2Callback, 70
- getHouseholder
 - JAMA::QR, 66
- getIdentifier
 - Sbs2HardwareMounter, 101
- getImagEigenvalues
 - JAMA::Eigenvalue, 48
- GetKeyLength
 - CRijndael, 44
- getL
 - JAMA::Cholesky, 41
 - JAMA::LU, 59
- getMean
 - math_utilities.cpp, 158
 - math_utilities.h, 160
- getNetworkAddresses
 - Sbs2Callback, 70
- getPacketZero
 - Sbs2DataHandler, 75
 - Sbs2FileHandler, 98
- getPivot
 - JAMA::LU, 59
- getPowerValues
 - Sbs2DataHandler, 75
- getQ
 - JAMA::QR, 66
- getR
 - JAMA::QR, 66
- getRawFilename
 - Sbs2Callback, 70
 - Sbs2DataHandler, 75
 - Sbs2FileHandler, 98
- getRealEigenvalues
 - JAMA::Eigenvalue, 48
- getRegionsToExtract
 - Sbs2Region, 106
- getRootAppPath
 - Sbs2Common, 72
- GetRounds
 - CRijndael, 44
- getS
 - JAMA::SVD, 116
- getSVD
 - DTU::DtuArray2D, 46
- getSingularValues
 - JAMA::SVD, 117
- getSourceReconstructionMeanValues
 - Sbs2DataHandler, 75
- getSourceReconstructionSpectrogramValues
 - Sbs2DataHandler, 75
- getU
 - JAMA::LU, 59
 - JAMA::SVD, 117
- getV
 - JAMA::Eigenvalue, 48
 - JAMA::SVD, 117
- getValue
 - Sbs2Emocap28Packet, 86
- getVerticesToExtract
 - Sbs2Region, 106
- getWindowType
 - Sbs2Spectrogram, 112
- gyroX
 - Sbs2Packet, 105
- gyroY
 - Sbs2Packet, 105
- HAMMING
 - Sbs2Spectrogram, 112
- HANN
 - Sbs2Spectrogram, 112
- HID_API_CALL
 - linux/hidapi.h, 148
 - osx/hidapi.h, 149
- HID_API_EXPORT
 - linux/hidapi.h, 148
 - osx/hidapi.h, 149
- HIDIOCGFEATURE
 - linux/hid.c, 144
- HIDIOCSFEATURE
 - linux/hid.c, 144
- hardware
 - Sbs2DataHandler, 77
- hardwareChanged
 - Sbs2Callback, 70
- hid_close
 - hidapi API, 18
- hid_device
 - linux/hidapi.h, 148
 - osx/hidapi.h, 150
- hid_device_
 - barrier, 54
 - blocking, 54
 - condition, 54
 - device_handle, 54, 55
 - disconnected, 55
 - input_report_buf, 55
 - input_reports, 55
 - max_input_report_len, 55

- mutex, 55
- next, 55
- run_loop, 55
- run_loop_mode, 55
- shutdown_barrier, 55
- shutdown_thread, 55
- source, 55
- thread, 55
- uses_numbered_reports, 55
- hid_device_info, 55
 - interface_number, 56
 - manufacturer_string, 56
 - next, 56
 - path, 56
 - product_id, 56
 - product_string, 56
 - release_number, 56
 - serial_number, 56
 - usage, 56
 - usage_page, 56
 - vendor_id, 56
- hid_enumerate
 - hidapi API, 18
- hid_error
 - hidapi API, 18
- hid_exit
 - hidapi API, 19
- hid_free_enumeration
 - hidapi API, 19
- hid_get_feature_report
 - hidapi API, 19
- hid_get_indexed_string
 - hidapi API, 19
- hid_get_manufacturer_string
 - hidapi API, 20
- hid_get_product_string
 - hidapi API, 20
- hid_get_serial_number_string
 - hidapi API, 20
- hid_init
 - hidapi API, 20
- hid_open
 - hidapi API, 21
- hid_open_path
 - hidapi API, 21
- hid_read
 - hidapi API, 22
- hid_read_timeout
 - hidapi API, 22
- hid_send_feature_report
 - hidapi API, 22
- hid_set_nonblocking
 - hidapi API, 23
- hid_write
 - hidapi API, 23
- hidapi API, 17
 - hid_close, 18
 - hid_enumerate, 18
 - hid_error, 18
 - hid_exit, 19
 - hid_free_enumeration, 19
 - hid_get_feature_report, 19
 - hid_get_indexed_string, 19
 - hid_get_manufacturer_string, 20
 - hid_get_product_string, 20
 - hid_get_serial_number_string, 20
 - hid_init, 20
 - hid_open, 21
 - hid_open_path, 21
 - hid_read, 22
 - hid_read_timeout, 22
 - hid_send_feature_report, 22
 - hid_set_nonblocking, 23
 - hid_write, 23
- hypot
 - TNT, 29
- i_refvec
 - TNT::i_refvec, 57
- identifier
 - Sbs2HardwareMounter, 101
- inMappingSignal
 - Sbs2Emocap28DataReader, 83
- init
 - Sbs2HardwareMounter, 101
- initialize
 - TNT::Matrix, 62
 - TNT::Vector, 119
- inject
 - TNT::Array1D, 36
 - TNT::Array2D, 38
 - TNT::Array3D, 39
 - TNT::Fortran_Array1D, 51
 - TNT::Fortran_Array2D, 52
 - TNT::Fortran_Array3D, 53
- input_report, 58
 - data, 58
 - len, 58
 - next, 58
- input_report_buf
 - hid_device_, 55
- input_reports
 - hid_device_, 55
- insertIntoMetaFile
 - Sbs2Callback, 70
 - Sbs2DataHandler, 76
 - Sbs2FileHandler, 98
- interface_number
 - hid_device_info, 56
- invalidate
 - Sbs2Emocap28Mounter, 84
 - Sbs2EmocapMounter, 89
 - Sbs2EmotivMounter, 94
 - Sbs2HardwareMounter, 101
- is_null
 - TNT::i_refvec, 58
- is_spd

- JAMA::Cholesky, 41
- isFullRank
 - JAMA::QR, 66
- isNonsingular
 - JAMA::LU, 60
- isRecording
 - Sbs2Callback, 71
- isSourceReconstructionReady
 - Sbs2DataHandler, 77
- iterator
 - TNT::Matrix, 62
 - TNT::Vector, 118
- JAMA, 25
- JAMA::Cholesky
 - Cholesky, 41
 - getL, 41
 - is_spd, 41
 - solve, 41
- JAMA::Cholesky< Real >, 40
- JAMA::Eigenvalue
 - Eigenvalue, 47
 - getD, 48
 - getImagEigenvalues, 48
 - getRealEigenvalues, 48
 - getV, 48
- JAMA::Eigenvalue< Real >, 46
- JAMA::LU
 - det, 59
 - getL, 59
 - getPivot, 59
 - getU, 59
 - isNonsingular, 60
 - LU, 59
 - solve, 60
- JAMA::LU< Real >, 58
- JAMA::QR
 - getHouseholder, 66
 - getQ, 66
 - getR, 66
 - isFullRank, 66
 - QR, 66
 - solve, 67
- JAMA::QR< Real >, 65
- JAMA::SVD
 - cond, 116
 - getS, 116
 - getSingularValues, 117
 - getU, 117
 - getV, 117
 - norm2, 117
 - rank, 117
 - SVD, 116
- JAMA::SVD< Real >, 116
- LU
 - JAMA::LU, 59
- lastReceiveRawDataCounter
 - Sbs2DataReader, 80
- lbound
 - TNT::Matrix, 62
 - TNT::Vector, 119
- len
 - input_report, 58
- lessThan
 - sbs2emocap28datareader.cpp, 125
- linux/hid.c
 - DEVICE_STRING_COUNT, 144
 - DEVICE_STRING_MANUFACTURER, 144
 - DEVICE_STRING_PRODUCT, 144
 - DEVICE_STRING_SERIAL, 144
- linux/hid.c
 - device_string_id, 144
 - device_string_names, 144
 - HIDIOCGFEATURE, 144
 - HIDIOCSFEATURE, 144
 - new_hid_device, 144
 - parse_uevent_info, 144
- linux/hidapi.h
 - HID_API_CALL, 148
 - HID_API_EXPORT, 148
 - hid_device, 148
- loadData
 - math_utilities.cpp, 158, 159
 - math_utilities.h, 160
- loadFilter
 - Sbs2Filter, 99
- MEAN
 - Sbs2SourceReconstrucionLoreta, 107
- m_
 - TNT::Array2D, 38
 - TNT::Matrix, 63
- MAX_BUFFER_SIZE
 - sbs2networkhandler.h, 154
- MakeKey
 - CRijndael, 44
- manufacturer_string
 - hid_device_info, 56
- mappingFailed
 - Sbs2Emocap28DataReader, 83
- mappingSuccessful
 - Sbs2Emocap28DataReader, 83
- math_utilities.cpp
 - copyMatrix, 158
 - getMean, 158
 - loadData, 158, 159
 - matrixFrobNorm, 159
 - matrixL21Norm, 159
 - matrixL21NormEachRow, 159
 - matrixMultiplicationComponentWise, 159
 - printMatrix, 159
 - printVector, 159
 - scalarDividedbyVectorComponentWise_insitu, 159
 - scalarMinusVector_insitu, 159
 - thresholding_insitu, 159
 - vectorOuterProduct, 159
- math_utilities.h

- copyMatrix, 160
- getMean, 160
- loadData, 160
- matrixFrobNorm, 160
- matrixL21Norm, 160
- matrixL21NormEachRow, 160
- matrixMultiplicationComponentWise, 160
- printMatrix, 160
- printVector, 160
- scalarDividedbyVectorComponentWise_insitu, 160
- scalarMinusVector_insitu, 160
- thresholding_insitu, 160
- vectorOuterProduct, 160
- matmult
 - TNT, 29
- Matrix
 - TNT::Matrix, 62
- matrixFrobNorm
 - math_utilities.cpp, 159
 - math_utilities.h, 160
- matrixL21Norm
 - math_utilities.cpp, 159
 - math_utilities.h, 160
- matrixL21NormEachRow
 - math_utilities.cpp, 159
 - math_utilities.h, 160
- matrixMultiplicationComponentWise
 - math_utilities.cpp, 159
 - math_utilities.h, 160
- max_input_report_len
 - hid_device_, 55
- messageReceived
 - Sbs2NetworkHandler, 103
- mn_
 - TNT::Matrix, 63
- mod
 - sbs2emocap28datareader.cpp, 125
- mount
 - Sbs2HardwareMounter, 101
- mountedHardware
 - Sbs2HardwareMounter, 101
- mult_element
 - TNT, 29
- multiply
 - DTU::DtuArray2D, 46
- multiplyR
 - DTU::DtuArray2D, 46
- mutex
 - hid_device_, 55
 - pthread_barrier, 64
- mySleep
 - Sbs2HardwareMounter, 101
- n_
 - TNT::Array2D, 38
 - TNT::Matrix, 63
 - TNT::Vector, 119
- NULL
 - tnt_i_refvec.h, 140
- networkAddresses
 - Sbs2Callback, 70
- networkSendRawDataOn
 - Sbs2DataHandler, 77
- New
 - Sbs2Emocap28DataReader, 83
 - Sbs2Emocap28Mounter, 84
 - Sbs2EmocapDataReader, 87
 - Sbs2EmocapMounter, 89
 - Sbs2EmotivDataReader, 91
 - Sbs2EmotivMounter, 94
 - Sbs2FakeDataReader, 96
 - Sbs2FileHandler, 98
 - Sbs2Filter, 99
- new_hid_device
 - linux/hid.c, 144
- newsize
 - TNT::Matrix, 62
 - TNT::Vector, 119
- next
 - hid_device_, 55
 - hid_device_info, 56
 - input_report, 58
- norm2
 - JAMA::SVD, 117
- normalize
 - Sbs2Common, 72
- num_cols
 - TNT::Matrix, 62
- num_rows
 - TNT::Matrix, 62
- NumNonzeros
 - TNT::Sparse_Matrix_CompRow, 115
- operator const T *
 - TNT::Array1D, 36
- operator const T **
 - TNT::Array2D, 38
- operator const T ***
 - TNT::Array3D, 40
- operator T *
 - TNT::Array1D, 36
- operator T **
 - TNT::Array2D, 38
 - TNT::Matrix, 62
- operator T ***
 - TNT::Array3D, 40
- operator<<
 - TNT, 32
- operator>>
 - TNT, 32, 33
- operator*
 - TNT, 29, 30
- operator*=
 - TNT, 30
- operator()
 - TNT::Fortran_Array1D, 51
 - TNT::Fortran_Array2D, 52
 - TNT::Fortran_Array3D, 54

- TNT::Matrix, [62](#), [63](#)
- TNT::Vector, [119](#)
- operator+
 - TNT, [30](#)
- operator+=
 - TNT, [30](#), [31](#)
- operator-
 - TNT, [31](#)
- operator-=
 - TNT, [31](#)
- operator/
 - TNT, [31](#)
- operator/=
 - TNT, [32](#)
- operator=
 - DTU::DtuArray2D, [46](#)
 - TNT::Array1D, [36](#)
 - TNT::Array2D, [38](#)
 - TNT::Array3D, [40](#)
 - TNT::Fortran_Array1D, [51](#)
 - TNT::Fortran_Array2D, [52](#)
 - TNT::Fortran_Array3D, [54](#)
 - TNT::i_refvec, [58](#)
 - TNT::Matrix, [63](#)
 - TNT::Sparse_Matrix_CompRow, [115](#)
 - TNT::Vector, [119](#)
- osx/hid.c
 - BUF_LEN, [146](#)
 - pthread_barrier_t, [146](#)
 - pthread_barrierattr_t, [146](#)
- osx/hidapi.h
 - HID_API_CALL, [149](#)
 - HID_API_EXPORT, [149](#)
 - hid_device, [150](#)
- POWER
 - Sbs2SourceReconstrucionLoreta, [107](#)
- PI
 - sbs2sourcereconstruction_loreta.h, [157](#)
 - sbs2spectrogram.h, [156](#)
- packetsSeen
 - Sbs2DataHandler, [77](#)
- params
 - Sbs2Callback, [71](#)
- parse_uevent_info
 - linux/hid.c, [144](#)
- path
 - hid_device_info, [56](#)
- pinv
 - DTU::DtuArray2D, [46](#)
- pointer
 - TNT::Matrix, [62](#)
 - TNT::Vector, [118](#)
- powerValues
 - Sbs2DataHandler, [77](#)
- preprocessData
 - Sbs2SourceReconstructionSparse, [111](#)
- print
 - DTU::DtuArray2D, [46](#)
- printMatrix
 - math_utilities.cpp, [159](#)
 - math_utilities.h, [160](#)
- printVector
 - math_utilities.cpp, [159](#)
 - math_utilities.h, [160](#)
- product_id
 - hid_device_info, [56](#)
- product_string
 - hid_device_info, [56](#)
- proximal_operator_standard_group_lasso
 - Sbs2SourceReconstructionSparse, [111](#)
- pthread_barrier, [63](#)
 - cond, [63](#)
 - count, [64](#)
 - mutex, [64](#)
 - trip_count, [64](#)
- pthread_barrier_t
 - osx/hid.c, [146](#)
- pthread_barrierattr_t
 - osx/hid.c, [146](#)
- QR
 - JAMA::QR, [66](#)
- QmlApplicationViewer
 - ScreenOrientationAuto, [64](#)
 - ScreenOrientationLockLandscape, [64](#)
 - ScreenOrientationLockPortrait, [64](#)
- QmlApplicationViewer, [64](#)
 - ~QmlApplicationViewer, [65](#)
 - addImportPath, [65](#)
 - create, [65](#)
 - QmlApplicationViewer, [65](#)
 - QmlApplicationViewer, [65](#)
 - QmlApplicationViewerPrivate, [65](#)
 - ScreenOrientation, [64](#)
 - setMainQmlFile, [65](#)
 - setOrientation, [65](#)
 - showExpanded, [65](#)
- QmlApplicationViewerPrivate, [65](#)
 - QmlApplicationViewer, [65](#)
- qmlapplicationviewer.cpp
 - createApplication, [150](#)
- qmlapplicationviewer.h
 - createApplication, [151](#)
- RECT
 - Sbs2Spectrogram, [112](#)
- rank
 - JAMA::SVD, [117](#)
- rawData
 - Sbs2Packet, [105](#)
- rawDataReceived
 - Sbs2NetworkHandler, [103](#)
- rawDataSentSignal
 - Sbs2NetworkHandler, [103](#)
- rawDataSize
 - Sbs2Common, [72](#)
- read

- TNT::Stopwatch, 115
- readHardwareParameters
 - Sbs2HardwareMouter, 101
- readMessage
 - Sbs2Callback, 70
 - Sbs2DataHandler, 76
 - Sbs2NetworkHandler, 103
- readOnlyFromNetwork
 - Sbs2DataReader, 80
- readRawData
 - Sbs2NetworkHandler, 103
- readyForData
 - Sbs2Emocap28DataReader, 83
- readyToReconstruct
 - Sbs2DataHandler, 77
- record
 - Sbs2DataHandler, 76
- recording
 - Sbs2DataHandler, 77
- ref
 - TNT::Array1D, 36
 - TNT::Array2D, 38
 - TNT::Array3D, 40
 - TNT::Fortran_Array1D, 51
 - TNT::Fortran_Array2D, 52
 - TNT::Fortran_Array3D, 54
- ref_count
 - TNT::Array1D, 36
 - TNT::Array2D, 38
 - TNT::Array3D, 40
 - TNT::Fortran_Array1D, 51
 - TNT::Fortran_Array2D, 52
 - TNT::Fortran_Array3D, 54
 - TNT::i_refvec, 58
- ref_count_data
 - TNT::Array2D, 38
- ref_count_dim1
 - TNT::Array2D, 38
- reference
 - TNT::Matrix, 62
 - TNT::Vector, 118
- release_number
 - hid_device_info, 56
- removeMessageUdpOutputHost
 - Sbs2Callback, 70
 - Sbs2DataHandler, 76
 - Sbs2NetworkHandler, 103
- removeRawDataHost
 - Sbs2Callback, 70
 - Sbs2DataHandler, 76
 - Sbs2NetworkHandler, 103
- rescale
 - FFTReal, 49
- reset
 - Sbs2DataHandler, 76
- ResetChain
 - CRijndael, 44
- resume
 - TNT::Stopwatch, 115
- rootMeanSquareError
 - Sbs2SourceReconstructionSparse, 111
- row_
 - TNT::Matrix, 63
- row_ptr
 - TNT::Sparse_Matrix_CompRow, 115
- rowm1_
 - TNT::Matrix, 63
- run
 - Waiter, 120
- run_loop
 - hid_device_, 55
- run_loop_mode
 - hid_device_, 55
- running
 - Sbs2DataReader, 80
 - TNT::Stopwatch, 115
- SVD
 - JAMA::SVD, 116
- samplesCollected
 - Sbs2Callback, 71
 - Sbs2DataHandler, 77
- samplingRate
 - Sbs2Common, 72
- Sbs2SourceReconstrucionLoreta
 - MEAN, 107
 - POWER, 107
- Sbs2Spectrogram
 - HAMMING, 112
 - HANN, 112
 - RECT, 112
- Sbs2Callback, 67
 - addMessageUdpOutputHost, 69
 - addRawDataHost, 69
 - batteryValue, 69
 - clearMessageUdpOutputHosts, 69
 - cqValue, 69
 - cqValues, 69
 - currentPacket, 71
 - currentPacketCounter, 71
 - deviceFound, 69
 - deviceFoundSignal, 69
 - devicePresent, 71
 - getCurrentPacket, 69
 - getCurrentPacketCounter, 70
 - getData, 70
 - getNetworkAddresses, 70
 - getRawFilename, 70
 - hardwareChanged, 70
 - insertIntoMetaFile, 70
 - isRecording, 71
 - networkAddresses, 70
 - params, 71
 - readMessage, 70
 - removeMessageUdpOutputHost, 70
 - removeRawDataHost, 70
 - samplesCollected, 71

- Sbs2Callback, 69
- sbs2DataHandler, 71
- sbs2Region, 71
- Sbs2Callback, 69
- sendMessage, 70
- setHardware, 70
- setPacket, 70
- setSbs2DataHandler, 70
- setWindowType, 70
- setWindowTypeSignal, 70
- spectrogramUpdated, 70
- spectrogramUpdatedSlot, 70
- startRecording, 70
- stopRecording, 70
- thisPacket, 71
- timeTick0, 70
- timeTick10, 70
- timeTick16, 70
- timeTick2, 70
- timeTick4, 70
- timeTick8, 71
- turnChannelSpectrogramOff, 71
- turnChannelSpectrogramOn, 71
- turnFilterOff, 71
- turnFilterOn, 71
- turnOnSourceReconstructioSparse, 71
- turnOnSourceReconstructionLoreta, 71
- turnReceiveMessageOff, 71
- turnReceiveMessageOn, 71
- turnSendRawDataOff, 71
- turnSendRawDataOn, 71
- udpMessageReceived, 71
- sbs2Callback
 - Sbs2DataReader, 80
- Sbs2Common, 72
 - channelsNo, 72
 - getCatalogPath, 72
 - getChannelNames, 72
 - getChannels, 72
 - getCqs, 72
 - getCqsMapping, 72
 - getCurrentHardware, 72
 - getRootAppPath, 72
 - normalize, 72
 - rawDataSize, 72
 - samplingRate, 72
 - setCatalogPath, 72
 - setDefaultCatalogPath, 72
 - setDefaultRootAppPath, 72
 - setHardware, 72
 - setRootAppPath, 73
 - verticesNo, 73
- Sbs2DataHandler, 73
 - ~Sbs2DataHandler, 75
 - addMessageUdpOutputHost, 75
 - addRawDataHost, 75
 - clearMessageUdpOutputHosts, 75
 - doSourceReconstruction, 75
 - doSourceReconstructionSpectrogram, 75
 - fbandHigh, 77
 - fbandLow, 77
 - filter, 75
 - filterOn, 77
 - filterOrder, 77
 - filterResultValues, 77
 - getPacketZero, 75
 - getPowerValues, 75
 - getRawFilename, 75
 - getSourceReconstructionMeanValues, 75
 - getSourceReconstructionSpectrogramValues, 75
 - hardware, 77
 - insertIntoMetaFile, 76
 - isSourceReconstructionReady, 77
 - networkSendRawDataOn, 77
 - packetsSeen, 77
 - powerValues, 77
 - readMessage, 76
 - readyToReconstruct, 77
 - record, 76
 - recording, 77
 - removeMessageUdpOutputHost, 76
 - removeRawDataHost, 76
 - reset, 76
 - samplesCollected, 77
 - Sbs2DataHandler, 75
 - sbs2FileHandler, 77
 - sbs2Filter, 77
 - sbs2NetworkHandler, 77
 - sbs2SourceReconstruction, 77
 - sbs2Spectrogram, 77
 - Sbs2DataHandler, 75
 - sendMessage, 76
 - sendRawData, 76
 - setSourceReconstructionVerticesToExtract, 76
 - setThisPacket, 76
 - setVerticesToExtract, 76
 - setWindowType, 76
 - setWindowTypeSignal, 76
 - sourceReconstructionDelta, 77
 - sourceReconstructionDeltaCollected, 78
 - sourceReconstructionMethod, 78
 - sourceReconstructionModelUpdateDelta, 78
 - sourceReconstructionModelUpdateLength, 78
 - sourceReconstructionOn, 78
 - sourceReconstructionReady, 76
 - sourceReconstructionSamples, 78
 - sourceReconstructionSpectrogramReady, 76
 - sourceReconstructionSpectrogramValues, 78
 - sourceReconstructionValues, 78
 - spectrogramChannel, 76
 - spectrogramChannelDelta, 78
 - spectrogramChannelDeltaCollected, 78
 - spectrogramChannelLength, 78
 - spectrogramChannelOn, 78
 - spectrogramChannelSamples, 78
 - spectrogramUpdated, 76

- spectrogramValues, 78
- startRecording, 76
- stopRecording, 76
- thisPacket, 78
- toFilterValues, 78
- toSourceReconstructionValues, 78
- toSpectrogramValues, 78
- turnChannelSpectrogramOff, 76
- turnChannelSpectrogramOn, 76
- turnFilterOff, 76
- turnFilterOn, 76
- turnOffSourceReconstruction, 76
- turnOnSourceReconstructionLoreta, 76
- turnOnSourceReconstructionSparse, 77
- turnReceiveMessageOff, 77
- turnReceiveMessageOn, 77
- turnSendRawDataOff, 77
- turnSendRawDataOn, 77
- udpMessageReceived, 77
- sbs2DataHandler
 - Sbs2Callback, 71
- Sbs2DataReader, 78
 - ~Sbs2DataReader, 79
 - aboutToQuit, 79
 - bufferIndex, 80
 - bufferSize, 80
 - currentIndex, 80
 - deviceFound, 80
 - deviceFoundSignal, 80
 - deviceLost, 80
 - execute, 80
 - framesRead, 80
 - lastReceiveRawDataCounter, 80
 - readOnlyFromNetwork, 80
 - running, 80
 - sbs2Callback, 80
 - Sbs2DataReader, 79
 - sbs2NetworkHandler, 80
 - Sbs2DataReader, 79
 - testDummyRead, 80
 - turnReceiveUdpDataOff, 80
 - turnReceiveUdpDataOn, 80
 - udpDataReceived, 80
- Sbs2Emocap28DataContainer, 80
 - counter, 81
 - data, 81
 - Sbs2Emocap28DataContainer, 81
 - Sbs2Emocap28DataContainer, 81
 - update, 81
- Sbs2Emocap28DataReader, 81
 - ~Sbs2Emocap28DataReader, 82
 - aboutToQuit, 82
 - alignedSignal, 82
 - amp1FoundSignal, 82
 - amp2FoundSignal, 83
 - deviceFound, 83
 - deviceLost, 83
 - inMappingSignal, 83
 - mappingFailed, 83
 - mappingSuccessful, 83
 - New, 83
 - readyForData, 83
 - turnReceiveUdpDataOff, 83
 - turnReceiveUdpDataOn, 83
 - udpDataReceived, 83
- Sbs2Emocap28Mounter, 83
 - ~Sbs2Emocap28Mounter, 84
 - invalidate, 84
 - New, 84
 - start, 84
 - stop, 84
- Sbs2Emocap28Packet, 85
 - getCounter, 85
 - getValue, 86
 - Sbs2Emocap28Packet, 85
 - Sbs2Emocap28Packet, 85
 - update, 86
- Sbs2EmocapDataReader, 86
 - ~Sbs2EmocapDataReader, 87
 - aboutToQuit, 87
 - deviceFound, 87
 - deviceLost, 87
 - New, 87
 - turnReceiveUdpDataOff, 87
 - turnReceiveUdpDataOn, 87
 - udpDataReceived, 87
- Sbs2EmocapMounter, 88
 - ~Sbs2EmocapMounter, 89
 - invalidate, 89
 - New, 89
 - start, 89
 - stop, 89
- Sbs2EmocapPacket, 89
 - Sbs2EmocapPacket, 90
 - Sbs2EmocapPacket, 90
 - update, 90
- Sbs2EmotivDataReader, 90
 - ~Sbs2EmotivDataReader, 91
 - aboutToQuit, 91
 - deviceFound, 91
 - deviceLost, 91
 - New, 91
 - turnReceiveUdpDataOff, 91
 - turnReceiveUdpDataOn, 92
 - udpDataReceived, 92
- Sbs2EmotivDecryptor, 92
 - decrypt, 93
 - Sbs2EmotivDecryptor, 93
 - Sbs2EmotivDecryptor, 93
 - setSerialNumber, 93
- Sbs2EmotivMounter, 93
 - ~Sbs2EmotivMounter, 94
 - invalidate, 94
 - New, 94
 - start, 94
 - stop, 94

- Sbs2EmotivPacket, 94
 - Sbs2EmotivPacket, 95
 - Sbs2EmotivPacket, 95
 - update, 95
- Sbs2FakeDataReader, 96
 - ~Sbs2FakeDataReader, 96
 - New, 96
 - setFilename, 96
 - start, 96
 - stop, 96
- Sbs2FakePacket, 97
 - Sbs2FakePacket, 97
 - Sbs2FakePacket, 97
 - update, 97
- Sbs2FileHandler, 97
 - ~Sbs2FileHandler, 98
 - close, 98
 - createMetaFile, 98
 - dumpRawData, 98
 - getPacketZero, 98
 - getRawFilename, 98
 - insertIntoMetaFile, 98
 - New, 98
- sbs2FileHandler
 - Sbs2DataHandler, 77
- Sbs2Filter, 99
 - ~Sbs2Filter, 99
 - doFilter, 99
 - loadFilter, 99
 - New, 99
 - updateFilter, 99
- sbs2Filter
 - Sbs2DataHandler, 77
- Sbs2HardwareMounter, 100
 - ~Sbs2HardwareMounter, 101
 - deviceFound, 101
 - deviceLost, 101
 - getIdentifier, 101
 - identifier, 101
 - init, 101
 - invalidate, 101
 - mount, 101
 - mountedHardware, 101
 - mySleep, 101
 - readHardwareParameters, 101
 - Sbs2HardwareMounter, 101
 - Sbs2HardwareMounter, 101
 - start, 101
 - stop, 101
 - umount, 101
- Sbs2NetworkHandler, 102
 - addMessageUdpOutputHost, 103
 - addRawDataHost, 103
 - clearMessageUdpOutputHosts, 103
 - messageReceived, 103
 - rawDataReceived, 103
 - rawDataSentSignal, 103
 - readMessage, 103
 - readRawData, 103
 - removeMessageUdpOutputHost, 103
 - removeRawDataHost, 103
 - Sbs2NetworkHandler, 103
 - Sbs2NetworkHandler, 103
 - sendMessage, 103
 - sendRawData, 103
 - turnReceiveMessageOff, 103
 - turnReceiveMessageOn, 103
 - turnReceiveRawDataOff, 103
 - turnReceiveRawDataOn, 103
 - turnSendRawDataOff, 103
 - turnSendRawDataOn, 103
- sbs2NetworkHandler
 - Sbs2DataHandler, 77
 - Sbs2DataReader, 80
- Sbs2Packet, 104
 - battery, 105
 - counter, 105
 - cq, 105
 - cqIndex, 105
 - cqName, 105
 - filteredValues, 105
 - gyroX, 105
 - gyroY, 105
 - rawData, 105
 - Sbs2Packet, 104
 - Sbs2Packet, 104
 - update, 104
 - values, 105
- Sbs2Region, 105
 - addRegion, 106
 - addRegionsIntersection, 106
 - clearVerticesToExtract, 106
 - getRegionsToExtract, 106
 - getVerticesToExtract, 106
 - Sbs2Region, 106
 - Sbs2Region, 106
- sbs2Region
 - Sbs2Callback, 71
- Sbs2SourceReconstrucionLoreta, 106
 - doRec, 107
 - doRecPow, 107
 - Sbs2SourceReconstrucionLoreta, 107
 - Sbs2SourceReconstrucionLoreta, 107
 - setAScaling, 108
 - setMeanExtraction, 108
 - setSumType, 108
 - setVerticesToExtract, 108
 - SumType, 107
 - tempModelUpdatedReady, 108
- Sbs2SourceReconstruction, 108
 - doReconstruction, 109
 - doReconstructionSpectrogram, 109
 - Sbs2SourceReconstruction, 109
 - Sbs2SourceReconstruction, 109
 - stopReconstruction, 109
 - turnOff, 109

- turnOnLoreta, 109
- turnOnSparse, 109
- sbs2SourceReconstruction
 - Sbs2DataHandler, 77
- Sbs2SourceReconstructionSparse, 109
 - calculateMean, 110
 - calculatePower, 110
 - cross_validation_k_channel, 110
 - derivative_square_loss_frobenius, 111
 - doRec, 111
 - doRecPow, 111
 - f_objective_general_group_lasso, 111
 - fista_method_group_lasso_v2, 111
 - preprocessData, 111
 - proximal_operator_standard_group_lasso, 111
 - rootMeanSquareError, 111
 - Sbs2SourceReconstructionSparse, 110
 - Sbs2SourceReconstructionSparse, 110
- Sbs2Spectrogram, 111
 - ~Sbs2Spectrogram, 112
 - doSpectrogram, 112
 - getWindowType, 112
 - Sbs2Spectrogram, 112
 - Sbs2Spectrogram, 112
 - setWindowType, 112
 - WindowType, 112
- sbs2Spectrogram
 - Sbs2DataHandler, 77
- Sbs2Timer, 112
 - Sbs2Timer, 113
 - Sbs2Timer, 113
 - tic, 113
 - tic_time, 113
 - toc, 113
- sbs2common.h
 - DEPLOYMENT, 152
- sbs2emocap28datareader.cpp
 - lessThan, 125
 - mod, 125
- sbs2networkhandler.h
 - MAX_BUFFER_SIZE, 154
- sbs2sourcereconstruction_loreta.h
 - PI, 157
- sbs2spectrogram.h
 - PI, 156
- scalarDividedbyVectorComponentWise_insitu
 - math_utilities.cpp, 159
 - math_utilities.h, 160
- scalarMinusVector_insitu
 - math_utilities.cpp, 159
 - math_utilities.h, 160
- ScreenOrientationAuto
 - QmlApplicationViewer, 64
- ScreenOrientationLockLandscape
 - QmlApplicationViewer, 64
- ScreenOrientationLockPortrait
 - QmlApplicationViewer, 64
- ScreenOrientation
 - QmlApplicationViewer, 64
- sendMessage
 - Sbs2Callback, 70
 - Sbs2DataHandler, 76
 - Sbs2NetworkHandler, 103
- sendRawData
 - Sbs2DataHandler, 76
 - Sbs2NetworkHandler, 103
- serial_number
 - hid_device_info, 56
- set
 - TNT::Matrix, 63
 - TNT::Vector, 119
- set_
 - TNT::i_refvec, 58
- setAScaling
 - Sbs2SourceReconstrucionLoreta, 108
- setCatalogPath
 - Sbs2Common, 72
- setDefaultCatalogPath
 - Sbs2Common, 72
- setDefaultRootAppPath
 - Sbs2Common, 72
- setFilename
 - Sbs2FakeDataReader, 96
- setHardware
 - Sbs2Callback, 70
 - Sbs2Common, 72
- setMainQmlFile
 - QmlApplicationViewer, 65
- setMeanExtraction
 - Sbs2SourceReconstrucionLoreta, 108
- setOrientation
 - QmlApplicationViewer, 65
- setPacket
 - Sbs2Callback, 70
- setRootAppPath
 - Sbs2Common, 73
- setSbs2DataHandler
 - Sbs2Callback, 70
- setSerialNumber
 - Sbs2EmotivDecryptor, 93
- setSourceReconstructionVerticesToExtract
 - Sbs2DataHandler, 76
- setSumType
 - Sbs2SourceReconstrucionLoreta, 108
- setThisPacket
 - Sbs2DataHandler, 76
- setVerticesToExtract
 - Sbs2DataHandler, 76
 - Sbs2SourceReconstrucionLoreta, 108
- setWindowType
 - Sbs2Callback, 70
 - Sbs2DataHandler, 76
 - Sbs2Spectrogram, 112
- setWindowTypeSignal
 - Sbs2Callback, 70
 - Sbs2DataHandler, 76

- showExpanded
 - QmlApplicationViewer, [65](#)
- shutdown_barrier
 - hid_device_, [55](#)
- shutdown_thread
 - hid_device_, [55](#)
- size
 - TNT::Matrix, [63](#)
 - TNT::Vector, [119](#)
- size_type
 - TNT::Matrix, [62](#)
 - TNT::Vector, [118](#)
- sm_chain0
 - CRijndael, [44](#)
- solve
 - JAMA::Cholesky, [41](#)
 - JAMA::LU, [60](#)
 - JAMA::QR, [67](#)
- source
 - hid_device_, [55](#)
- sourceReconstructionDelta
 - Sbs2DataHandler, [77](#)
- sourceReconstructionDeltaCollected
 - Sbs2DataHandler, [78](#)
- sourceReconstructionMethod
 - Sbs2DataHandler, [78](#)
- sourceReconstructionModelUpdateDelta
 - Sbs2DataHandler, [78](#)
- sourceReconstructionModelUpdateLength
 - Sbs2DataHandler, [78](#)
- sourceReconstructionOn
 - Sbs2DataHandler, [78](#)
- sourceReconstructionReady
 - Sbs2DataHandler, [76](#)
- sourceReconstructionSamples
 - Sbs2DataHandler, [78](#)
- sourceReconstructionSpectrogramReady
 - Sbs2DataHandler, [76](#)
- sourceReconstructionSpectrogramValues
 - Sbs2DataHandler, [78](#)
- sourceReconstructionValues
 - Sbs2DataHandler, [78](#)
- Sparse_Matrix_CompRow
 - TNT::Sparse_Matrix_CompRow, [114](#)
- spectrogramChannel
 - Sbs2DataHandler, [76](#)
- spectrogramChannelDelta
 - Sbs2DataHandler, [78](#)
- spectrogramChannelDeltaCollected
 - Sbs2DataHandler, [78](#)
- spectrogramChannelLength
 - Sbs2DataHandler, [78](#)
- spectrogramChannelOn
 - Sbs2DataHandler, [78](#)
- spectrogramChannelSamples
 - Sbs2DataHandler, [78](#)
- spectrogramUpdated
 - Sbs2Callback, [70](#)
 - Sbs2DataHandler, [76](#)
- spectrogramUpdatedSlot
 - Sbs2Callback, [70](#)
- spectrogramValues
 - Sbs2DataHandler, [78](#)
- start
 - Sbs2Emocap28Mounter, [84](#)
 - Sbs2EmocapMounter, [89](#)
 - Sbs2EmotivMounter, [94](#)
 - Sbs2FakeDataReader, [96](#)
 - Sbs2HardwareMounter, [101](#)
 - TNT::Stopwatch, [115](#)
- startRecording
 - Sbs2Callback, [70](#)
 - Sbs2DataHandler, [76](#)
- stop
 - Sbs2Emocap28Mounter, [84](#)
 - Sbs2EmocapMounter, [89](#)
 - Sbs2EmotivMounter, [94](#)
 - Sbs2FakeDataReader, [96](#)
 - Sbs2HardwareMounter, [101](#)
 - TNT::Stopwatch, [115](#)
- stopReconstruction
 - Sbs2SourceReconstruction, [109](#)
- stopRecording
 - Sbs2Callback, [70](#)
 - Sbs2DataHandler, [76](#)
- Stopwatch
 - TNT::Stopwatch, [115](#)
- subarray
 - TNT::Array1D, [36](#)
 - TNT::Array2D, [38](#)
 - TNT::Array3D, [40](#)
 - TNT::Fortran_Array1D, [51](#)
- Subscript
 - TNT, [29](#)
- subtract
 - DTU::DtuArray2D, [46](#)
- SumType
 - Sbs2SourceReconstrucionLoreta, [107](#)
- TNT, [26](#)
 - dot_prod, [29](#)
 - hypot, [29](#)
 - matmult, [29](#)
 - mult_element, [29](#)
 - operator<<, [32](#)
 - operator>>, [32, 33](#)
 - operator*, [29, 30](#)
 - operator*==, [30](#)
 - operator+, [30](#)
 - operator+==, [30, 31](#)
 - operator-, [31](#)
 - operator-=, [31](#)
 - operator/, [31](#)
 - operator/=, [32](#)
 - Subscript, [29](#)
 - transpose, [33](#)
- TNT::Array1D

- ~Array1D, 36
- Array1D, 35, 36
- copy, 36
- dim, 36
- dim1, 36
- inject, 36
- operator const T *, 36
- operator T *, 36
- operator=, 36
- ref, 36
- ref_count, 36
- subarray, 36
- value_type, 35
- TNT::Array1D< T >, 35
- TNT::Array2D
 - ~Array2D, 37
 - Array2D, 37
 - copy, 38
 - data_, 38
 - dim1, 38
 - dim2, 38
 - inject, 38
 - m_, 38
 - n_, 38
 - operator const T **, 38
 - operator T **, 38
 - operator=, 38
 - ref, 38
 - ref_count, 38
 - ref_count_data, 38
 - ref_count_dim1, 38
 - subarray, 38
 - v_, 38
 - value_type, 37
- TNT::Array2D< T >, 36
- TNT::Array3D
 - ~Array3D, 39
 - Array3D, 39
 - copy, 39
 - dim1, 39
 - dim2, 39
 - dim3, 39
 - inject, 39
 - operator const T ***, 40
 - operator T ***, 40
 - operator=, 40
 - ref, 40
 - ref_count, 40
 - subarray, 40
 - value_type, 39
- TNT::Array3D< T >, 38
- TNT::Fortran_Array1D
 - ~Fortran_Array1D, 50
 - copy, 50
 - dim, 50
 - dim1, 50
 - Fortran_Array1D, 50
 - inject, 51
 - operator(), 51
 - operator=, 51
 - ref, 51
 - ref_count, 51
 - subarray, 51
 - value_type, 50
- TNT::Fortran_Array1D< T >, 50
- TNT::Fortran_Array2D
 - ~Fortran_Array2D, 52
 - copy, 52
 - dim1, 52
 - dim2, 52
 - Fortran_Array2D, 52
 - inject, 52
 - operator(), 52
 - operator=, 52
 - ref, 52
 - ref_count, 52
 - value_type, 52
- TNT::Fortran_Array2D< T >, 51
- TNT::Fortran_Array3D
 - ~Fortran_Array3D, 53
 - copy, 53
 - dim1, 53
 - dim2, 53
 - dim3, 53
 - Fortran_Array3D, 53
 - inject, 53
 - operator(), 54
 - operator=, 54
 - ref, 54
 - ref_count, 54
 - value_type, 53
- TNT::Fortran_Array3D< T >, 52
- TNT::Matrix
 - ~Matrix, 62
 - const_iterator, 62
 - const_reference, 62
 - copy, 62
 - destroy, 62
 - dim, 62
 - element_type, 62
 - initialize, 62
 - iterator, 62
 - lbound, 62
 - m_, 63
 - Matrix, 62
 - mn_, 63
 - n_, 63
 - newsize, 62
 - num_cols, 62
 - num_rows, 62
 - operator T **, 62
 - operator(), 62, 63
 - operator=, 63
 - pointer, 62
 - reference, 62
 - row_, 63

- rowm1_, 63
- set, 63
- size, 63
- size_type, 62
- v_, 63
- value_type, 62
- vm1_, 63
- TNT::Matrix< T >, 60
- TNT::Sparse_Matrix_CompRow
 - col_ind, 115
 - dim1, 115
 - dim2, 115
 - NumNonzeros, 115
 - operator=, 115
 - row_ptr, 115
 - Sparse_Matrix_CompRow, 114
 - val, 115
- TNT::Sparse_Matrix_CompRow< T >, 114
- TNT::Stopwatch, 115
 - read, 115
 - resume, 115
 - running, 115
 - start, 115
 - stop, 115
 - Stopwatch, 115
- TNT::Vector
 - ~Vector, 119
 - begin, 119
 - const_iterator, 118
 - const_reference, 118
 - copy, 119
 - destroy, 119
 - dim, 119
 - element_type, 118
 - end, 119
 - initialize, 119
 - iterator, 118
 - lbound, 119
 - n_, 119
 - newsize, 119
 - operator(), 119
 - operator=, 119
 - pointer, 118
 - reference, 118
 - set, 119
 - size, 119
 - size_type, 118
 - v_, 119
 - value_type, 118
 - Vector, 119
 - vm1_, 120
- TNT::Vector< T >, 117
- TNT::i_refvec
 - ~i_refvec, 57
 - begin, 57
 - copy_, 57
 - destroy, 57
 - i_refvec, 57
 - is_null, 58
 - operator=, 58
 - ref_count, 58
 - set_, 58
- TNT::i_refvec< T >, 57
- TNT_BASE_OFFSET
 - tnt_subscript.h, 141
- TNT_MAJOR_VERSION
 - tnt_version.h, 142
- TNT_MINOR_VERSION
 - tnt_version.h, 142
- TNT_SUBSCRIPT_TYPE
 - tnt_subscript.h, 141
- TNT_VERSION_STRING
 - tnt_version.h, 142
- tempModelUpdatedReady
 - Sbs2SourceReconstrucionLoreta, 108
- testDummyRead
 - Sbs2DataReader, 80
- thisPacket
 - Sbs2Callback, 71
 - Sbs2DataHandler, 78
- thread
 - hid_device_, 55
- thresholding_insitu
 - math_utilities.cpp, 159
 - math_utilities.h, 160
- tic
 - Sbs2Timer, 113
- tic_time
 - Sbs2Timer, 113
- timeTick0
 - Sbs2Callback, 70
- timeTick10
 - Sbs2Callback, 70
- timeTick16
 - Sbs2Callback, 70
- timeTick2
 - Sbs2Callback, 70
- timeTick4
 - Sbs2Callback, 70
- timeTick8
 - Sbs2Callback, 71
- tnt_i_refvec.h
 - NULL, 140
- tnt_subscript.h
 - TNT_BASE_OFFSET, 141
- tnt_version.h
 - TNT_MAJOR_VERSION, 142
 - TNT_MINOR_VERSION, 142
- toFilterValues
 - Sbs2DataHandler, 78
- toIdentityMatrix
 - DTU::DtuArray2D, 46
- toSourceReconstructionValues
 - Sbs2DataHandler, 78
- toSpectrogramValues
 - Sbs2DataHandler, 78

- toTntArray2D
 - DTU::DtuArray2D, [46](#)
- toc
 - Sbs2Timer, [113](#)
- trace
 - DTU::DtuArray2D, [46](#)
- transpose
 - DTU::DtuArray2D, [46](#)
 - TNT, [33](#)
- transpose_insitu
 - DTU::DtuArray2D, [46](#)
- trip_count
 - pthread_barrier, [64](#)
- turnChannelSpectrogramOff
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [76](#)
- turnChannelSpectrogramOn
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [76](#)
- turnFilterOff
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [76](#)
- turnFilterOn
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [76](#)
- turnOff
 - Sbs2SourceReconstruction, [109](#)
- turnOffSourceReconstruction
 - Sbs2DataHandler, [76](#)
- turnOnLoreta
 - Sbs2SourceReconstruction, [109](#)
- turnOnSourceReconstructioSparse
 - Sbs2Callback, [71](#)
- turnOnSourceReconstructionLoreta
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [76](#)
- turnOnSourceReconstructionSparse
 - Sbs2DataHandler, [77](#)
- turnOnSparse
 - Sbs2SourceReconstruction, [109](#)
- turnReceiveMessageOff
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [77](#)
 - Sbs2NetworkHandler, [103](#)
- turnReceiveMessageOn
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [77](#)
 - Sbs2NetworkHandler, [103](#)
- turnReceiveRawDataOff
 - Sbs2NetworkHandler, [103](#)
- turnReceiveRawDataOn
 - Sbs2NetworkHandler, [103](#)
- turnReceiveUdpDataOff
 - Sbs2DataReader, [80](#)
 - Sbs2Emocap28DataReader, [83](#)
 - Sbs2EmocapDataReader, [87](#)
 - Sbs2EmotivDataReader, [91](#)
- turnReceiveUdpDataOn
 - Sbs2DataReader, [80](#)
 - Sbs2Emocap28DataReader, [83](#)
 - Sbs2EmocapDataReader, [87](#)
 - Sbs2EmotivDataReader, [92](#)
- turnSendRawDataOff
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [77](#)
 - Sbs2NetworkHandler, [103](#)
- turnSendRawDataOn
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [77](#)
 - Sbs2NetworkHandler, [103](#)
- udpDataReceived
 - Sbs2DataReader, [80](#)
 - Sbs2Emocap28DataReader, [83](#)
 - Sbs2EmocapDataReader, [87](#)
 - Sbs2EmotivDataReader, [92](#)
- udpMessageReceived
 - Sbs2Callback, [71](#)
 - Sbs2DataHandler, [77](#)
- umount
 - Sbs2HardwareMounter, [101](#)
- update
 - Sbs2Emocap28DataContainer, [81](#)
 - Sbs2Emocap28Packet, [86](#)
 - Sbs2EmocapPacket, [90](#)
 - Sbs2EmotivPacket, [95](#)
 - Sbs2FakePacket, [97](#)
 - Sbs2Packet, [104](#)
- updateFilter
 - Sbs2Filter, [99](#)
- usage
 - hid_device_info, [56](#)
- usage_page
 - hid_device_info, [56](#)
- uses_numbered_reports
 - hid_device_, [55](#)
- v_
 - TNT::Array2D, [38](#)
 - TNT::Matrix, [63](#)
 - TNT::Vector, [119](#)
- val
 - TNT::Sparse_Matrix_CompRow, [115](#)
- value_type
 - DTU::DtuArray2D, [45](#)
 - TNT::Array1D, [35](#)
 - TNT::Array2D, [37](#)
 - TNT::Array3D, [39](#)
 - TNT::Fortran_Array1D, [50](#)
 - TNT::Fortran_Array2D, [52](#)
 - TNT::Fortran_Array3D, [53](#)
 - TNT::Matrix, [62](#)
 - TNT::Vector, [118](#)
- values
 - Sbs2Packet, [105](#)
- Vector
 - TNT::Vector, [119](#)

- vectorOuterProduct
 - math_utilities.cpp, [159](#)
 - math_utilities.h, [160](#)
- vendor_id
 - hid_device_info, [56](#)
- verticesNo
 - Sbs2Common, [73](#)
- vm1_
 - TNT::Matrix, [63](#)
 - TNT::Vector, [120](#)
- Waiter, [120](#)
 - run, [120](#)
 - Waiter, [120](#)
- WindowType
 - Sbs2Spectrogram, [112](#)