

BMeasure-lib

1.0.0

Generated by Doxygen 1.8.15

1 Main Page	1
1.1 Introduction	1
1.2 Overview	2
1.3 API Usage	2
1.4 API Usage	3
2 Namespace Index	7
2.1 Namespace List	7
3 Hierarchical Index	9
3.1 Class Hierarchy	9
4 Class Index	11
4.1 Class List	11
5 File Index	13
5.1 File List	13
6 Namespace Documentation	15
6.1 BMeasureApi Namespace Reference	15
6.1.1 Typedef Documentation	19
6.1.1.1 ChannelConfigs	19
6.1.2 Enumeration Type Documentation	19
6.1.2.1 AlarmMode	19
6.1.2.2 AlarmOutput	19
6.1.2.3 AwgMode	20
6.1.2.4 AwgOutput	20
6.1.2.5 BlockTypes	20
6.1.2.6 BMeasFileType	21
6.1.2.7 CalibrateStage	21
6.1.2.8 ChannelType	21
6.1.2.9 DataSend	22
6.1.2.10 DataType	22
6.1.2.11 DigitalMode	22
6.1.2.12 ErrorNum	22
6.1.2.13 EventMode	23
6.1.2.14 FilesysDeleteType	23
6.1.2.15 FileType	23
6.1.2.16 LogData	24
6.1.2.17 LogDataMode	24
6.1.2.18 MeasureMode	24
6.1.2.19 MeasureOption	24
6.1.2.20 MessageSource	25
6.1.2.21 Mode	25

6.1.2.22 NetworkMode	25
6.1.2.23 NodeType	25
6.1.2.24 Rs485Mode	26
6.1.2.25 SampleType	26
6.1.2.26 SecurityMode	26
6.1.2.27 Status	27
6.1.2.28 SyncMode	27
6.1.2.29 TdsDataType	27
6.1.2.30 TriggerConfig	28
6.1.2.31 TriggerMode	28
6.1.2.32 WifiMode	28
6.1.3 Function Documentation	29
6.1.3.1 channelTypeString()	29
6.1.3.2 fromBString() [1/31]	29
6.1.3.3 fromBString() [2/31]	29
6.1.3.4 fromBString() [3/31]	29
6.1.3.5 fromBString() [4/31]	29
6.1.3.6 fromBString() [5/31]	29
6.1.3.7 fromBString() [6/31]	30
6.1.3.8 fromBString() [7/31]	30
6.1.3.9 fromBString() [8/31]	30
6.1.3.10 fromBString() [9/31]	30
6.1.3.11 fromBString() [10/31]	30
6.1.3.12 fromBString() [11/31]	30
6.1.3.13 fromBString() [12/31]	31
6.1.3.14 fromBString() [13/31]	31
6.1.3.15 fromBString() [14/31]	31
6.1.3.16 fromBString() [15/31]	31
6.1.3.17 fromBString() [16/31]	31
6.1.3.18 fromBString() [17/31]	31
6.1.3.19 fromBString() [18/31]	32
6.1.3.20 fromBString() [19/31]	32
6.1.3.21 fromBString() [20/31]	32
6.1.3.22 fromBString() [21/31]	32
6.1.3.23 fromBString() [22/31]	32
6.1.3.24 fromBString() [23/31]	32
6.1.3.25 fromBString() [24/31]	33
6.1.3.26 fromBString() [25/31]	33
6.1.3.27 fromBString() [26/31]	33
6.1.3.28 fromBString() [27/31]	33
6.1.3.29 fromBString() [28/31]	33
6.1.3.30 fromBString() [29/31]	33

6.1.3.31 fromBString() [30/31]	34
6.1.3.32 fromBString() [31/31]	34
6.1.3.33 round512()	34
6.1.3.34 roundDown512()	34
6.1.3.35 sampleTypeString()	34
6.1.3.36 toBString() [1/31]	34
6.1.3.37 toBString() [2/31]	34
6.1.3.38 toBString() [3/31]	35
6.1.3.39 toBString() [4/31]	35
6.1.3.40 toBString() [5/31]	35
6.1.3.41 toBString() [6/31]	35
6.1.3.42 toBString() [7/31]	35
6.1.3.43 toBString() [8/31]	35
6.1.3.44 toBString() [9/31]	35
6.1.3.45 toBString() [10/31]	36
6.1.3.46 toBString() [11/31]	36
6.1.3.47 toBString() [12/31]	36
6.1.3.48 toBString() [13/31]	36
6.1.3.49 toBString() [14/31]	36
6.1.3.50 toBString() [15/31]	36
6.1.3.51 toBString() [16/31]	36
6.1.3.52 toBString() [17/31]	37
6.1.3.53 toBString() [18/31]	37
6.1.3.54 toBString() [19/31]	37
6.1.3.55 toBString() [20/31]	37
6.1.3.56 toBString() [21/31]	37
6.1.3.57 toBString() [22/31]	37
6.1.3.58 toBString() [23/31]	37
6.1.3.59 toBString() [24/31]	38
6.1.3.60 toBString() [25/31]	38
6.1.3.61 toBString() [26/31]	38
6.1.3.62 toBString() [27/31]	38
6.1.3.63 toBString() [28/31]	38
6.1.3.64 toBString() [29/31]	38
6.1.3.65 toBString() [30/31]	38
6.1.3.66 toBString() [31/31]	39
6.1.3.67 toBStringJson() [1/31]	39
6.1.3.68 toBStringJson() [2/31]	39
6.1.3.69 toBStringJson() [3/31]	39
6.1.3.70 toBStringJson() [4/31]	39
6.1.3.71 toBStringJson() [5/31]	39
6.1.3.72 toBStringJson() [6/31]	40

6.1.3.73 toBStringJson() [7/31]	40
6.1.3.74 toBStringJson() [8/31]	40
6.1.3.75 toBStringJson() [9/31]	40
6.1.3.76 toBStringJson() [10/31]	40
6.1.3.77 toBStringJson() [11/31]	40
6.1.3.78 toBStringJson() [12/31]	41
6.1.3.79 toBStringJson() [13/31]	41
6.1.3.80 toBStringJson() [14/31]	41
6.1.3.81 toBStringJson() [15/31]	41
6.1.3.82 toBStringJson() [16/31]	41
6.1.3.83 toBStringJson() [17/31]	41
6.1.3.84 toBStringJson() [18/31]	42
6.1.3.85 toBStringJson() [19/31]	42
6.1.3.86 toBStringJson() [20/31]	42
6.1.3.87 toBStringJson() [21/31]	42
6.1.3.88 toBStringJson() [22/31]	42
6.1.3.89 toBStringJson() [23/31]	42
6.1.3.90 toBStringJson() [24/31]	43
6.1.3.91 toBStringJson() [25/31]	43
6.1.3.92 toBStringJson() [26/31]	43
6.1.3.93 toBStringJson() [27/31]	43
6.1.3.94 toBStringJson() [28/31]	43
6.1.3.95 toBStringJson() [29/31]	43
6.1.3.96 toBStringJson() [30/31]	44
6.1.3.97 toBStringJson() [31/31]	44
6.1.3.98 TocBigEndian()	44
6.1.3.99 TocDaqRawData()	44
6.1.3.100 TocInterleavedData()	44
6.1.3.101 TocMetaData()	44
6.1.3.102 TocNewObjList()	44
6.1.3.103 TocRawData()	45
6.1.3.104 toFloat()	45
6.1.3.105 unitSort()	45
6.1.4 Variable Documentation	45
6.1.4.1 apiVersion	45
7 Class Documentation	47
7.1 BMeasureApi::AlarmConfig Class Reference	47
7.1.1 Member Function Documentation	47
7.1.1.1 getMembers()	47
7.1.2 Member Data Documentation	48
7.1.2.1 levelHigh	48

7.1.2.2 levelLow	48
7.1.2.3 mode	48
7.1.2.4 output	48
7.1.2.5 outputChannel	48
7.1.2.6 spare	49
7.2 BMeasureApi::AwgConfig Class Reference	49
7.2.1 Member Function Documentation	49
7.2.1.1 getMembers()	49
7.2.2 Member Data Documentation	50
7.2.2.1 amplitude	50
7.2.2.2 duty	50
7.2.2.3 frequency	50
7.2.2.4 mode	50
7.2.2.5 numSamples	50
7.2.2.6 offset	51
7.2.2.7 output	51
7.2.2.8 spare	51
7.2.2.9 trackChannel	51
7.3 BFirmwareInfo Struct Reference	51
7.3.1 Member Data Documentation	51
7.3.1.1 checksum	52
7.3.1.2 length	52
7.3.1.3 magic	52
7.3.1.4 type	52
7.3.1.5 ver0	52
7.3.1.6 ver1	52
7.3.1.7 ver2	52
7.4 BMdns Class Reference	53
7.4.1 Constructor & Destructor Documentation	53
7.4.1.1 BMdns()	53
7.4.1.2 ~BMdns()	53
7.4.2 Member Function Documentation	53
7.4.2.1 findServices()	53
7.4.2.2 init()	54
7.4.3 Member Data Documentation	54
7.4.3.1 osocket	54
7.4.3.2 otransactionId	54
7.5 BMdnsService Class Reference	54
7.5.1 Member Data Documentation	54
7.5.1.1 address	54
7.5.1.2 extra	55
7.5.1.3 hostname	55

7.5.1.4 name	55
7.6 BMeasureApi::BMeasure Class Reference	55
7.6.1 Constructor & Destructor Documentation	58
7.6.1.1 BMeasure()	58
7.6.2 Member Function Documentation	58
7.6.2.1 alarmsClear()	59
7.6.2.2 alarmsClearServe()	59
7.6.2.3 calibrate()	59
7.6.2.4 calibrateServe()	59
7.6.2.5 changePassword()	59
7.6.2.6 changePasswordServe()	59
7.6.2.7 factoryReset()	60
7.6.2.8 factoryResetServe()	60
7.6.2.9 fileClose()	60
7.6.2.10 fileCloseServe()	60
7.6.2.11 fileDelete()	60
7.6.2.12 fileDeleteServe()	60
7.6.2.13 fileList()	61
7.6.2.14 fileListServe()	61
7.6.2.15 fileOpen()	61
7.6.2.16 fileOpenServe()	61
7.6.2.17 fileRead()	61
7.6.2.18 fileReadServe()	62
7.6.2.19 filesysDelete()	62
7.6.2.20 filesysDeleteServe()	62
7.6.2.21 filesysInfo()	62
7.6.2.22 filesysInfoServe()	62
7.6.2.23 fileWrite()	62
7.6.2.24 fileWriteServe()	63
7.6.2.25 functionUnLock()	63
7.6.2.26 functionUnLockServe()	63
7.6.2.27 getAwgConfig()	63
7.6.2.28 getAwgConfigServe()	63
7.6.2.29 getBoardConfig()	63
7.6.2.30 getBoardConfigServe()	64
7.6.2.31 getChannelConfig()	64
7.6.2.32 getChannelConfigServe()	64
7.6.2.33 getConfig()	64
7.6.2.34 getConfigServe()	64
7.6.2.35 getDigital()	64
7.6.2.36 getDigitalServe()	65
7.6.2.37 getInfoBlock()	65

7.6.2.38	getInfoBlockServe()	65
7.6.2.39	getInformation()	65
7.6.2.40	getInformationServe()	65
7.6.2.41	getMeasurementConfig()	65
7.6.2.42	getMeasurementConfigServe()	66
7.6.2.43	getNodeInfo()	66
7.6.2.44	getNodeInfoServe()	66
7.6.2.45	getStatus()	66
7.6.2.46	getStatusServe()	66
7.6.2.47	getSwitch()	66
7.6.2.48	getSwitchServe()	67
7.6.2.49	login()	67
7.6.2.50	loginServe()	67
7.6.2.51	logout()	67
7.6.2.52	logoutServe()	67
7.6.2.53	measure()	67
7.6.2.54	measureServe()	68
7.6.2.55	processRequest()	68
7.6.2.56	runBoardTest()	68
7.6.2.57	runBoardTestServe()	68
7.6.2.58	sendChannelConfig()	68
7.6.2.59	sendChannelConfigServe()	68
7.6.2.60	sendData()	69
7.6.2.61	sendDataEnable()	69
7.6.2.62	sendDataEnableServe()	69
7.6.2.63	sendDataServe()	69
7.6.2.64	sendInfo()	69
7.6.2.65	sendInfoServe()	69
7.6.2.66	sendMessage()	70
7.6.2.67	sendMessageServe()	70
7.6.2.68	sendStatus()	70
7.6.2.69	sendStatusServe()	70
7.6.2.70	sendTime()	70
7.6.2.71	sendTimeServe()	70
7.6.2.72	setAnalogueOut()	71
7.6.2.73	setAnalogueOutServe()	71
7.6.2.74	setAwgConfig()	71
7.6.2.75	setAwgConfigServe()	71
7.6.2.76	setAwgWaveform()	71
7.6.2.77	setAwgWaveformServe()	72
7.6.2.78	setBoardConfig()	72
7.6.2.79	setBoardConfigServe()	72

7.6.2.80	setChannelConfig()	72
7.6.2.81	setChannelConfigFull()	72
7.6.2.82	setChannelConfigFullServe()	72
7.6.2.83	setChannelConfigServe()	73
7.6.2.84	setConfig()	73
7.6.2.85	setConfigServe()	73
7.6.2.86	setDigital()	73
7.6.2.87	setDigitalServe()	73
7.6.2.88	setMeasurementConfig()	73
7.6.2.89	setMeasurementConfigServe()	74
7.6.2.90	setMode()	74
7.6.2.91	setModeServe()	74
7.6.2.92	setRelay()	74
7.6.2.93	setRelayServe()	74
7.7	BMeasureApi::BMeasureUnit Class Reference	75
7.7.1	Constructor & Destructor Documentation	76
7.7.1.1	BMeasureUnit()	76
7.7.1.2	~BMeasureUnit()	76
7.7.2	Member Function Documentation	76
7.7.2.1	connect()	76
7.7.2.2	device()	77
7.7.2.3	disconnect()	77
7.7.2.4	disconnected()	77
7.7.2.5	findDevices()	77
7.7.2.6	findDevicesNetwork()	77
7.7.2.7	findDevicesUsb()	77
7.7.2.8	info()	78
7.7.2.9	numChannels()	78
7.7.2.10	processdataBlock()	78
7.7.2.11	run()	78
7.7.2.12	sendDataServe()	78
7.7.2.13	sendDataServe1()	78
7.7.2.14	serialNumber()	79
7.7.2.15	setChannelConfig()	79
7.7.2.16	setMeasurementConfig()	79
7.7.3	Member Data Documentation	79
7.7.3.1	blockNumChannels	79
7.7.3.2	blockNumSamples	79
7.7.3.3	oblockCount	79
7.7.3.4	ochannels	79
7.7.3.5	oconfigMeasurement	80
7.7.3.6	odataBlock	80

7.7.3.7 odevice	80
7.7.3.8 odisconnecting	80
7.7.3.9 oinfo	80
7.7.3.10 onodeInfo	80
7.7.3.11 osampleCount	80
7.7.3.12 osequenceNext	81
7.8 BMeasureApi::BMeasureUnit1 Class Reference	81
7.8.1 Constructor & Destructor Documentation	81
7.8.1.1 BMeasureUnit1()	82
7.8.2 Member Function Documentation	82
7.8.2.1 disconnected()	82
7.8.2.2 sendDataServe1()	82
7.8.2.3 sendMessageServe()	82
7.8.2.4 serialNumber()	82
7.8.2.5 setSerialNumber()	83
7.8.3 Member Data Documentation	83
7.8.3.1 oconnected	83
7.8.3.2 oenabled	83
7.8.3.3 omeasureUnits	83
7.8.3.4 oorder	83
7.8.3.5 oserialNumber	83
7.8.3.6 osource	83
7.9 BMeasureApi::BMeasureUnitDevice Class Reference	84
7.9.1 Constructor & Destructor Documentation	84
7.9.1.1 BMeasureUnitDevice()	84
7.9.2 Member Data Documentation	84
7.9.2.1 device	84
7.9.2.2 serialNumber	84
7.10 BMeasureApi::BMeasureUnits Class Reference	85
7.10.1 Constructor & Destructor Documentation	87
7.10.1.1 BMeasureUnits()	87
7.10.1.2 ~BMeasureUnits()	87
7.10.2 Member Function Documentation	87
7.10.2.1 alarmsClear()	87
7.10.2.2 changePassword()	88
7.10.2.3 clear()	88
7.10.2.4 dataAvailable()	88
7.10.2.5 dataClear()	88
7.10.2.6 dataDone()	88
7.10.2.7 dataEvent()	88
7.10.2.8 dataProcDone()	88
7.10.2.9 dataProcEvent()	89

7.10.2.10 dataProcRead()	89
7.10.2.11 dataRead()	89
7.10.2.12 dataSetNumStreams()	89
7.10.2.13 dataStreamEnable()	89
7.10.2.14 dataWait()	89
7.10.2.15 debugPrint()	90
7.10.2.16 disconnected()	90
7.10.2.17 getAwgConfig()	90
7.10.2.18 getChannelConfig()	90
7.10.2.19 getConfig()	90
7.10.2.20 getFreeBlock()	90
7.10.2.21 getInfoBlock()	91
7.10.2.22 getInformation()	91
7.10.2.23 getMeasurementConfig()	91
7.10.2.24 getNodeInfo()	91
7.10.2.25 getStatus()	91
7.10.2.26 login()	91
7.10.2.27 logout()	92
7.10.2.28 numChannels()	92
7.10.2.29 outputBlock()	92
7.10.2.30 run()	92
7.10.2.31 sendDataEnable()	92
7.10.2.32 sendDataProcess()	92
7.10.2.33 sendDataProcessTrigger()	93
7.10.2.34 sendDataProcQueue()	93
7.10.2.35 sendDataQueue()	93
7.10.2.36 sendDataServe1()	93
7.10.2.37 sendMessage()	93
7.10.2.38 sendMessageServe()	93
7.10.2.39 sendTime()	93
7.10.2.40 setAwgConfig()	94
7.10.2.41 setChannelConfig()	94
7.10.2.42 setConfig()	94
7.10.2.43 setMeasurementConfig()	94
7.10.2.44 setMode()	94
7.10.2.45 unit()	95
7.10.2.46 unitAdd()	95
7.10.2.47 unitDelete()	95
7.10.2.48 unitMaster()	95
7.10.2.49 unitsConnect()	95
7.10.2.50 unitsConnected()	95
7.10.2.51 unitsConnectedNum()	95

7.10.2.52 unitsDisconnect()	96
7.10.2.53 unitSetEnabled()	96
7.10.2.54 unitSetOrder()	96
7.10.2.55 unitsFind()	96
7.10.2.56 unitsNum()	96
7.10.3 Member Data Documentation	96
7.10.3.1 odataBlocksFree	96
7.10.3.2 odataBlocksIn	97
7.10.3.3 odataBlocksOut	97
7.10.3.4 odataBlocksOutCount	97
7.10.3.5 odataBlocksProcess	97
7.10.3.6 odataBlocksProcessNum	97
7.10.3.7 odataProcBlocks	97
7.10.3.8 odataStreamNum	97
7.10.3.9 ofill	97
7.10.3.10 olocalTrigger	98
7.10.3.11 olockInput	98
7.10.3.12 olockOutput	98
7.10.3.13 olockProInput	98
7.10.3.14 olockUnits	98
7.10.3.15 onumBlocks	98
7.10.3.16 onumChannels	98
7.10.3.17 onumConnected	98
7.10.3.18 oprocEnable	99
7.10.3.19 oprocRunning	99
7.10.3.20 ostartSample	99
7.10.3.21 otriggered	99
7.10.3.22 ounitMaster	99
7.10.3.23 ounits	99
7.11 BMeasureApi::BMeasureUnitsDataBlock Class Reference	99
7.11.1 Constructor & Destructor Documentation	100
7.11.1.1 BMeasureUnitsDataBlock()	100
7.11.1.2 ~BMeasureUnitsDataBlock()	100
7.11.2 Member Function Documentation	100
7.11.2.1 init()	100
7.11.3 Member Data Documentation	100
7.11.3.1 odataBlock	101
7.11.3.2 ofill	101
7.11.3.3 oinUse	101
7.12 BMeasureApi::BoardConfig Class Reference	101
7.12.1 Member Function Documentation	102
7.12.1.1 getMembers()	102

7.12.2 Member Data Documentation	102
7.12.2.1 buildTime	102
7.12.2.2 calibAdcOffsets	102
7.12.2.3 calibAdcScales	102
7.12.2.4 calibAttenScales	102
7.12.2.5 calibDacOffsets	102
7.12.2.6 calibDacScales	103
7.12.2.7 calibFiveVolts	103
7.12.2.8 calibTemp	103
7.12.2.9 calibTime	103
7.12.2.10 fpgaVersion	103
7.12.2.11 hardwareVersion	103
7.12.2.12 macAddress	103
7.12.2.13 magic	103
7.12.2.14 serialNumber	104
7.12.2.15 spare	104
7.12.2.16 spare0	104
7.12.2.17 testMode	104
7.12.2.18 wifiVersion	104
7.13 BMeasureApi::CalibrateInfo Class Reference	104
7.13.1 Member Function Documentation	105
7.13.1.1 getMembers()	105
7.13.2 Member Data Documentation	105
7.13.2.1 calibrateAmplitude	105
7.13.2.2 calibrateFrequency	105
7.13.2.3 calibrateTime	105
7.13.2.4 stage	106
7.13.2.5 value	106
7.14 BMeasureApi::ChannelConfig Class Reference	106
7.14.1 Member Function Documentation	107
7.14.1.1 getMembers()	107
7.14.2 Member Data Documentation	107
7.14.2.1 attenuator	107
7.14.2.2 calibOffset	107
7.14.2.3 calibScale	107
7.14.2.4 calibScaleAtten1	108
7.14.2.5 dataChannel	108
7.14.2.6 enabled	108
7.14.2.7 id	108
7.14.2.8 name	108
7.14.2.9 number	108
7.14.2.10 offset	109

7.14.2.11 pgaGain	109
7.14.2.12 process	109
7.14.2.13 sampleType	109
7.14.2.14 scale	109
7.14.2.15 siUnits	109
7.14.2.16 spare0	109
7.14.2.17 type	110
7.15 BMeasureApi::CommsNet Class Reference	110
7.15.1 Constructor & Destructor Documentation	110
7.15.1.1 CommsNet()	111
7.15.1.2 ~CommsNet()	111
7.15.2 Member Function Documentation	111
7.15.2.1 connect()	111
7.15.2.2 disconnect()	111
7.15.2.3 init()	111
7.15.2.4 read()	111
7.15.2.5 readAvailable()	112
7.15.2.6 wait()	112
7.15.2.7 write()	112
7.15.2.8 writeAvailable()	112
7.15.2.9 writeChunks()	112
7.15.3 Member Data Documentation	112
7.15.3.1 oinWait	113
7.15.3.2 osocket	113
7.15.3.3 oterminating	113
7.16 BMeasureApi::CommsSerial Class Reference	113
7.16.1 Constructor & Destructor Documentation	114
7.16.1.1 CommsSerial()	114
7.16.1.2 ~CommsSerial()	114
7.16.2 Member Function Documentation	114
7.16.2.1 connect()	114
7.16.2.2 disconnect()	114
7.16.2.3 read()	114
7.16.2.4 readAvailable()	115
7.16.2.5 wait()	115
7.16.2.6 write()	115
7.16.3 Member Data Documentation	115
7.16.3.1 odevice	115
7.16.3.2 oserialPort	115
7.17 BMeasureApi::CommsUsb Class Reference	116
7.17.1 Constructor & Destructor Documentation	116
7.17.1.1 CommsUsb()	116

7.17.1.2 ~CommsUsb()	117
7.17.2 Member Function Documentation	117
7.17.2.1 connect()	117
7.17.2.2 disconnect()	117
7.17.2.3 read()	117
7.17.2.4 readAvailable()	117
7.17.2.5 readChunk()	117
7.17.2.6 wait()	118
7.17.2.7 write()	118
7.17.3 Member Data Documentation	118
7.17.3.1 obuffer	118
7.17.3.2 ocontext	118
7.17.3.3 odev	118
7.17.3.4 odevice	118
7.17.3.5 onum	119
7.17.3.6 oterminated	119
7.17.3.7 oterminating	119
7.18 BMeasureApi::ConfigItem Class Reference	119
7.18.1 Member Function Documentation	119
7.18.1.1 getMembers()	119
7.18.2 Member Data Documentation	120
7.18.2.1 name	120
7.18.2.2 spare	120
7.18.2.3 type	120
7.18.2.4 value	120
7.19 BMeasureApi::Configuration Class Reference	120
7.19.1 Member Function Documentation	122
7.19.1.1 getMembers()	122
7.19.2 Member Data Documentation	122
7.19.2.1 alarms	122
7.19.2.2 digitalMode	122
7.19.2.3 digitalPins	122
7.19.2.4 emailAddress	123
7.19.2.5 emailMode	123
7.19.2.6 location	123
7.19.2.7 logData	123
7.19.2.8 logDataDevice	123
7.19.2.9 logDataMode	123
7.19.2.10 mode	124
7.19.2.11 mqttMode	124
7.19.2.12 mqttPort	124
7.19.2.13 mqttServer	124

7.19.2.14 name	124
7.19.2.15 networkAddress	124
7.19.2.16 networkGateway	125
7.19.2.17 networkMask	125
7.19.2.18 networkMode	125
7.19.2.19 networkNameServer0	125
7.19.2.20 networkTimeServer	125
7.19.2.21 program	125
7.19.2.22 rs485BaudRate	126
7.19.2.23 rs485Bits	126
7.19.2.24 rs485Mode	126
7.19.2.25 rs485StopBits	126
7.19.2.26 sampleFrequencyMode	126
7.19.2.27 securityMode	126
7.19.2.28 source	127
7.19.2.29 spare1	127
7.19.2.30 spare2	127
7.19.2.31 spare3	127
7.19.2.32 spare4	127
7.19.2.33 spare5	127
7.19.2.34 spare6	127
7.19.2.35 version	128
7.19.2.36 wifiAp0	128
7.19.2.37 wifiMode	128
7.20 BMeasureApi::DataBlock Class Reference	128
7.20.1 Member Function Documentation	129
7.20.1.1 getMembers()	129
7.20.2 Member Data Documentation	129
7.20.2.1 data	129
7.20.2.2 numChannels	129
7.20.2.3 numSamples	129
7.20.2.4 sequence	129
7.20.2.5 source	130
7.20.2.6 spare	130
7.20.2.7 status	130
7.20.2.8 time	130
7.20.2.9 type	130
7.21 BMeasureApi::DataBlockProc Class Reference	130
7.21.1 Member Function Documentation	131
7.21.1.1 getMembers()	131
7.21.2 Member Data Documentation	131
7.21.2.1 analogueData	131

7.21.2.2 digitalData	131
7.21.2.3 numChannels	132
7.21.2.4 numSamples	132
7.21.2.5 sequence	132
7.21.2.6 source	132
7.21.2.7 spare	132
7.21.2.8 status	132
7.21.2.9 time	133
7.21.2.10 type	133
7.22 BMeasureApi::DataFile Class Reference	133
7.22.1 Constructor & Destructor Documentation	134
7.22.1.1 DataFile()	134
7.22.1.2 ~DataFile()	134
7.22.2 Member Function Documentation	134
7.22.2.1 close()	134
7.22.2.2 getFileName()	135
7.22.2.3 init()	135
7.22.2.4 open()	135
7.22.2.5 readData()	135
7.22.2.6 readInfo()	135
7.22.2.7 validateFormat()	135
7.22.2.8 writeData() [1/2]	136
7.22.2.9 writeData() [2/2]	136
7.22.2.10 writeEnd()	136
7.22.2.11 writeInfo()	136
7.22.2.12 writeInfoBMeas()	136
7.22.2.13 writeInfoCsv()	136
7.22.2.14 writeInfoTdms()	137
7.22.3 Member Data Documentation	137
7.22.3.1 ofile	137
7.22.3.2 ofileName	137
7.22.3.3 oformat	137
7.22.3.4 omode	137
7.22.3.5 opacket	137
7.22.3.6 opacketLen	137
7.23 BMeasureApi::DataProc Class Reference	138
7.23.1 Member Function Documentation	138
7.23.1.1 getMembers()	138
7.23.2 Member Data Documentation	138
7.23.2.1 alarm	138
7.23.2.2 mean	139
7.23.2.3 peakHigh	139

7.23.2.4 peakLow	139
7.23.2.5 rms	139
7.23.2.6 spare	139
7.24 Dfu Class Reference	139
7.24.1 Detailed Description	140
7.24.2 Constructor & Destructor Documentation	140
7.24.2.1 Dfu()	140
7.24.2.2 ~Dfu()	140
7.24.3 Member Function Documentation	141
7.24.3.1 clearStatus()	141
7.24.3.2 connect()	141
7.24.3.3 detectDevice()	141
7.24.3.4 disconnect()	141
7.24.3.5 getStatus()	141
7.24.3.6 init()	141
7.24.3.7 reset()	142
7.24.3.8 upload()	142
7.24.3.9 upload_cmd()	142
7.24.3.10 validateFile()	142
7.24.4 Member Data Documentation	142
7.24.4.1 oconnected	142
7.24.4.2 ocontext	143
7.24.4.3 odev	143
7.24.4.4 overbose	143
7.25 DfuStatus Struct Reference	143
7.25.1 Member Data Documentation	143
7.25.1.1 iString	143
7.25.1.2 pollTimeout	144
7.25.1.3 state	144
7.25.1.4 status	144
7.26 BMeasureApi::FileData Class Reference	144
7.26.1 Member Function Documentation	144
7.26.1.1 getMembers()	144
7.26.2 Member Data Documentation	145
7.26.2.1 data	145
7.26.2.2 length	145
7.27 BMeasureApi::FileInfo Class Reference	145
7.27.1 Member Function Documentation	145
7.27.1.1 getMembers()	146
7.27.2 Member Data Documentation	146
7.27.2.1 fileLength	146
7.27.2.2 fileType	146

7.27.2.3 name	146
7.27.2.4 spare	146
7.27.2.5 time	146
7.28 BMeasureApi::FilesysInfo Class Reference	147
7.28.1 Member Function Documentation	147
7.28.1.1 getMembers()	147
7.28.2 Member Data Documentation	147
7.28.2.1 free	147
7.28.2.2 name	147
7.28.2.3 size	148
7.29 BMeasureApi::InfoBlock Class Reference	148
7.29.1 Member Function Documentation	148
7.29.1.1 getMembers()	148
7.29.2 Member Data Documentation	149
7.29.2.1 dataType	149
7.29.2.2 fileType	149
7.29.2.3 location	149
7.29.2.4 measureConfig	149
7.29.2.5 name	149
7.29.2.6 nodeInfo	149
7.29.2.7 numChannels	150
7.29.2.8 source	150
7.29.2.9 time	150
7.29.2.10 version	150
7.30 BMeasureApi::Information Class Reference	150
7.30.1 Member Function Documentation	151
7.30.1.1 getMembers()	151
7.30.2 Member Data Documentation	151
7.30.2.1 calibTime	152
7.30.2.2 networkAddress	152
7.30.2.3 networkGateway	152
7.30.2.4 networkMacAddress	152
7.30.2.5 networkMask	152
7.30.2.6 networkMode	152
7.30.2.7 networkNameServer0	153
7.30.2.8 networkTimeServer	153
7.30.2.9 nodeInfo	153
7.30.2.10 numChannels	153
7.30.2.11 numConfigItems	153
7.30.2.12 spare0	153
7.30.2.13 spare1	153
7.30.2.14 spare2	154

7.30.2.15 spare3	154
7.30.2.16 time	154
7.30.2.17 wifiAddress	154
7.30.2.18 wifiGateway	154
7.30.2.19 wifiMacAddress	154
7.30.2.20 wifiMask	154
7.30.2.21 wifiMode	155
7.31 BMeasureApi::MeasurementConfig Class Reference	155
7.31.1 Member Function Documentation	156
7.31.1.1 getMembers()	156
7.31.2 Member Data Documentation	156
7.31.2.1 description	156
7.31.2.2 measureMode	156
7.31.2.3 measureOptions	156
7.31.2.4 measurePeriod	156
7.31.2.5 numSamples0	156
7.31.2.6 numSamples1	157
7.31.2.7 numSamples2	157
7.31.2.8 numSamplesBlock	157
7.31.2.9 peakFilter	157
7.31.2.10 sampleRate	157
7.31.2.11 spare1	157
7.31.2.12 spare2	157
7.31.2.13 triggerChannel	158
7.31.2.14 triggerConfig	158
7.31.2.15 triggerDelay	158
7.31.2.16 triggerLevel	158
7.31.2.17 triggerMode	158
7.32 BMeasureApi::NodeInfo Class Reference	158
7.32.1 Member Function Documentation	159
7.32.1.1 getMembers()	159
7.32.2 Member Data Documentation	159
7.32.2.1 apiVersion	159
7.32.2.2 fpgaVersion	159
7.32.2.3 hardwareVersion	159
7.32.2.4 securityMode	159
7.32.2.5 serialNumber	160
7.32.2.6 softwareVersion	160
7.32.2.7 spare	160
7.32.2.8 variant	160
7.32.2.9 wifiVersion	160
7.33 BMeasureApi::NodeStatus Class Reference	160

7.33.1 Member Function Documentation	161
7.33.1.1 getMembers()	161
7.33.2 Member Data Documentation	161
7.33.2.1 error	161
7.33.2.2 errorStr	161
7.33.2.3 ethernetStatus	161
7.33.2.4 mode	161
7.33.2.5 spare	162
7.33.2.6 status	162
7.33.2.7 time	162
7.33.2.8 wifiStatus	162
7.34 BMeasureApi::Version Class Reference	162
7.34.1 Member Function Documentation	162
7.34.1.1 getMembers()	163
7.34.2 Member Data Documentation	163
7.34.2.1 type	163
7.34.2.2 ver0	163
7.34.2.3 ver1	163
7.34.2.4 ver2	163
8 File Documentation	165
8.1 BMdns.cpp File Reference	165
8.1.1 Macro Definition Documentation	166
8.1.1.1 BDEBUGL1	166
8.1.2 Enumeration Type Documentation	166
8.1.2.1 MdnsClass	166
8.1.2.2 MdnsEntryType	166
8.1.2.3 MdnsRecordType	166
8.1.3 Function Documentation	167
8.1.3.1 mdns_read_string()	167
8.1.3.2 mdns_read_strings()	167
8.1.3.3 mdns_write_string()	167
8.2 BMdns.h File Reference	167
8.3 BMeasureB-1.cpp File Reference	167
8.4 BMeasureB.cpp File Reference	168
8.5 BMeasureB.h File Reference	168
8.6 BMeasureD.cpp File Reference	168
8.6.1 Macro Definition Documentation	170
8.6.1.1 boffsetof	171
8.7 BMeasureD.h File Reference	171
8.8 BMeasureLib.cpp File Reference	175
8.8.1 Macro Definition Documentation	175

8.8.1.1 BDEBUGL1	175
8.8.1.2 BDEBUGL2	176
8.8.2 Function Documentation	176
8.8.2.1 toBStringJson() [1/3]	176
8.8.2.2 toBStringJson() [2/3]	176
8.8.2.3 toBStringJson() [3/3]	176
8.9 BMeasureLib.h File Reference	176
8.9.1 Function Documentation	177
8.9.1.1 toBStringJson() [1/3]	177
8.9.1.2 toBStringJson() [2/3]	177
8.9.1.3 toBStringJson() [3/3]	177
8.10 BMeasureS.cpp File Reference	177
8.11 BMeasureUnit.cpp File Reference	178
8.11.1 Macro Definition Documentation	178
8.11.1.1 BDEBUGL1	178
8.11.1.2 BDEBUGL2	178
8.11.1.3 BDEBUGL3	179
8.11.1.4 CONVERT_FLOAT	179
8.12 BMeasureUnit.h File Reference	179
8.13 BMeasureUnits.cpp File Reference	179
8.13.1 Macro Definition Documentation	180
8.13.1.1 BDEBUGL1	180
8.13.1.2 BDEBUGL2	180
8.13.1.3 BDEBUGL3	180
8.14 BMeasureUnits.h File Reference	180
8.15 CommsNet.cpp File Reference	181
8.15.1 Macro Definition Documentation	181
8.15.1.1 BDEBUGL1	181
8.15.1.2 BDEBUGL2	181
8.15.1.3 BDEBUGL3	181
8.16 CommsNet.h File Reference	182
8.17 CommsSerial.cpp File Reference	182
8.18 CommsSerial.h File Reference	182
8.19 CommsUsb.cpp File Reference	182
8.19.1 Macro Definition Documentation	183
8.19.1.1 BDEBUGL1	183
8.19.1.2 BDEBUGL2	183
8.20 CommsUsb.h File Reference	183
8.21 DataFile.cpp File Reference	183
8.21.1 Macro Definition Documentation	184
8.21.1.1 BDEBUGL1	184
8.21.1.2 BDEBUGL2	184

8.22 DataFile.h File Reference	185
8.23 Dfu.cpp File Reference	185
8.23.1 Macro Definition Documentation	186
8.23.1.1 BDEBUGL1	186
8.23.1.2 BDEBUGL2	187
8.23.1.3 DFU_ABORT	187
8.23.1.4 DFU_CLRSTATUS	187
8.23.1.5 DFU_DETACH	187
8.23.1.6 DFU_DNLOAD	187
8.23.1.7 DFU_GETSTATE	187
8.23.1.8 DFU_GETSTATUS	187
8.23.1.9 DFU_IFF_ALT	187
8.23.1.10 DFU_IFF_CONFIG	188
8.23.1.11 DFU_IFF_DEVNUM	188
8.23.1.12 DFU_IFF_DFU	188
8.23.1.13 DFU_IFF_IFACE	188
8.23.1.14 DFU_IFF_PATH	188
8.23.1.15 DFU_IFF_PRODUCT	188
8.23.1.16 DFU_IFF_VENDOR	188
8.23.1.17 DFU_STATUS_ERROR_ADDRESS	188
8.23.1.18 DFU_STATUS_ERROR_CHECK_ERASED	189
8.23.1.19 DFU_STATUS_ERROR_ERASE	189
8.23.1.20 DFU_STATUS_ERROR_FILE	189
8.23.1.21 DFU_STATUS_ERROR_FIRMWARE	189
8.23.1.22 DFU_STATUS_ERROR_NOTDONE	189
8.23.1.23 DFU_STATUS_ERROR_POR	189
8.23.1.24 DFU_STATUS_ERROR_PROG	189
8.23.1.25 DFU_STATUS_ERROR_STALLEDPKT	189
8.23.1.26 DFU_STATUS_ERROR_TARGET	190
8.23.1.27 DFU_STATUS_ERROR_UNKNOWN	190
8.23.1.28 DFU_STATUS_ERROR_USBR	190
8.23.1.29 DFU_STATUS_ERROR_VENDOR	190
8.23.1.30 DFU_STATUS_ERROR_VERIFY	190
8.23.1.31 DFU_STATUS_ERROR_WRITE	190
8.23.1.32 DFU_STATUS_OK	190
8.23.1.33 DFU_UPLOAD	190
8.23.1.34 STATE_APP_DETACH	191
8.23.1.35 STATE_APP_IDLE	191
8.23.1.36 STATE_DFU_DOWNLOAD_BUSY	191
8.23.1.37 STATE_DFU_DOWNLOAD_IDLE	191
8.23.1.38 STATE_DFU_DOWNLOAD_SYNC	191
8.23.1.39 STATE_DFU_ERROR	191

8.23.1.40 STATE_DFU_IDLE	191
8.23.1.41 STATE_DFU_MANIFEST	191
8.23.1.42 STATE_DFU_MANIFEST_SYNC	192
8.23.1.43 STATE_DFU_MANIFEST_WAIT_RESET	192
8.23.1.44 STATE_DFU_UPLOAD_IDLE	192
8.23.2 Enumeration Type Documentation	192
8.23.2.1 dfuse_command	192
8.23.3 Function Documentation	192
8.23.3.1 pageAddress()	192
8.23.3.2 pageNumber()	193
8.23.4 Variable Documentation	193
8.23.4.1 BFirmwareInfoEncrypt1	193
8.23.4.2 BFirmwareInfoMagic	193
8.24 Dfu.h File Reference	193
8.25 overview.dox File Reference	193
Index	195

Chapter 1

Main Page

Author

Dr Terry Barnaby

Version

1.0.0

Date

2020-02-09

1.1 Introduction

The Beam BMeasure-125i unit is a flexible and powerful IoT system for data capture, data logging and control in the laboratory, industrial and remote sensing arenas. It is based around an 8 channel, fully differential, synchronous sampling, 24 bit ADC that can sample at speeds up to 128 ksps. Multiple units can be connected together to provide more synchronously sampled channels.

This reference information describes the data types and functions provided by the host API library allowing programs to be written to control the operation of a BMeasure unit and acquire the data from it. The API operates over a number of different physical interfaces including: USB 2.0, Ethernet, Wifi and RS485.

In addition there is a software manual providing an overview of using this API which should be read first. This document is available at: <https://portal.beam.ltd.uk/files/products/bmeasure-125i/doc/BMeasure-lib.pdf>

1.2 Overview

The BMeasure API library, `bmeasure-lib`, is implemented in the C++ computer language. It has bindings layered on top of this for Python, with Matlab due to be supported soon. The API has an object orientated architecture. It has been designed as a general purpose API library for the Beam BMeasure-125i and future BMeasure products. Currently it has ports to Linux (Redhat7, Fedora29, Debian) and Microsoft Windows 7, 8 and 10.

The API provides the following functionality:

- Find BMeasure units on the USB bus or local Ethernet and Wifi networks.
- Connect to one or more BMeasure units.
- Fetch information and configure the BMeasure units.
- Start the BMeasure unit capturing and processing the sensor inputs.
- Capture the data from all of the analogue and digital channels from one or a combined set of BMeasure units running in sync.
- Access the data log files on the unit and download them to the host.
- Configure the AWG to produce waveforms or set voltages on the analogue output channels.
- Operate relays, read switches and other auxiliary operations.

The BMeasure API is implemented using the Beam BOAP (Beam Object Access protocol) communications system. It offers an `BMeasureUnit` API class to access an individual BMeasure unit in a relatively low level manner and an `BMeasureUnits` API class to access a set of BMeasure units synchronised together to operate as a single unit and with a queued data reception system..

The API supports threaded and non-threaded operation.

The referenve information provided describes the API from a C++ programming perspective. The Python and other language bindings are very similar the differences being noted under the particular language bindings section in the software manual..

1.3 API Usage

To use the API the core procedure is:

1. Either find the available BMeasure units using: `BMeasureApi::BMeasureUnit::findDevices()` or use a B↔Measure URL string..
2. Choose to use the simple single unit interface `BMeasureApi::BMeasureUnit` or the `BMeasureApi::BMeasureUnits` classes.
3. If using the simple single unit interface, connect to the unit using the `BMeasureApi::BMeasureUnit::connect()` function.
4. If using the multiple unit interface, add the units using the `BMeasureApi::BMeasureUnits::unitAdd()` function and connect using the `BMeasureApi::BMeasureUnits::unitsConnect()` function.
5. Use the interface to communicate to the unit.

See the examples below and the software manual for more details.

1.4 API Usage

There are some examples of client applications using the BMeasure API in the **examples** directory of the source code. Some simple client examples are listed below:

Simple example to access and read single sets of data samples in C++

```

/*****
 *      Example005-dataClient-single.cpp
 *      T.Barnaby,      BEAM Ltd,      2019-10-09
 *****/
#include <BMeasureUnit.h>
#include <unistd.h>
using namespace BMeasureApi;
// Function to read some data
BError test1(){
    BError          err;
    BList<BMeasureUnitDevice>  devices;
    BString         device;
    BMeasureUnit    bmeasure;
    Information     info;
    Configuration   config;
    MeasurementConfig mc;
    DataBlock       data;
    BUInt           c;

    printf("Start Processing Task\n");
    bmeasure.start();

    printf("Find BMeasure units\n");
    if(err = BMeasureUnit::findDevicesUsb(devices)){
        return err;
    }
    if(devices.number() == 0){
        return err.set(1, "No USB BMeasure units found\n");
    }
    device = devices[0].device;

    printf("Connect\n");
    if(err = bmeasure.connect(device))
        return err;

    //printf("Exit\n"); return err;
    printf("Get Info\n");
    if(err = bmeasure.getInformation(info))
        return err;

    printf("NumChannels: %d\n", info.numChannels);
    //printf("Exit\n"); return err;
    printf("Configure measurement\n");
    mc.measureMode = MeasureModeOneShot;
    mc.triggerMode = TriggerModeOff;
    mc.triggerConfig = TriggerConfigNone;
    mc.triggerChannel = 0;
    mc.triggerLevel = 0;
    mc.triggerDelay = 0;
    mc.sampleRate = 8000.0;
    mc.measurePeriod = 0;
    mc.numSamples0 = 1;
    mc.numSamples1 = 0;
    if(err = bmeasure.setMeasurementConfig(0, mc))
        return err;

    printf("Run single measurement\n");
    if(err = bmeasure.measure(DataTypeFloat32, data))
        return err;

    printf("DataBlock: from: %d numChannels: %d numSamples: %d\n", data.source, data.numChannels,
    data.numSamples);
    for(c = 0; c < data.numChannels; c++){
        printf("%f ", data.data[c]);
    }
    printf("\n");
    return err;
}
int main(){
    BError err;
    if(err = test1()){
        printf("Error: %d %s\n", err.getErrorNo(), err.str());
        return 1;
    }
    printf("Complete\n");
    return 0;
}

```

Simple example to access and read single sets of data samples in Python

```
#!/usr/bin/python3
import sys
import time
import getopt
from threading import Thread
from bmeasure import *
# Function to read some data
def test1():
    bmeasure = BMeasureUnit(True);
    print("Find BMeasure units");
    (err, devices) = BMeasureUnit.findDevicesUsb();
    if(err):
        return err;
    if(devices.number() == 0):
        return err.set(1, "No USB BMeasure units found\n");
    print("Found", len(devices));
    device = devices[0].device;
    print("Start Processing Task");
    bmeasure.start();
    print("Connect to BMeasure");
    err = bmeasure.connect(device);
    if(err):
        return err;
    print("Get Info");
    (err, info) = bmeasure.getInformation();
    if(err):
        return err;

    print("NumChannels: ", info.numChannels);
    print("Configure measurement");
    mc = MeasurementConfig();
    mc.measureMode = MeasureModeOneShot;
    mc.triggerMode = TriggerModeOff;
    mc.triggerConfig = TriggerConfigNone;
    mc.triggerChannel = 0;
    mc.triggerLevel = 0;
    mc.triggerDelay = 0;
    mc.sampleRate = 4000;
    mc.numSamples0 = 1;
    mc.numSamples1 = 0;
    mc.measurePeriod = 0;
    err = bmeasure.setMeasurementConfig(False, mc);
    if(err):
        return err;

    print("Run single measurement");
    (err, data) = bmeasure.measure();
    if(err):
        return err;
    print("DataBlock: from: %d numChannels: %d numSamples: %d" % (data.source, data.numChannels,
    data.numSamples));
    for c in range(0, data.numChannels):
        print("Chan:", c, data.data[c]);
    return err;
def main():
    err = test1();
    if(err):
        print("Error:", err.getErrorNo(), err.getString());
        return 1;
    print("Complete");

    return 0;
if __name__ == "__main__":
    main();
```

Simple example to show operating the relays in Python

```
#!/usr/bin/python3
import sys
import time
import getopt
from threading import Thread
from bmeasure import *
# Function to set the relays on/off
def test1():
    bmeasure = BMeasureUnit(True);
    print("Find BMeasure units");
    (err, devices) = BMeasureUnit.findDevicesUsb();
    if(err):
        return err;
    if(devices.number() == 0):
        return err.set(1, "No USB BMeasure units found\n");
    print("Found", len(devices));
    device = devices[0].device;
    print("Start Communications Task");
    bmeasure.start();
```

```
print("Connect");
err = bmeasure.connect(device);
if(err):
    return err;
print("Get Info");
(err, info) = bmeasure.getInformation();
if(err):
    return err;

print("NumChannels: ", info.numChannels);
# Toggle relay1
state = 0;
for i in range(0, 6):
    if(state):
        state = False;
    else:
        state = True;
    print("Set relay 0: %d" % (state));
    err = bmeasure.setRelay(0, state);
    if(err):
        return err;

    time.sleep(1);
return err;
def main():
    if(0):
        err = find();
        if(err):
            print("Error:", err.getErrorNo(), err.getString());
            return 1;

    err = test1();
    if(err):
        print("Error:", err.getErrorNo(), err.getString());
        return 1;
    print("Complete");

    return 0;
if __name__ == "__main__":
    main();
```


Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

[BMeasureApi](#) 15

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

BMeasureApi::AlarmConfig	47
BMeasureApi::AwgConfig	49
BComms [external]	
BMeasureApi::CommsNet	110
BMeasureApi::CommsSerial	113
BMeasureApi::CommsUsb	116
BFirmwareInfo	51
BMdns	53
BMdnsService	54
BMeasureApi::BMeasureUnitDevice	84
BMeasureApi::BMeasureUnitsDataBlock	99
BoapMc1Comms [external]	
BMeasureApi::BMeasure	55
BMeasureApi::BMeasureUnit	75
BMeasureApi::BMeasureUnit1	81
BMeasureApi::BoardConfig	101
BTask [external]	
BMeasureApi::BMeasureUnit	75
BMeasureApi::BMeasureUnits	85
BMeasureApi::CalibrateInfo	104
BMeasureApi::ChannelConfig	106
BMeasureApi::ConfigItem	119
BMeasureApi::Configuration	120
BMeasureApi::DataBlock	128
BMeasureApi::DataBlockProc	130
BMeasureApi::DataFile	133
BMeasureApi::DataProc	138
Dfu	139
DfuStatus	143
BMeasureApi::FileData	144
BMeasureApi::FileInfo	145
BMeasureApi::FilesysInfo	147
BMeasureApi::InfoBlock	148
BMeasureApi::Information	150
BMeasureApi::MeasurementConfig	155
BMeasureApi::NodeInfo	158
BMeasureApi::NodeStatus	160
BMeasureApi::Version	162

Chapter 4

Class Index

4.1 Class List

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

- BMeasureApi::AlarmConfig 47
- BMeasureApi::AwgConfig 49
- BFirmwareInfo 51
- BMdns 53
- BMdnsService 54
- BMeasureApi::BMeasure 55
- BMeasureApi::BMeasureUnit 75
- BMeasureApi::BMeasureUnit1 81
- BMeasureApi::BMeasureUnitDevice 84
- BMeasureApi::BMeasureUnits 85
- BMeasureApi::BMeasureUnitsDataBlock 99
- BMeasureApi::BoardConfig 101
- BMeasureApi::CalibrateInfo 104
- BMeasureApi::ChannelConfig 106
- BMeasureApi::CommsNet 110
- BMeasureApi::CommsSerial 113
- BMeasureApi::CommsUsb 116
- BMeasureApi::ConfigItem 119
- BMeasureApi::Configuration 120
- BMeasureApi::DataBlock 128
- BMeasureApi::DataBlockProc 130
- BMeasureApi::DataFile 133
- BMeasureApi::DataProc 138
- Dfu
 - The Dfu access class 139
- DfuStatus 143
- BMeasureApi::FileData 144
- BMeasureApi::FileInfo 145
- BMeasureApi::FilesysInfo 147
- BMeasureApi::InfoBlock 148
- BMeasureApi::Information 150
- BMeasureApi::MeasurementConfig 155
- BMeasureApi::NodeInfo 158
- BMeasureApi::NodeStatus 160
- BMeasureApi::Version 162

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

BMdns.cpp	165
BMdns.h	167
BMeasureB-1.cpp	167
BMeasureB.cpp	168
BMeasureB.h	168
BMeasureD.cpp	168
BMeasureD.h	171
BMeasureLib.cpp	175
BMeasureLib.h	176
BMeasureS.cpp	177
BMeasureUnit.cpp	178
BMeasureUnit.h	179
BMeasureUnits.cpp	179
BMeasureUnits.h	180
CommsNet.cpp	181
CommsNet.h	182
CommsSerial.cpp	182
CommsSerial.h	182
CommsUsb.cpp	182
CommsUsb.h	183
DataFile.cpp	183
DataFile.h	185
Dfu.cpp	185
Dfu.h	193

Chapter 6

Namespace Documentation

6.1 BMeasureApi Namespace Reference

Classes

- class [AlarmConfig](#)
- class [AwgConfig](#)
- class [BMeasure](#)
- class [BMeasureUnit](#)
- class [BMeasureUnit1](#)
- class [BMeasureUnitDevice](#)
- class [BMeasureUnits](#)
- class [BMeasureUnitsDataBlock](#)
- class [BoardConfig](#)
- class [CalibrateInfo](#)
- class [ChannelConfig](#)
- class [CommsNet](#)
- class [CommsSerial](#)
- class [CommsUsb](#)
- class [ConfigItem](#)
- class [Configuration](#)
- class [DataBlock](#)
- class [DataBlockProc](#)
- class [DataFile](#)
- class [DataProc](#)
- class [FileData](#)
- class [FileInfo](#)
- class [FilesysInfo](#)
- class [InfoBlock](#)
- class [Information](#)
- class [MeasurementConfig](#)
- class [NodeInfo](#)
- class [NodeStatus](#)
- class [Version](#)

Typedefs

- typedef **BArray**< [ChannelConfig](#) > [ChannelConfigs](#)

Enumerations

- enum `ErrorNum` { `ErrorNumSystem` = 64, `ErrorNumDataOverrun` = 65, `ErrorNumToFast` = 66 }
- enum `NodeType` { `NodeTypeNone` = 0, `NodeTypeBMeasure1` = 1 }
- enum `SecurityMode` { `SecurityModeBasic`, `SecurityModeConfig`, `SecurityModeFull` }
- enum `Status` {
`StatusNone` = 0x00, `StatusError` = 0x01, `StatusWarning` = 0x02, `StatusRun` = 0x04,
`StatusTriggerWait` = 0x08, `StatusEnd0` = 0x10, `StatusEnd1` = 0x20, `StatusDataOverrun` = 0x40,
`StatusFpgaOverrun` = 0x80, `StatusAlarm` = 0x0100 }
- enum `Mode` {
`ModelIdle` = 0, `ModeRun` = 1, `ModeRunProgram` = 2, `ModelInternal` = 3,
`ModeSleep` = 4, `ModeDemo1` = 5 }
- enum `BlockTypes` { `BlockTypeInfo` = 0x424E4531, `BlockTypeData` = 0x424E4532 }
- enum `ChannelType` {
`ChannelTypeNone` = 0, `ChannelTypeAnalogueIn` = 1, `ChannelTypeAnalogueOut` = 2, `ChannelTypeDigitalIn` = 3,
`ChannelTypeDigitalOut` = 4 }
- enum `SampleType` {
`SampleTypeNone` = 0, `SampleTypeBool` = 1, `SampleTypeInt8` = 2, `SampleTypeInt16` = 3,
`SampleTypeInt32` = 4, `SampleTypeFloat32` = 5, `SampleTypeFloat64` = 6 }
- enum `SyncMode` { `SyncModeOff` = 0, `SyncModeMaster` = 1, `SyncModeSlave` = 2 }
- enum `MeasureMode` { `MeasureModeOff` = 0, `MeasureModeOneShot` = 1, `MeasureModeRepeat` = 2, `MeasureModeContinuous` = 3 }
- enum `MeasureOption` { `MeasureOptionNone` = 0, `MeasureOptionProcess` = 0x01 }
- enum `TriggerMode` { `TriggerModeOff` = 0, `TriggerModePositive` = 1, `TriggerModeNegative` = 2 }
- enum `TriggerConfig` { `TriggerConfigNone` = 0 }
- enum `DigitalMode` {
`DigitalModeInput` = 0, `DigitalModeOutput` = 1, `DigitalModeInOut` = 2, `DigitalModeSyncMaster` = 3,
`DigitalModeSyncSlave` = 4 }
- enum `AwgMode` {
`AwgModeNone`, `AwgModeDc`, `AwgModeSine`, `AwgModeSquare`,
`AwgModeTriangle`, `AwgModeNoise`, `AwgModeTrackRms`, `AwgModeTrackMean`,
`AwgModeArbitrary` }
- enum `AwgOutput` { `AwgOutputNone`, `AwgOutputAO0`, `AwgOutputAO1`, `AwgOutputAO01` }
- enum `FileType` { `FileTypeNone`, `FileTypeFile`, `FileTypeDir` }
- enum `FilesysDeleteType` { `FilesysDeleteTypeNone`, `FilesysDeleteTypeData`, `FilesysDeleteTypeFormat` }
- enum `LogData` { `LogDataOff`, `LogDataRaw` = 0x01, `LogDataProcessed` = 0x02 }
- enum `LogDataMode` { `LogDataModeNormal`, `LogDataModeDeleteOld` }
- enum `Data Type` { `Data TypeFloat32`, `Data Type125i`, `Data TypeProc` }
- enum `DataSend` { `DataSendOff`, `DataSendStatus` = 0x01, `DataSendRaw` = 0x02, `DataSendProcessed` = 0x04 }
- enum `CalibrateStage` {
`CalibrateStageNone` = 0, `CalibrateStageClear` = 1, `CalibrateStageSettle` = 2, `CalibrateStageAdcOffsets` = 3,
`CalibrateStageDacOffsets` = 4, `CalibrateStageDacScaling0` = 5, `CalibrateStageDacScaling1` = 6,
`CalibrateStageAdcScaling` = 7,
`CalibrateStageAttenScaling` = 8, `CalibrationStageFiveVolts` = 9 }
- enum `MessageSource` {
`MessageSourceGeneral` = 0, `MessageSourceDebug` = 1, `MessageSourceTest` = 2, `MessageSourceWifi` = 3,
`MessageSourceWifiTest` = 4 }
- enum `NetworkMode` { `NetworkModeOff` = 0, `NetworkModeDhcp` = 1, `NetworkModeManual` = 2 }
- enum `WifiMode` { `WifiModeOff`, `WifiModeClient`, `WifiModeAp` }
- enum `AlarmMode` { `AlarmModeOff`, `AlarmModeHigh`, `AlarmModeLow`, `AlarmModeRange` }
- enum `AlarmOutput` {
`AlarmOutputOff`, `AlarmOutputDioHigh`, `AlarmOutputDioLow`, `AlarmOutputRelayOn`,
`AlarmOutputRelayOff` }
- enum `EventMode` { `EventModeOff`, `EventModeAlarm`, `EventModeSecond` }

- enum [Rs485Mode](#) { [Rs485ModeOff](#), [Rs485ModeBoap](#) }
- enum [BMeasFileType](#) { [BMeasFileTypeBlock512](#), [BMeasFileTypeStream](#) }
- enum [TdsDataType](#) {
[TdsTypeVoid](#), [TdsType8](#), [TdsType16](#), [TdsType32](#),
[TdsType64](#), [TdsTypeU8](#), [TdsTypeU16](#), [TdsTypeU32](#),
[TdsTypeU64](#), [TdsTypeSingleFloat](#), [TdsTypeDoubleFloat](#), [TdsTypeExtendedFloat](#),
[TdsTypeSingleFloatWithUnit](#) =0x19, [TdsTypeDoubleFloatWithUnit](#), [TdsTypeExtendedFloatWithUnit](#),
[TdsTypeString](#) =0x20,
[TdsTypeBoolean](#) =0x21, [TdsTypeTimeStamp](#) =0x44, [TdsTypeFixedPoint](#) =0x4F, [TdsTypeComplexSingleFloat](#)
=0x08000c,
[TdsTypeComplexDoubleFloat](#) =0x10000d, [TdsTypeDAQmxRawData](#) =0xFFFFFFFF }

Functions

- [BString](#) [toBString](#) ([ErrorNum](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [ErrorNum](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [ErrorNum](#) v)
- [BString](#) [toBString](#) ([NodeType](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [NodeType](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [NodeType](#) v)
- [BString](#) [toBString](#) ([SecurityMode](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [SecurityMode](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [SecurityMode](#) v)
- [BString](#) [toBString](#) ([Status](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [Status](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [Status](#) v)
- [BString](#) [toBString](#) ([Mode](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [Mode](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [Mode](#) v)
- [BString](#) [toBString](#) ([BlockTypes](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [BlockTypes](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [BlockTypes](#) v)
- [BString](#) [toBString](#) ([ChannelType](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [ChannelType](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [ChannelType](#) v)
- [BString](#) [toBString](#) ([SampleType](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [SampleType](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [SampleType](#) v)
- [BString](#) [toBString](#) ([SyncMode](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [SyncMode](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [SyncMode](#) v)
- [BString](#) [toBString](#) ([MeasureMode](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [MeasureMode](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [MeasureMode](#) v)
- [BString](#) [toBString](#) ([MeasureOption](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [MeasureOption](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [MeasureOption](#) v)
- [BString](#) [toBString](#) ([TriggerMode](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [TriggerMode](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [TriggerMode](#) v)
- [BString](#) [toBString](#) ([TriggerConfig](#) v)
- [BError](#) [fromBString](#) ([BString](#) str, [TriggerConfig](#) &v)
- [BString](#) [toBStringJson](#) ([BString](#) n, [TriggerConfig](#) v)
- [BString](#) [toBString](#) ([DigitalMode](#) v)

- **BError** fromBString (**BString** str, [DigitalMode](#) &v)
- **BString** toBStringJson (**BString** n, [DigitalMode](#) v)
- **BString** toBString ([AwgMode](#) v)
- **BError** fromBString (**BString** str, [AwgMode](#) &v)
- **BString** toBStringJson (**BString** n, [AwgMode](#) v)
- **BString** toBString ([AwgOutput](#) v)
- **BError** fromBString (**BString** str, [AwgOutput](#) &v)
- **BString** toBStringJson (**BString** n, [AwgOutput](#) v)
- **BString** toBString ([FileType](#) v)
- **BError** fromBString (**BString** str, [FileType](#) &v)
- **BString** toBStringJson (**BString** n, [FileType](#) v)
- **BString** toBString ([FilesysDeleteType](#) v)
- **BError** fromBString (**BString** str, [FilesysDeleteType](#) &v)
- **BString** toBStringJson (**BString** n, [FilesysDeleteType](#) v)
- **BString** toBString ([LogData](#) v)
- **BError** fromBString (**BString** str, [LogData](#) &v)
- **BString** toBStringJson (**BString** n, [LogData](#) v)
- **BString** toBString ([LogDataMode](#) v)
- **BError** fromBString (**BString** str, [LogDataMode](#) &v)
- **BString** toBStringJson (**BString** n, [LogDataMode](#) v)
- **BString** toBString ([DataType](#) v)
- **BError** fromBString (**BString** str, [DataType](#) &v)
- **BString** toBStringJson (**BString** n, [DataType](#) v)
- **BString** toBString ([DataSend](#) v)
- **BError** fromBString (**BString** str, [DataSend](#) &v)
- **BString** toBStringJson (**BString** n, [DataSend](#) v)
- **BString** toBString ([CalibrateStage](#) v)
- **BError** fromBString (**BString** str, [CalibrateStage](#) &v)
- **BString** toBStringJson (**BString** n, [CalibrateStage](#) v)
- **BString** toBString ([MessageSource](#) v)
- **BError** fromBString (**BString** str, [MessageSource](#) &v)
- **BString** toBStringJson (**BString** n, [MessageSource](#) v)
- **BString** toBString ([NetworkMode](#) v)
- **BError** fromBString (**BString** str, [NetworkMode](#) &v)
- **BString** toBStringJson (**BString** n, [NetworkMode](#) v)
- **BString** toBString ([WifiMode](#) v)
- **BError** fromBString (**BString** str, [WifiMode](#) &v)
- **BString** toBStringJson (**BString** n, [WifiMode](#) v)
- **BString** toBString ([AlarmMode](#) v)
- **BError** fromBString (**BString** str, [AlarmMode](#) &v)
- **BString** toBStringJson (**BString** n, [AlarmMode](#) v)
- **BString** toBString ([AlarmOutput](#) v)
- **BError** fromBString (**BString** str, [AlarmOutput](#) &v)
- **BString** toBStringJson (**BString** n, [AlarmOutput](#) v)
- **BString** toBString ([EventMode](#) v)
- **BError** fromBString (**BString** str, [EventMode](#) &v)
- **BString** toBStringJson (**BString** n, [EventMode](#) v)
- **BString** toBString ([Rs485Mode](#) v)
- **BError** fromBString (**BString** str, [Rs485Mode](#) &v)
- **BString** toBStringJson (**BString** n, [Rs485Mode](#) v)
- **BString** toBString ([BMeasFileType](#) v)
- **BError** fromBString (**BString** str, [BMeasFileType](#) &v)
- **BString** toBStringJson (**BString** n, [BMeasFileType](#) v)
- const char * [channelTypeString](#) ([ChannelType](#) type)
- const char * [sampleTypeString](#) ([SampleType](#) type)

- **BFloat32** toFloat (**BUInt32** v)
- static int unitSort (BMeasureUnit1 *&u1, BMeasureUnit1 *&u2)
- static **BUInt32** roundDown512 (**BUInt32** size)
- const **BUInt32** TocMetaData (1<< 1)
- const **BUInt32** TocNewObjList (1<< 2)
- const **BUInt32** TocRawData (1<< 3)
- const **BUInt32** TocInterleavedData (1<< 5)
- const **BUInt32** TocBigEndian (1<< 6)
- const **BUInt32** TocDaqRawData (1<< 7)
- **BUInt32** round512 (**BUInt32** s)

Variables

- const **BUInt32** apiVersion = 0

6.1.1 Typedef Documentation

6.1.1.1 ChannelConfigs

```
typedef BArray<ChannelConfig> BMeasureApi::ChannelConfigs
```

6.1.2 Enumeration Type Documentation

6.1.2.1 AlarmMode

```
enum BMeasureApi::AlarmMode
```

Enumerator

AlarmModeOff	
AlarmModeHigh	
AlarmModeLow	
AlarmModeRange	

6.1.2.2 AlarmOutput

```
enum BMeasureApi::AlarmOutput
```

Enumerator

AlarmOutputOff	
AlarmOutputDioHigh	
AlarmOutputDioLow	
AlarmOutputRelayOn	
AlarmOutputRelayOff	

6.1.2.3 AwgMode

enum `BMeasureApi::AwgMode`

Enumerator

AwgModeNone	
AwgModeDc	
AwgModeSine	
AwgModeSquare	
AwgModeTriangle	
AwgModeNoise	
AwgModeTrackRms	
AwgModeTrackMean	
AwgModeArbitrary	

6.1.2.4 AwgOutput

enum `BMeasureApi::AwgOutput`

Enumerator

AwgOutputNone	
AwgOutputAO0	
AwgOutputAO1	
AwgOutputAO01	

6.1.2.5 BlockTypes

enum `BMeasureApi::BlockTypes`

Enumerator

BlockTypeInfo	
BlockTypeData	

6.1.2.6 BMeasFileType

```
enum BMeasureApi::BMeasFileType
```

Enumerator

BMeasFileTypeBlock512	
BMeasFileTypeStream	

6.1.2.7 CalibrateStage

```
enum BMeasureApi::CalibrateStage
```

Enumerator

CalibrateStageNone	
CalibrateStageClear	
CalibrateStageSettle	
CalibrateStageAdcOffsets	
CalibrateStageDacOffsets	
CalibrateStageDacScaling0	
CalibrateStageDacScaling1	
CalibrateStageAdcScaling	
CalibrateStageAttenScaling	
CalibrationStageFiveVolts	

6.1.2.8 ChannelType

```
enum BMeasureApi::ChannelType
```

Enumerator

ChannelTypeNone	
ChannelTypeAnalogueIn	
ChannelTypeAnalogueOut	
ChannelTypeDigitalIn	
ChannelTypeDigitalOut	

6.1.2.9 DataSend

enum `BMeasureApi::DataSend`

Enumerator

DataSendOff	
DataSendStatus	
DataSendRaw	
DataSendProcessed	

6.1.2.10 DataType

enum `BMeasureApi::DataType`

Enumerator

DataTypeFloat32	
DataType125i	
DataTypeProc	

6.1.2.11 DigitalMode

enum `BMeasureApi::DigitalMode`

Enumerator

DigitalModeInput	
DigitalModeOutput	
DigitalModeInOut	
DigitalModeSyncMaster	
DigitalModeSyncSlave	

6.1.2.12 ErrorNum

enum `BMeasureApi::ErrorNum`

Enumerator

ErrorNumSystem	
ErrorNumDataOverrun	
ErrorNumToFast	

6.1.2.13 EventMode

```
enum BMeasureApi::EventMode
```

Enumerator

EventModeOff	
EventModeAlarm	
EventModeSecond	

6.1.2.14 FilesysDeleteType

```
enum BMeasureApi::FilesysDeleteType
```

Enumerator

FilesysDeleteTypeNone	
FilesysDeleteTypeData	
FilesysDeleteTypeFormat	

6.1.2.15 FileType

```
enum BMeasureApi::FileType
```

Enumerator

FileTypeNone	
FileTypeFile	
FileTypeDir	

6.1.2.16 LogData

enum `BMeasureApi::LogData`

Enumerator

LogDataOff	
LogDataRaw	
LogDataProcessed	

6.1.2.17 LogDataMode

enum `BMeasureApi::LogDataMode`

Enumerator

LogDataModeNormal	
LogDataModeDeleteOld	

6.1.2.18 MeasureMode

enum `BMeasureApi::MeasureMode`

Enumerator

MeasureModeOff	
MeasureModeOneShot	
MeasureModeRepeat	
MeasureModeContinuous	

6.1.2.19 MeasureOption

enum `BMeasureApi::MeasureOption`

Enumerator

MeasureOptionNone	
MeasureOptionProcess	

6.1.2.20 MessageSource

enum [BMeasureApi::MessageSource](#)

Enumerator

MessageSourceGeneral	
MessageSourceDebug	
MessageSourceTest	
MessageSourceWifi	
MessageSourceWifiTest	

6.1.2.21 Mode

enum [BMeasureApi::Mode](#)

Enumerator

ModeIdle	
ModeRun	
ModeRunProgram	
ModeInternal	
ModeSleep	
ModeDemo1	

6.1.2.22 NetworkMode

enum [BMeasureApi::NetworkMode](#)

Enumerator

NetworkModeOff	
NetworkModeDhcp	
NetworkModeManual	

6.1.2.23 NodeType

enum [BMeasureApi::NodeType](#)

Enumerator

NodeTypeNone	
NodeTypeBMeasure1	

6.1.2.24 Rs485Mode

enum `BMeasureApi::Rs485Mode`

Enumerator

Rs485ModeOff	
Rs485ModeBoap	

6.1.2.25 SampleType

enum `BMeasureApi::SampleType`

Enumerator

SampleTypeNone	
SampleTypeBool	
SampleTypeInt8	
SampleTypeInt16	
SampleTypeInt32	
SampleTypeFloat32	
SampleTypeFloat64	

6.1.2.26 SecurityMode

enum `BMeasureApi::SecurityMode`

Enumerator

SecurityModeBasic	
SecurityModeConfig	
SecurityModeFull	

6.1.2.27 Status

enum `BMeasureApi::Status`

Enumerator

StatusNone	
StatusError	
StatusWarning	
StatusRun	
StatusTriggerWait	
StatusEnd0	
StatusEnd1	
StatusDataOverrun	
StatusFpgaOverrun	
StatusAlarm	

6.1.2.28 SyncMode

enum `BMeasureApi::SyncMode`

Enumerator

SyncModeOff	
SyncModeMaster	
SyncModeSlave	

6.1.2.29 TdsDataType

enum `BMeasureApi::TdsDataType`

Enumerator

TdsTypeVoid	
TdsTypeI8	
TdsTypeI16	
TdsTypeI32	
TdsTypeI64	
TdsTypeU8	
TdsTypeU16	
TdsTypeU32	
TdsTypeU64	
TdsTypeSingleFloat	
TdsTypeDoubleFloat	

Enumerator

TdsTypeExtendedFloat	
TdsTypeSingleFloatWithUnit	
TdsTypeDoubleFloatWithUnit	
TdsTypeExtendedFloatWithUnit	
TdsTypeString	
TdsTypeBoolean	
TdsTypeTimeStamp	
TdsTypeFixedPoint	
TdsTypeComplexSingleFloat	
TdsTypeComplexDoubleFloat	
TdsTypeDAQmxRawData	

6.1.2.30 TriggerConfig

```
enum BMeasureApi::TriggerConfig
```

Enumerator

TriggerConfigNone	
-------------------	--

6.1.2.31 TriggerMode

```
enum BMeasureApi::TriggerMode
```

Enumerator

TriggerModeOff	
TriggerModePositive	
TriggerModeNegative	

6.1.2.32 WifiMode

```
enum BMeasureApi::WifiMode
```

Enumerator

WifiModeOff	
WifiModeClient	
WifiModeAp	

6.1.3 Function Documentation

6.1.3.1 channelTypeString()

```
const char * BMeasureApi::channelTypeString (
    ChannelType type )
```

6.1.3.2 fromBString() [1/31]

```
BError BMeasureApi::fromBString (
    BString str,
    ErrorNum & v )
```

6.1.3.3 fromBString() [2/31]

```
BError BMeasureApi::fromBString (
    BString str,
    NodeType & v )
```

6.1.3.4 fromBString() [3/31]

```
BError BMeasureApi::fromBString (
    BString str,
    SecurityMode & v )
```

6.1.3.5 fromBString() [4/31]

```
BError BMeasureApi::fromBString (
    BString str,
    Status & v )
```

6.1.3.6 fromBString() [5/31]

```
BError BMeasureApi::fromBString (
    BString str,
    Mode & v )
```

6.1.3.7 fromBString() [6/31]

```
BError BMeasureApi::fromBString (
    BString str,
    BlockTypes & v )
```

6.1.3.8 fromBString() [7/31]

```
BError BMeasureApi::fromBString (
    BString str,
    ChannelType & v )
```

6.1.3.9 fromBString() [8/31]

```
BError BMeasureApi::fromBString (
    BString str,
    SampleType & v )
```

6.1.3.10 fromBString() [9/31]

```
BError BMeasureApi::fromBString (
    BString str,
    SyncMode & v )
```

6.1.3.11 fromBString() [10/31]

```
BError BMeasureApi::fromBString (
    BString str,
    MeasureMode & v )
```

6.1.3.12 fromBString() [11/31]

```
BError BMeasureApi::fromBString (
    BString str,
    MeasureOption & v )
```


6.1.3.13 fromBString() [12/31]

```
BError BMeasureApi::fromBString (
    BString str,
    TriggerMode & v )
```

6.1.3.14 fromBString() [13/31]

```
BError BMeasureApi::fromBString (
    BString str,
    TriggerConfig & v )
```

6.1.3.15 fromBString() [14/31]

```
BError BMeasureApi::fromBString (
    BString str,
    DigitalMode & v )
```

6.1.3.16 fromBString() [15/31]

```
BError BMeasureApi::fromBString (
    BString str,
    AwgMode & v )
```

6.1.3.17 fromBString() [16/31]

```
BError BMeasureApi::fromBString (
    BString str,
    AwgOutput & v )
```

6.1.3.18 fromBString() [17/31]

```
BError BMeasureApi::fromBString (
    BString str,
    FileType & v )
```

6.1.3.19 fromBString() [18/31]

```
BError BMeasureApi::fromBString (
    BString str,
    FilesysDeleteType & v )
```

6.1.3.20 fromBString() [19/31]

```
BError BMeasureApi::fromBString (
    BString str,
    LogData & v )
```

6.1.3.21 fromBString() [20/31]

```
BError BMeasureApi::fromBString (
    BString str,
    LogDataMode & v )
```

6.1.3.22 fromBString() [21/31]

```
BError BMeasureApi::fromBString (
    BString str,
    DataType & v )
```

6.1.3.23 fromBString() [22/31]

```
BError BMeasureApi::fromBString (
    BString str,
    DataSend & v )
```

6.1.3.24 fromBString() [23/31]

```
BError BMeasureApi::fromBString (
    BString str,
    CalibrateStage & v )
```

6.1.3.25 fromBString() [24/31]

```
BError BMeasureApi::fromBString (
    BString str,
    MessageSource & v )
```

6.1.3.26 fromBString() [25/31]

```
BError BMeasureApi::fromBString (
    BString str,
    NetworkMode & v )
```

6.1.3.27 fromBString() [26/31]

```
BError BMeasureApi::fromBString (
    BString str,
    WifiMode & v )
```

6.1.3.28 fromBString() [27/31]

```
BError BMeasureApi::fromBString (
    BString str,
    AlarmMode & v )
```

6.1.3.29 fromBString() [28/31]

```
BError BMeasureApi::fromBString (
    BString str,
    AlarmOutput & v )
```

6.1.3.30 fromBString() [29/31]

```
BError BMeasureApi::fromBString (
    BString str,
    EventMode & v )
```

6.1.3.31 fromBString() [30/31]

```
BError BMeasureApi::fromBString (
    BString str,
    Rs485Mode & v )
```

6.1.3.32 fromBString() [31/31]

```
BError BMeasureApi::fromBString (
    BString str,
    BMeasFileType & v )
```

6.1.3.33 round512()

```
BUInt32 BMeasureApi::round512 (
    BUInt32 s )
```

6.1.3.34 roundDown512()

```
static BUInt32 BMeasureApi::roundDown512 (
    BUInt32 size ) [static]
```

6.1.3.35 sampleTypeString()

```
const char * BMeasureApi::sampleTypeString (
    SampleType type )
```

6.1.3.36 toBString() [1/31]

```
BString BMeasureApi::toBString (
    ErrorNum v )
```

6.1.3.37 toBString() [2/31]

```
BString BMeasureApi::toBString (
    NodeType v )
```

6.1.3.38 toBString() [3/31]

```
BString BMeasureApi::toBString (
    SecurityMode v )
```

6.1.3.39 toBString() [4/31]

```
BString BMeasureApi::toBString (
    Status v )
```

6.1.3.40 toBString() [5/31]

```
BString BMeasureApi::toBString (
    Mode v )
```

6.1.3.41 toBString() [6/31]

```
BString BMeasureApi::toBString (
    BlockTypes v )
```

6.1.3.42 toBString() [7/31]

```
BString BMeasureApi::toBString (
    ChannelType v )
```

6.1.3.43 toBString() [8/31]

```
BString BMeasureApi::toBString (
    SampleType v )
```

6.1.3.44 toBString() [9/31]

```
BString BMeasureApi::toBString (
    SyncMode v )
```

6.1.3.45 toBString() [10/31]

```
BString BMeasureApi::toBString (
    MeasureMode v )
```

6.1.3.46 toBString() [11/31]

```
BString BMeasureApi::toBString (
    MeasureOption v )
```

6.1.3.47 toBString() [12/31]

```
BString BMeasureApi::toBString (
    TriggerMode v )
```

6.1.3.48 toBString() [13/31]

```
BString BMeasureApi::toBString (
    TriggerConfig v )
```

6.1.3.49 toBString() [14/31]

```
BString BMeasureApi::toBString (
    DigitalMode v )
```

6.1.3.50 toBString() [15/31]

```
BString BMeasureApi::toBString (
    AwgMode v )
```

6.1.3.51 toBString() [16/31]

```
BString BMeasureApi::toBString (
    AwgOutput v )
```

6.1.3.52 toBString() [17/31]

```
BString BMeasureApi::toBString (
    FileType v )
```

6.1.3.53 toBString() [18/31]

```
BString BMeasureApi::toBString (
    FilesysDeleteType v )
```

6.1.3.54 toBString() [19/31]

```
BString BMeasureApi::toBString (
    LogData v )
```

6.1.3.55 toBString() [20/31]

```
BString BMeasureApi::toBString (
    LogDataMode v )
```

6.1.3.56 toBString() [21/31]

```
BString BMeasureApi::toBString (
    DataType v )
```

6.1.3.57 toBString() [22/31]

```
BString BMeasureApi::toBString (
    DataSend v )
```

6.1.3.58 toBString() [23/31]

```
BString BMeasureApi::toBString (
    CalibrateStage v )
```

6.1.3.59 toBString() [24/31]

```
BString BMeasureApi::toBString (
    MessageSource v )
```

6.1.3.60 toBString() [25/31]

```
BString BMeasureApi::toBString (
    NetworkMode v )
```

6.1.3.61 toBString() [26/31]

```
BString BMeasureApi::toBString (
    WifiMode v )
```

6.1.3.62 toBString() [27/31]

```
BString BMeasureApi::toBString (
    AlarmMode v )
```

6.1.3.63 toBString() [28/31]

```
BString BMeasureApi::toBString (
    AlarmOutput v )
```

6.1.3.64 toBString() [29/31]

```
BString BMeasureApi::toBString (
    EventMode v )
```

6.1.3.65 toBString() [30/31]

```
BString BMeasureApi::toBString (
    Rs485Mode v )
```


6.1.3.66 toBString() [31/31]

```
BString BMeasureApi::toBString (
    BMeasFileType v )
```

6.1.3.67 toBStringJson() [1/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    ErrorNum v )
```

6.1.3.68 toBStringJson() [2/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    NodeType v )
```

6.1.3.69 toBStringJson() [3/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    SecurityMode v )
```

6.1.3.70 toBStringJson() [4/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    Status v )
```

6.1.3.71 toBStringJson() [5/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    Mode v )
```

6.1.3.72 toBStringJson() [6/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    BlockTypes v )
```

6.1.3.73 toBStringJson() [7/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    ChannelType v )
```

6.1.3.74 toBStringJson() [8/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    SampleType v )
```

6.1.3.75 toBStringJson() [9/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    SyncMode v )
```

6.1.3.76 toBStringJson() [10/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    MeasureMode v )
```

6.1.3.77 toBStringJson() [11/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    MeasureOption v )
```

6.1.3.78 toBStringJson() [12/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    TriggerMode v )
```

6.1.3.79 toBStringJson() [13/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    TriggerConfig v )
```

6.1.3.80 toBStringJson() [14/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    DigitalMode v )
```

6.1.3.81 toBStringJson() [15/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    AwgMode v )
```

6.1.3.82 toBStringJson() [16/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    AwgOutput v )
```

6.1.3.83 toBStringJson() [17/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    FileType v )
```

6.1.3.84 toBStringJson() [18/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    FilesysDeleteType v )
```

6.1.3.85 toBStringJson() [19/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    LogData v )
```

6.1.3.86 toBStringJson() [20/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    LogDataMode v )
```

6.1.3.87 toBStringJson() [21/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    DataType v )
```

6.1.3.88 toBStringJson() [22/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    DataSend v )
```

6.1.3.89 toBStringJson() [23/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    CalibrateStage v )
```

6.1.3.90 toBStringJson() [24/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    MessageSource v )
```

6.1.3.91 toBStringJson() [25/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    NetworkMode v )
```

6.1.3.92 toBStringJson() [26/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    WifiMode v )
```

6.1.3.93 toBStringJson() [27/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    AlarmMode v )
```

6.1.3.94 toBStringJson() [28/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    AlarmOutput v )
```

6.1.3.95 toBStringJson() [29/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    EventMode v )
```

6.1.3.96 toBStringJson() [30/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    Rs485Mode v )
```

6.1.3.97 toBStringJson() [31/31]

```
BString BMeasureApi::toBStringJson (
    BString n,
    BMeasFileType v )
```

6.1.3.98 TocBigEndian()

```
const BUInt32 BMeasureApi::TocBigEndian (
    1<< 6 )
```

6.1.3.99 TocDaqRawData()

```
const BUInt32 BMeasureApi::TocDaqRawData (
    1<< 7 )
```

6.1.3.100 TocInterleavedData()

```
const BUInt32 BMeasureApi::TocInterleavedData (
    1<< 5 )
```

6.1.3.101 TocMetaData()

```
const BUInt32 BMeasureApi::TocMetaData (
    1<< 1 )
```

6.1.3.102 TocNewObjList()

```
const BUInt32 BMeasureApi::TocNewObjList (
    1<< 2 )
```

6.1.3.103 TocRawData()

```
const BUInt32 BMeasureApi::TocRawData (
    1 << 3 )
```

6.1.3.104 toFloat()

```
BFloat32 BMeasureApi::toFloat (
    BUInt32 v ) [inline]
```

6.1.3.105 unitSort()

```
static int BMeasureApi::unitSort (
    BMeasureUnit1 *u1,
    BMeasureUnit1 *u2 ) [static]
```

6.1.4 Variable Documentation

6.1.4.1 apiVersion

```
const BUInt32 BMeasureApi::apiVersion = 0
```


Chapter 7

Class Documentation

7.1 BMeasureApi::AlarmConfig Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- [AlarmMode mode](#)
Alarm mode.
- [AlarmOutput output](#)
Alarm output.
- **BUint8** [outputChannel](#)
Alarm output channel.
- **BUint8** [spare](#)
- **BFloat32** [levelHigh](#)
Alarm level high.
- **BFloat32** [levelLow](#)
Alarm level low.

7.1.1 Member Function Documentation

7.1.1.1 [getMembers\(\)](#)

```
const BObjMember * BMeasureApi::AlarmConfig::getMembers ( ) [static]
```

7.1.2 Member Data Documentation

7.1.2.1 levelHigh

BFloat32 BMeasureApi::AlarmConfig::levelHigh

Alarm level high.

7.1.2.2 levelLow

BFloat32 BMeasureApi::AlarmConfig::levelLow

Alarm level low.

7.1.2.3 mode

[AlarmMode](#) BMeasureApi::AlarmConfig::mode

Alarm mode.

7.1.2.4 output

[AlarmOutput](#) BMeasureApi::AlarmConfig::output

Alarm output.

7.1.2.5 outputChannel

BUInt8 BMeasureApi::AlarmConfig::outputChannel

Alarm output channel.

7.1.2.6 spare

```
BUInt8 BMeasureApi::AlarmConfig::spare
```

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.2 BMeasureApi::AwgConfig Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- [AwgMode](#) *mode*
The mode including waveform type.
- [AwgOutput](#) *output*
The output channels.
- **BUInt8** [trackChannel](#)
Input channel to track.
- **BUInt8** [spare](#)
- **BFloat32** [frequency](#)
The frequency.
- **BFloat32** [amplitude](#)
The peak amplitude in Volts.
- **BFloat32** [offset](#)
The DC offset in volts.
- **BFloat32** [duty](#)
The Duty cycle in %.
- **BUInt32** [numSamples](#)
The number of samples when using arbitrary waveforms.

7.2.1 Member Function Documentation

7.2.1.1 [getMembers\(\)](#)

```
const BObjMember * BMeasureApi::AwgConfig::getMembers ( ) [static]
```

7.2.2 Member Data Documentation

7.2.2.1 amplitude

BFloat32 BMeasureApi::AwgConfig::amplitude

The peak amplitude in Volts.

7.2.2.2 duty

BFloat32 BMeasureApi::AwgConfig::duty

The Duty cycle in %.

7.2.2.3 frequency

BFloat32 BMeasureApi::AwgConfig::frequency

The frequency.

7.2.2.4 mode

[AwgMode](#) BMeasureApi::AwgConfig::mode

The mode including waveform type.

7.2.2.5 numSamples

BUInt32 BMeasureApi::AwgConfig::numSamples

The number of samples when using arbitrary waveforms.

7.2.2.6 offset

BFloat32 BMeasureApi::AwgConfig::offset

The DC offset in volts.

7.2.2.7 output

[AwgOutput](#) BMeasureApi::AwgConfig::output

The output channels.

7.2.2.8 spare

BUInt8 BMeasureApi::AwgConfig::spare

7.2.2.9 trackChannel

BUInt8 BMeasureApi::AwgConfig::trackChannel

Input channel to track.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.3 BFirmwareInfo Struct Reference

Public Attributes

- **BUInt32** [magic](#)
- **BUInt32** [length](#)
- **BUInt32** [checksum](#)
- **BUInt8** [type](#)
- **BUInt8** [ver0](#)
- **BUInt8** [ver1](#)
- **BUInt8** [ver2](#)

7.3.1 Member Data Documentation

7.3.1.1 checksum

BUInt32 BFirmwareInfo::checksum

7.3.1.2 length

BUInt32 BFirmwareInfo::length

7.3.1.3 magic

BUInt32 BFirmwareInfo::magic

7.3.1.4 type

BUInt8 BFirmwareInfo::type

7.3.1.5 ver0

BUInt8 BFirmwareInfo::ver0

7.3.1.6 ver1

BUInt8 BFirmwareInfo::ver1

7.3.1.7 ver2

BUInt8 BFirmwareInfo::ver2

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

- [Dfu.cpp](#)

7.4 BMdns Class Reference

```
#include <BMdns.h>
```

Public Member Functions

- [BMdns](#) ()
- [~BMdns](#) ()
- [BError](#) [init](#) ()
- [BError](#) [findServices](#) ([BString](#) service, [BUInt32](#) timeoutMs, [BList](#)< [BMdnsService](#) > &services)

Private Attributes

- [BSocket](#) [osocket](#)
- [BUInt32](#) [otransactionId](#)

7.4.1 Constructor & Destructor Documentation

7.4.1.1 BMdns()

```
BMdns::BMdns ( )
```

7.4.1.2 ~BMdns()

```
BMdns::~~BMdns ( )
```

7.4.2 Member Function Documentation

7.4.2.1 findServices()

```
BError BMdns::findServices (
    BString service,
    BUInt32 timeoutMs,
    BList< BMdnsService > & services )
```

Unicast response, class IN

7.4.2.2 init()

```
BError BMdns::init ( )
```

7.4.3 Member Data Documentation

7.4.3.1 osocket

```
BSocket BMdns::osocket [private]
```

7.4.3.2 otransactionId

```
BUInt32 BMdns::otransactionId [private]
```

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

- [BMdns.h](#)
- [BMdns.cpp](#)

7.5 BMdnsService Class Reference

```
#include <BMdns.h>
```

Public Attributes

- **BString** [name](#)
- **BSocketAddressINET** [address](#)
- **BString** [hostname](#)
- **BStringList** [extra](#)

7.5.1 Member Data Documentation

7.5.1.1 address

```
BSocketAddressINET BMdnsService::address
```


7.5.1.2 extra

```
BStringList BMdnsService::extra
```

7.5.1.3 hostname

```
BString BMdnsService::hostname
```

7.5.1.4 name

```
BString BMdnsService::name
```

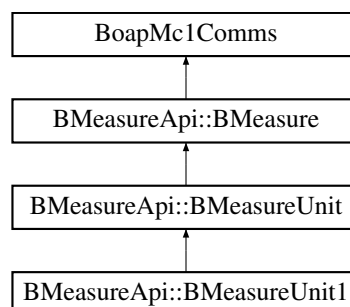
The documentation for this class was generated from the following file:

- [BMdns.h](#)

7.6 BMeasureApi::BMeasure Class Reference

```
#include <BMeasureB.h>
```

Inheritance diagram for BMeasureApi::BMeasure:



Public Member Functions

- **BMeasure** (**Bool** threaded=0, **BUInt** reqSize=512)
- **BError** **getNodeInfo** (**NodeInfo** &nodeInfo)
Get node information.
- **BError** **login** (const **BString** &userid, const **BString** &password)
Provides user/password information for secure connection.
- **BError** **logout** ()
Logs out.
- **BError** **changePassword** (const **BString** &userid, const **BString** &oldPassword, const **BString** &newPassword)
Changes the users password.
- void **factoryReset** (const **Bool** &bootLoader, const **Bool** &resetConfig)
Factory reset.
- void **sendTime** (const **BTimeUs** &time)
Sends the current time.
- **BError** **setMode** (const **Mode** &mode)
Set the current operational mode.
- **BError** **getStatus** (**NodeStatus** &nodeStatus)
Get the node status.
- void **sendStatus** (const **NodeStatus** &nodeStatus)
Sends the current status.
- **BError** **getInformation** (**Information** &info)
- **BError** **getInfoBlock** (**InfoBlock** &infoBlock)
- **BError** **getChannelConfig** (const **BUInt32** &channelNumber, **ChannelConfig** &channelConfig)
- **BError** **setChannelConfig** (const **BUInt32** &channelNumber, const **ChannelConfig** &channelConfig)
- **BError** **setChannelConfigFull** (const **BUInt64** &key, const **BUInt32** &channelNumber, const **ChannelConfig** &channelConfig)
- **BError** **getConfig** (**Configuration** &config)
Return units configuration.
- **BError** **setConfig** (const **Configuration** &config)
Set units configuration.
- **BError** **getMeasurementConfig** (const **Bool** &saved, **MeasurementConfig** &measurementConfig)
Get measurement config.
- **BError** **setMeasurementConfig** (const **Bool** &save, const **MeasurementConfig** &measurementConfig)
Set measurement config.
- **BError** **sendDataEnable** (const **BUInt8** &sendSet)
Enable the sending of different data streams.
- **BError** **measure** (const **DataType** &dataType, **DataBlock** &dataBlock)
Performs a single sample measurement.
- void **sendInfo** (const **InfoBlock** &infoBlock)
Sends an info block.
- void **sendData** (const **DataBlock** &dataBlock)
Sends a data block.
- void **sendChannelConfig** (const **ChannelConfig** &channelConfig)
Sends a ChannelConfig block.
- **BError** **getAwgConfig** (const **Bool** &saved, **AwgConfig** &awgConfig)
Get AWG Configuration.
- **BError** **setAwgConfig** (const **Bool** &save, const **AwgConfig** &awgConfig)
Configure AWG.
- **BError** **setAwgWaveform** (const **BUInt32** &chan, const **BUInt32** &pos, const **FileData** &dataBlock)
Configure AWG Arbitrary waveform.

- **BError** [setAnalogueOut](#) (const **BUInt32** &chan, const **BFloat32** &value)
Set analogue output value.
- **BError** [setDigital](#) (const **BUInt32** &bits)
Set digital bits.
- **BError** [getDigital](#) (**BUInt32** &bits)
Get digital bits.
- **BError** [setRelay](#) (const **BUInt32** &relayNum, const **Bool** &state)
Set relay.
- **BError** [getSwitch](#) (const **BUInt32** &switchNum, **Bool** &state)
Get digital bits.
- **BError** [alarmsClear](#) (const **BUInt32** &bits)
Clear alarms.
- **BError** [fileSysInfo](#) (const **BString** &path, [FileSysInfo](#) &fileSysInfo)
- **BError** [fileSysDelete](#) (const **BString** &path, const [FileSysDeleteType](#) &deleteType)
- **BError** [fileList](#) (const **BString** &path, const **BUInt32** &pos, [FileInfo](#) &fileInfo)
- **BError** [fileOpen](#) (const **BString** &name, const **BString** &mode, **BUInt32** &handle)
- **BError** [fileRead](#) (const **BUInt32** &handle, const **BUInt32** &pos, const **BUInt32** &len, [FileData](#) & data)
- **BError** [fileWrite](#) (const **BUInt32** &handle, const **BUInt32** &pos, const [FileData](#) & data)
- **BError** [fileClose](#) (const **BUInt32** &handle)
- **BError** [fileDelete](#) (const **BString** &name)
- **BError** [functionUnLock](#) (const **BUInt32** &unlocks, const **BString** &key)
UnLock/Lock special functions.
- **BError** [getBoardConfig](#) ([BoardConfig](#) &config)
Get the boards configuration.
- **BError** [setBoardConfig](#) (const [BoardConfig](#) &config)
Sets the boards configuration, requires key.
- **BError** [runBoardTest](#) (const **BString** &test)
Runs the given board test.
- **BError** [calibrate](#) (const [CalibrateInfo](#) &calibInfo)
Calibrate system.
- void [sendMessage](#) (const **BUInt32** &source, const **BString** &message)
Send text messages.
- **BError** [processRequest](#) ()
- virtual **BError** [getNodeInfoServe](#) ([NodeInfo](#) &nodeInfo)
- virtual **BError** [loginServe](#) (const **BString** &userid, const **BString** &password)
- virtual **BError** [logoutServe](#) ()
- virtual **BError** [changePasswordServe](#) (const **BString** &userid, const **BString** &oldPassword, const **BString** &newPassword)
- virtual void [factoryResetServe](#) (const **Bool** &bootLoader, const **Bool** &resetConfig)
- virtual void [sendTimeServe](#) (const **BTimeUs** &time)
- virtual **BError** [setModeServe](#) (const [Mode](#) &mode)
- virtual **BError** [getStatusServe](#) ([NodeStatus](#) &nodeStatus)
- virtual void [sendStatusServe](#) (const [NodeStatus](#) &nodeStatus)
- virtual **BError** [getInformationServe](#) ([Information](#) &info)
- virtual **BError** [getInfoBlockServe](#) ([InfoBlock](#) &infoBlock)
- virtual **BError** [getChannelConfigServe](#) (const **BUInt32** &channelNumber, [ChannelConfig](#) &channelConfig)
- virtual **BError** [setChannelConfigServe](#) (const **BUInt32** &channelNumber, const [ChannelConfig](#) &channelConfig)
- virtual **BError** [setChannelConfigFullServe](#) (const **BUInt64** &key, const **BUInt32** &channelNumber, const [ChannelConfig](#) &channelConfig)
- virtual **BError** [getConfigServe](#) ([Configuration](#) &config)
- virtual **BError** [setConfigServe](#) (const [Configuration](#) &config)

- virtual **BError** [getMeasurementConfigServe](#) (const **Bool** &saved, [MeasurementConfig](#) &measurement←
Config)
- virtual **BError** [setMeasurementConfigServe](#) (const **Bool** &save, const [MeasurementConfig](#) &measurement←
Config)
- virtual **BError** [sendDataEnableServe](#) (const **BUInt8** &sendSet)
- virtual **BError** [measureServe](#) (const [DataType](#) &dataType, [DataBlock](#) &dataBlock)
- virtual void [sendInfoServe](#) (const [InfoBlock](#) &infoBlock)
- virtual void [sendDataServe](#) (const [DataBlock](#) &dataBlock)
- virtual void [sendChannelConfigServe](#) (const [ChannelConfig](#) &channelConfig)
- virtual **BError** [getAwgConfigServe](#) (const **Bool** &saved, [AwgConfig](#) &awgConfig)
- virtual **BError** [setAwgConfigServe](#) (const **Bool** &save, const [AwgConfig](#) &awgConfig)
- virtual **BError** [setAwgWaveformServe](#) (const **BUInt32** &chan, const **BUInt32** &pos, const [FileData](#) &data←
Block)
- virtual **BError** [setAnalogueOutServe](#) (const **BUInt32** &chan, const **BFloat32** &value)
- virtual **BError** [setDigitalServe](#) (const **BUInt32** &bits)
- virtual **BError** [getDigitalServe](#) (**BUInt32** &bits)
- virtual **BError** [setRelayServe](#) (const **BUInt32** &relayNum, const **Bool** &state)
- virtual **BError** [getSwitchServe](#) (const **BUInt32** &switchNum, **Bool** &state)
- virtual **BError** [alarmsClearServe](#) (const **BUInt32** &bits)
- virtual **BError** [filesysInfoServe](#) (const **BString** &path, [FilesysInfo](#) &filesysInfo)
- virtual **BError** [filesysDeleteServe](#) (const **BString** &path, const [FilesysDeleteType](#) &deleteType)
- virtual **BError** [fileListServe](#) (const **BString** &path, const **BUInt32** &pos, [FileInfo](#) &fileInfo)
- virtual **BError** [fileOpenServe](#) (const **BString** &name, const **BString** &mode, **BUInt32** &handle)
- virtual **BError** [fileReadServe](#) (const **BUInt32** &handle, const **BUInt32** &pos, const **BUInt32** &len, [FileData](#)
& data)
- virtual **BError** [fileWriteServe](#) (const **BUInt32** &handle, const **BUInt32** &pos, const [FileData](#) & data)
- virtual **BError** [fileCloseServe](#) (const **BUInt32** &handle)
- virtual **BError** [fileDeleteServe](#) (const **BString** &name)
- virtual **BError** [functionUnLockServe](#) (const **BUInt32** &unlocks, const **BString** &key)
- virtual **BError** [getBoardConfigServe](#) ([BoardConfig](#) &config)
- virtual **BError** [setBoardConfigServe](#) (const [BoardConfig](#) &config)
- virtual **BError** [runBoardTestServe](#) (const **BString** &test)
- virtual **BError** [calibrateServe](#) (const [CalibratInfo](#) &calibInfo)
- virtual void [sendMessageServe](#) (const **BUInt32** &source, const **BString** &message)

Additional Inherited Members

7.6.1 Constructor & Destructor Documentation

7.6.1.1 BMeasure()

```
BMeasureApi::BMeasure::BMeasure (
    Bool threaded = 0,
    BUInt reqSize = 512 )
```

7.6.2 Member Function Documentation

7.6.2.1 alarmsClear()

```
BError BMeasureApi::BMeasure::alarmsClear (
    const BUInt32 & bits )
```

Clear alarms.

7.6.2.2 alarmsClearServe()

```
BError BMeasureApi::BMeasure::alarmsClearServe (
    const BUInt32 & bits ) [virtual]
```

7.6.2.3 calibrate()

```
BError BMeasureApi::BMeasure::calibrate (
    const CalibrateInfo & calibInfo )
```

Calibrate system.

7.6.2.4 calibrateServe()

```
BError BMeasureApi::BMeasure::calibrateServe (
    const CalibrateInfo & calibInfo ) [virtual]
```

7.6.2.5 changePassword()

```
BError BMeasureApi::BMeasure::changePassword (
    const BString & userid,
    const BString & oldPassword,
    const BString & newPassword )
```

Changes the users password.

7.6.2.6 changePasswordServe()

```
BError BMeasureApi::BMeasure::changePasswordServe (
    const BString & userid,
    const BString & oldPassword,
    const BString & newPassword ) [virtual]
```

7.6.2.7 factoryReset()

```
void BMeasureApi::BMeasure::factoryReset (
    const Bool & bootLoader,
    const Bool & resetConfig )
```

Factory reset.

7.6.2.8 factoryResetServe()

```
void BMeasureApi::BMeasure::factoryResetServe (
    const Bool & bootLoader,
    const Bool & resetConfig ) [virtual]
```

7.6.2.9 fileClose()

```
BError BMeasureApi::BMeasure::fileClose (
    const BUInt32 & handle )
```

7.6.2.10 fileCloseServe()

```
BError BMeasureApi::BMeasure::fileCloseServe (
    const BUInt32 & handle ) [virtual]
```

7.6.2.11 fileDelete()

```
BError BMeasureApi::BMeasure::fileDelete (
    const BString & name )
```

7.6.2.12 fileDeleteServe()

```
BError BMeasureApi::BMeasure::fileDeleteServe (
    const BString & name ) [virtual]
```

7.6.2.13 `fileList()`

```
BError BMeasureApi::BMeasure::fileList (
    const BString & path,
    const BUInt32 & pos,
    FileInfo & fileInfo )
```

7.6.2.14 `fileListServe()`

```
BError BMeasureApi::BMeasure::fileListServe (
    const BString & path,
    const BUInt32 & pos,
    FileInfo & fileInfo ) [virtual]
```

7.6.2.15 `fileOpen()`

```
BError BMeasureApi::BMeasure::fileOpen (
    const BString & name,
    const BString & mode,
    BUInt32 & handle )
```

7.6.2.16 `fileOpenServe()`

```
BError BMeasureApi::BMeasure::fileOpenServe (
    const BString & name,
    const BString & mode,
    BUInt32 & handle ) [virtual]
```

7.6.2.17 `fileRead()`

```
BError BMeasureApi::BMeasure::fileRead (
    const BUInt32 & handle,
    const BUInt32 & pos,
    const BUInt32 & len,
    FileData & data )
```

7.6.2.18 fileReadServe()

```
BError BMeasureApi::BMeasure::fileReadServe (
    const BUInt32 & handle,
    const BUInt32 & pos,
    const BUInt32 & len,
    FileData & data ) [virtual]
```

7.6.2.19 fileSysDelete()

```
BError BMeasureApi::BMeasure::fileSysDelete (
    const BString & path,
    const FileSysDeleteType & deleteType )
```

7.6.2.20 fileSysDeleteServe()

```
BError BMeasureApi::BMeasure::fileSysDeleteServe (
    const BString & path,
    const FileSysDeleteType & deleteType ) [virtual]
```

7.6.2.21 fileSysInfo()

```
BError BMeasureApi::BMeasure::fileSysInfo (
    const BString & path,
    FileSysInfo & fileSysInfo )
```

7.6.2.22 fileSysInfoServe()

```
BError BMeasureApi::BMeasure::fileSysInfoServe (
    const BString & path,
    FileSysInfo & fileSysInfo ) [virtual]
```

7.6.2.23 fileWrite()

```
BError BMeasureApi::BMeasure::fileWrite (
    const BUInt32 & handle,
    const BUInt32 & pos,
    const FileData & data )
```


7.6.2.24 fileWriteServe()

```
BError BMeasureApi::BMeasure::fileWriteServe (
    const BUInt32 & handle,
    const BUInt32 & pos,
    const FileData & data ) [virtual]
```

7.6.2.25 functionUnLock()

```
BError BMeasureApi::BMeasure::functionUnLock (
    const BUInt32 & unlocks,
    const BString & key )
```

UnLock/Lock special functions.

7.6.2.26 functionUnLockServe()

```
BError BMeasureApi::BMeasure::functionUnLockServe (
    const BUInt32 & unlocks,
    const BString & key ) [virtual]
```

7.6.2.27 getAwgConfig()

```
BError BMeasureApi::BMeasure::getAwgConfig (
    const Bool & saved,
    AwgConfig & awgConfig )
```

Get AWG [Configuration](#).

7.6.2.28 getAwgConfigServe()

```
BError BMeasureApi::BMeasure::getAwgConfigServe (
    const Bool & saved,
    AwgConfig & awgConfig ) [virtual]
```

7.6.2.29 getBoardConfig()

```
BError BMeasureApi::BMeasure::getBoardConfig (
    BoardConfig & config )
```

Get the boards configuration.

7.6.2.30 getBoardConfigServe()

```
BError BMeasureApi::BMeasure::getBoardConfigServe (
    BoardConfig & config ) [virtual]
```

7.6.2.31 getChannelConfig()

```
BError BMeasureApi::BMeasure::getChannelConfig (
    const BUInt32 & channelNumber,
    ChannelConfig & channelConfig )
```

7.6.2.32 getChannelConfigServe()

```
BError BMeasureApi::BMeasure::getChannelConfigServe (
    const BUInt32 & channelNumber,
    ChannelConfig & channelConfig ) [virtual]
```

7.6.2.33 getConfig()

```
BError BMeasureApi::BMeasure::getConfig (
    Configuration & config )
```

Return units configuration.

7.6.2.34 getConfigServe()

```
BError BMeasureApi::BMeasure::getConfigServe (
    Configuration & config ) [virtual]
```

7.6.2.35 getDigital()

```
BError BMeasureApi::BMeasure::getDigital (
    BUInt32 & bits )
```

Get digital bits.

7.6.2.36 getDigitalServe()

```
BError BMeasureApi::BMeasure::getDigitalServe (
    BUInt32 & bits ) [virtual]
```

7.6.2.37 getInfoBlock()

```
BError BMeasureApi::BMeasure::getInfoBlock (
    InfoBlock & infoBlock )
```

7.6.2.38 getInfoBlockServe()

```
BError BMeasureApi::BMeasure::getInfoBlockServe (
    InfoBlock & infoBlock ) [virtual]
```

7.6.2.39 getInformation()

```
BError BMeasureApi::BMeasure::getInformation (
    Information & info )
```

7.6.2.40 getInformationServe()

```
BError BMeasureApi::BMeasure::getInformationServe (
    Information & info ) [virtual]
```

7.6.2.41 getMeasurementConfig()

```
BError BMeasureApi::BMeasure::getMeasurementConfig (
    const Bool & saved,
    MeasurementConfig & measurementConfig )
```

Get measurement config.

7.6.2.42 getMeasurementConfigServe()

```
BError BMeasureApi::BMeasure::getMeasurementConfigServe (
    const Bool & saved,
    MeasurementConfig & measurementConfig ) [virtual]
```

7.6.2.43 getNodeInfo()

```
BError BMeasureApi::BMeasure::getNodeInfo (
    NodeInfo & nodeInfo )
```

Get node information.

7.6.2.44 getNodeInfoServe()

```
BError BMeasureApi::BMeasure::getNodeInfoServe (
    NodeInfo & nodeInfo ) [virtual]
```

7.6.2.45 getStatus()

```
BError BMeasureApi::BMeasure::getStatus (
    NodeStatus & nodeStatus )
```

Get the node status.

7.6.2.46 getStatusServe()

```
BError BMeasureApi::BMeasure::getStatusServe (
    NodeStatus & nodeStatus ) [virtual]
```

7.6.2.47 getSwitch()

```
BError BMeasureApi::BMeasure::getSwitch (
    const BUInt32 & switchNum,
    Bool & state )
```

Get digital bits.

7.6.2.48 getSwitchServe()

```
BError BMeasureApi::BMeasure::getSwitchServe (
    const BUInt32 & switchNum,
    Bool & state ) [virtual]
```

7.6.2.49 login()

```
BError BMeasureApi::BMeasure::login (
    const BString & userid,
    const BString & password )
```

Provides user/password information for secure connection.

7.6.2.50 loginServe()

```
BError BMeasureApi::BMeasure::loginServe (
    const BString & userid,
    const BString & password ) [virtual]
```

7.6.2.51 logout()

```
BError BMeasureApi::BMeasure::logout ( )
```

Logs out.

7.6.2.52 logoutServe()

```
BError BMeasureApi::BMeasure::logoutServe ( ) [virtual]
```

7.6.2.53 measure()

```
BError BMeasureApi::BMeasure::measure (
    const DataType & dataType,
    DataBlock & dataBlock )
```

Performs a single sample measurement.

7.6.2.54 measureServe()

```
BError BMeasureApi::BMeasure::measureServe (
    const DataType & dataType,
    DataBlock & dataBlock ) [virtual]
```

7.6.2.55 processRequest()

```
BError BMeasureApi::BMeasure::processRequest ( )
```

7.6.2.56 runBoardTest()

```
BError BMeasureApi::BMeasure::runBoardTest (
    const BString & test )
```

Runs the given board test.

7.6.2.57 runBoardTestServe()

```
BError BMeasureApi::BMeasure::runBoardTestServe (
    const BString & test ) [virtual]
```

7.6.2.58 sendChannelConfig()

```
void BMeasureApi::BMeasure::sendChannelConfig (
    const ChannelConfig & channelConfig )
```

Sends a [ChannelConfig](#) block.

7.6.2.59 sendChannelConfigServe()

```
void BMeasureApi::BMeasure::sendChannelConfigServe (
    const ChannelConfig & channelConfig ) [virtual]
```

7.6.2.60 sendData()

```
void BMeasureApi::BMeasure::sendData (
    const DataBlock & dataBlock )
```

Sends a data block.

7.6.2.61 sendDataEnable()

```
BError BMeasureApi::BMeasure::sendDataEnable (
    const BUInt8 & sendSet )
```

Enable the sending of different data streams.

7.6.2.62 sendDataEnableServe()

```
BError BMeasureApi::BMeasure::sendDataEnableServe (
    const BUInt8 & sendSet ) [virtual]
```

7.6.2.63 sendDataServe()

```
void BMeasureApi::BMeasure::sendDataServe (
    const DataBlock & dataBlock ) [virtual]
```

Reimplemented in [BMeasureApi::BMeasureUnit](#).

7.6.2.64 sendInfo()

```
void BMeasureApi::BMeasure::sendInfo (
    const InfoBlock & infoBlock )
```

Sends an info block.

7.6.2.65 sendInfoServe()

```
void BMeasureApi::BMeasure::sendInfoServe (
    const InfoBlock & infoBlock ) [virtual]
```

7.6.2.66 sendMessage()

```
void BMeasureApi::BMeasure::sendMessage (
    const BUInt32 & source,
    const BString & message )
```

Sends text messages.

7.6.2.67 sendMessageServe()

```
void BMeasureApi::BMeasure::sendMessageServe (
    const BUInt32 & source,
    const BString & message ) [virtual]
```

Reimplemented in [BMeasureApi::BMeasureUnit1](#).

7.6.2.68 sendStatus()

```
void BMeasureApi::BMeasure::sendStatus (
    const NodeStatus & nodeStatus )
```

Sends the current status.

7.6.2.69 sendStatusServe()

```
void BMeasureApi::BMeasure::sendStatusServe (
    const NodeStatus & nodeStatus ) [virtual]
```

7.6.2.70 sendTime()

```
void BMeasureApi::BMeasure::sendTime (
    const BTimeUs & time )
```

Sends the current time.

7.6.2.71 sendTimeServe()

```
void BMeasureApi::BMeasure::sendTimeServe (
    const BTimeUs & time ) [virtual]
```


7.6.2.72 setAnalogueOut()

```
BError BMeasureApi::BMeasure::setAnalogueOut (
    const BUInt32 & chan,
    const BFloat32 & value )
```

Set analogue output value.

7.6.2.73 setAnalogueOutServe()

```
BError BMeasureApi::BMeasure::setAnalogueOutServe (
    const BUInt32 & chan,
    const BFloat32 & value ) [virtual]
```

7.6.2.74 setAwgConfig()

```
BError BMeasureApi::BMeasure::setAwgConfig (
    const Bool & save,
    const AwgConfig & awgConfig )
```

Configure AWG.

7.6.2.75 setAwgConfigServe()

```
BError BMeasureApi::BMeasure::setAwgConfigServe (
    const Bool & save,
    const AwgConfig & awgConfig ) [virtual]
```

7.6.2.76 setAwgWaveform()

```
BError BMeasureApi::BMeasure::setAwgWaveform (
    const BUInt32 & chan,
    const BUInt32 & pos,
    const FileData & dataBlock )
```

Configure AWG Arbitrary waveform.

7.6.2.77 setAwgWaveformServe()

```
BError BMeasureApi::BMeasure::setAwgWaveformServe (
    const BUInt32 & chan,
    const BUInt32 & pos,
    const FileData & dataBlock ) [virtual]
```

7.6.2.78 setBoardConfig()

```
BError BMeasureApi::BMeasure::setBoardConfig (
    const BoardConfig & config )
```

Sets the boards configuration, requires key.

7.6.2.79 setBoardConfigServe()

```
BError BMeasureApi::BMeasure::setBoardConfigServe (
    const BoardConfig & config ) [virtual]
```

7.6.2.80 setChannelConfig()

```
BError BMeasureApi::BMeasure::setChannelConfig (
    const BUInt32 & channelNumber,
    const ChannelConfig & channelConfig )
```

7.6.2.81 setChannelConfigFull()

```
BError BMeasureApi::BMeasure::setChannelConfigFull (
    const BUInt64 & key,
    const BUInt32 & channelNumber,
    const ChannelConfig & channelConfig )
```

7.6.2.82 setChannelConfigFullServe()

```
BError BMeasureApi::BMeasure::setChannelConfigFullServe (
    const BUInt64 & key,
    const BUInt32 & channelNumber,
    const ChannelConfig & channelConfig ) [virtual]
```

7.6.2.83 setChannelConfigServe()

```
BError BMeasureApi::BMeasure::setChannelConfigServe (
    const BUInt32 & channelNumber,
    const ChannelConfig & channelConfig ) [virtual]
```

7.6.2.84 setConfig()

```
BError BMeasureApi::BMeasure::setConfig (
    const Configuration & config )
```

Set units configuration.

7.6.2.85 setConfigServe()

```
BError BMeasureApi::BMeasure::setConfigServe (
    const Configuration & config ) [virtual]
```

7.6.2.86 setDigital()

```
BError BMeasureApi::BMeasure::setDigital (
    const BUInt32 & bits )
```

Set digital bits.

7.6.2.87 setDigitalServe()

```
BError BMeasureApi::BMeasure::setDigitalServe (
    const BUInt32 & bits ) [virtual]
```

7.6.2.88 setMeasurementConfig()

```
BError BMeasureApi::BMeasure::setMeasurementConfig (
    const Bool & save,
    const MeasurementConfig & measurementConfig )
```

Set measurement config.

7.6.2.89 setMeasurementConfigServe()

```
BError BMeasureApi::BMeasure::setMeasurementConfigServe (
    const Bool & save,
    const MeasurementConfig & measurementConfig ) [virtual]
```

7.6.2.90 setMode()

```
BError BMeasureApi::BMeasure::setMode (
    const Mode & mode )
```

Set the current operational mode.

7.6.2.91 setModeServe()

```
BError BMeasureApi::BMeasure::setModeServe (
    const Mode & mode ) [virtual]
```

7.6.2.92 setRelay()

```
BError BMeasureApi::BMeasure::setRelay (
    const BUInt32 & relayNum,
    const Bool & state )
```

Set relay.

7.6.2.93 setRelayServe()

```
BError BMeasureApi::BMeasure::setRelayServe (
    const BUInt32 & relayNum,
    const Bool & state ) [virtual]
```

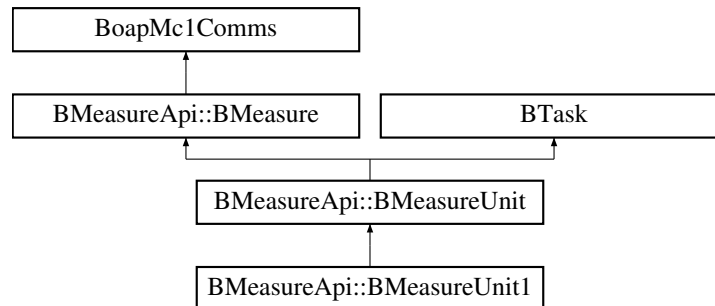
The documentation for this class was generated from the following files:

- [BMeasureB.h](#)
- [BMeasureB-1.cpp](#)
- [BMeasureB.cpp](#)

7.7 BMeasureApi::BMeasureUnit Class Reference

```
#include <BMeasureUnit.h>
```

Inheritance diagram for BMeasureApi::BMeasureUnit:



Public Member Functions

- [BMeasureUnit](#) (**Bool** threaded=0, **BUInt** reqSize=2048)
- virtual [~BMeasureUnit](#) ()
- **BError** [connect](#) (**BString** device)
 - Connect to a device.*
- void [disconnect](#) ()
- **BString** [device](#) ()
- **BString** [serialNumber](#) ()
- **BString** [info](#) ()
- **BUInt** [numChannels](#) ()
 - The number of channels of data.*
- void [run](#) ()
 - Threaded run mode.*
- virtual void [disconnected](#) ()
- virtual void [sendDataServe](#) (const [DataBlock](#) &dataBlock)
- virtual void [sendDataServe1](#) (const [DataBlock](#) &dataBlock)
- virtual **BError** [setMeasurementConfig](#) (const **Bool** &save, const [MeasurementConfig](#) &configMeasurement)
- virtual **BError** [setChannelConfig](#) (const **BUInt8** &channelNumber, const [ChannelConfig](#) &channelConfig)

Static Public Member Functions

- static **BError** [findDevices](#) (**BList**< [BMeasureUnitDevice](#) > &devices)
 - Find available devices.*
- static **BError** [findDevicesUsb](#) (**BList**< [BMeasureUnitDevice](#) > &devices)
 - Find available devices on USB bus.*
- static **BError** [findDevicesNetwork](#) (**BList**< [BMeasureUnitDevice](#) > &devices)
 - Find available devices on Network.*
- static void [processdataBlock](#) (const [DataBlock](#) &dataBlock, [DataBlock](#) *dataBlockOut)

Static Public Attributes

- static int [blockNumChannels](#) = 16
- static int [blockNumSamples](#) = 13

Protected Attributes

- **BString** `odevice`
- **NodeInfo** `onodeInfo`
- **Information** `oinfo`
Instrument info.
- **MeasurementConfig** `oconfigMeasurement`
- **BArray**< **ChannelConfig** > `ochannels`
- **DataBlock** * `odataBlock`
- **BUInt32** `osequenceNext`
- **BUInt32** `osampleCount`
- **BUInt32** `oblockCount`
- **Bool** `odisconnecting`

Additional Inherited Members

7.7.1 Constructor & Destructor Documentation

7.7.1.1 BMeasureUnit()

```
BMeasureApi::BMeasureUnit::BMeasureUnit (
    Bool threaded = 0,
    BUInt reqSize = 2048 )
```

7.7.1.2 ~BMeasureUnit()

```
BMeasureApi::BMeasureUnit::~~BMeasureUnit ( ) [virtual]
```

7.7.2 Member Function Documentation

7.7.2.1 connect()

```
BError BMeasureApi::BMeasureUnit::connect (
    BString device )
```

Connect to a device.

7.7.2.2 device()

```
BString BMeasureApi::BMeasureUnit::device ( )
```

7.7.2.3 disconnect()

```
void BMeasureApi::BMeasureUnit::disconnect ( )
```

7.7.2.4 disconnected()

```
void BMeasureApi::BMeasureUnit::disconnected ( ) [virtual]
```

Reimplemented in [BMeasureApi::BMeasureUnit1](#).

7.7.2.5 findDevices()

```
BError BMeasureApi::BMeasureUnit::findDevices (
    BList< BMeasureUnitDevice > & devices ) [static]
```

Find available devices.

7.7.2.6 findDevicesNetwork()

```
BError BMeasureApi::BMeasureUnit::findDevicesNetwork (
    BList< BMeasureUnitDevice > & devices ) [static]
```

Find available devices on Network.

7.7.2.7 findDevicesUsb()

```
BError BMeasureApi::BMeasureUnit::findDevicesUsb (
    BList< BMeasureUnitDevice > & devices ) [static]
```

Find available devices on USB bus.

7.7.2.8 info()

```
BString BMeasureApi::BMeasureUnit::info ( )
```

7.7.2.9 numChannels()

```
BUInt BMeasureApi::BMeasureUnit::numChannels ( )
```

The number of channels of data.

7.7.2.10 processDataBlock()

```
void BMeasureApi::BMeasureUnit::processdataBlock (
    const DataBlock & dataBlock,
    DataBlock * dataBlockOut ) [static]
```

7.7.2.11 run()

```
void BMeasureApi::BMeasureUnit::run ( ) [virtual]
```

Threaded run mode.

Reimplemented from **BTask**.

7.7.2.12 sendDataServe()

```
void BMeasureApi::BMeasureUnit::sendDataServe (
    const DataBlock & dataBlock ) [virtual]
```

Reimplemented from [BMeasureApi::BMeasure](#).

7.7.2.13 sendDataServe1()

```
void BMeasureApi::BMeasureUnit::sendDataServe1 (
    const DataBlock & dataBlock ) [virtual]
```

Reimplemented in [BMeasureApi::BMeasureUnit1](#).

7.7.2.14 serialNumber()

```
BString BMeasureApi::BMeasureUnit::serialNumber ( )
```

7.7.2.15 setChannelConfig()

```
BError BMeasureApi::BMeasureUnit::setChannelConfig (
    const BUInt8 & channelNumber,
    const ChannelConfig & channelConfig ) [virtual]
```

7.7.2.16 setMeasurementConfig()

```
BError BMeasureApi::BMeasureUnit::setMeasurementConfig (
    const Bool & save,
    const MeasurementConfig & configMeasurement ) [virtual]
```

7.7.3 Member Data Documentation

7.7.3.1 blockNumChannels

```
int BMeasureApi::BMeasureUnit::blockNumChannels = 16 [static]
```

7.7.3.2 blockNumSamples

```
int BMeasureApi::BMeasureUnit::blockNumSamples = 13 [static]
```

7.7.3.3 oblockCount

```
BUInt32 BMeasureApi::BMeasureUnit::oblockCount [protected]
```

7.7.3.4 ochannels

```
BArray<ChannelConfig> BMeasureApi::BMeasureUnit::ochannels [protected]
```

7.7.3.5 oconfigMeasurement

`MeasurementConfig` `BMeasureApi::BMeasureUnit::oconfigMeasurement` [protected]

7.7.3.6 odataBlock

`DataBlock*` `BMeasureApi::BMeasureUnit::odataBlock` [protected]

7.7.3.7 odevice

`BString` `BMeasureApi::BMeasureUnit::odevice` [protected]

7.7.3.8 odisconnecting

`Bool` `BMeasureApi::BMeasureUnit::odisconnecting` [protected]

7.7.3.9 oinfo

`Information` `BMeasureApi::BMeasureUnit::oinfo` [protected]

Instrument info.

7.7.3.10 onodeInfo

`NodeInfo` `BMeasureApi::BMeasureUnit::onodeInfo` [protected]

7.7.3.11 osampleCount

`BUInt32` `BMeasureApi::BMeasureUnit::osampleCount` [protected]

7.7.3.12 osequenceNext

```
BUInt32 BMeasureApi::BMeasureUnit::osequenceNext [protected]
```

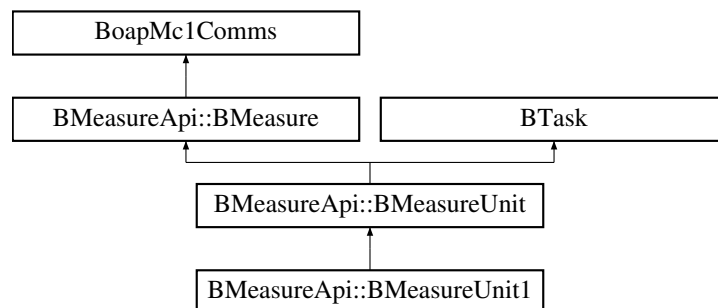
The documentation for this class was generated from the following files:

- [BMeasureUnit.h](#)
- [BMeasureUnit.cpp](#)

7.8 BMeasureApi::BMeasureUnit1 Class Reference

```
#include <BMeasureUnits.h>
```

Inheritance diagram for BMeasureApi::BMeasureUnit1:



Public Member Functions

- [BMeasureUnit1](#) ([BMeasureUnits](#) &measureUnits, [BString](#) device, [Bool](#) threaded=0, [BUInt](#) reqSize=2048)
- [BString](#) serialNumber ()
- void [setSerialNumber](#) ([BString](#) serialNumber)
- void [disconnected](#) ()
- void [sendDataServe1](#) (const [DataBlock](#) &dataBlock)
- void [sendMessageServe](#) (const [BUInt32](#) &source, const [BString](#) &message)

Public Attributes

- [BMeasureUnits](#) & omeasureUnits
- [Bool](#) oenabled
- [Bool](#) oconnected
- [BUInt](#) oorder
- [BUInt](#) osource
- [BString](#) oserialNumber

Additional Inherited Members

7.8.1 Constructor & Destructor Documentation

7.8.1.1 BMeasureUnit1()

```
BMeasureApi::BMeasureUnit1::BMeasureUnit1 (
    BMeasureUnits & measureUnits,
    BString device,
    Bool threaded = 0,
    BUInt reqSize = 2048 )
```

7.8.2 Member Function Documentation

7.8.2.1 disconnected()

```
void BMeasureApi::BMeasureUnit1::disconnected ( ) [virtual]
```

Reimplemented from [BMeasureApi::BMeasureUnit](#).

7.8.2.2 sendDataServe1()

```
void BMeasureApi::BMeasureUnit1::sendDataServe1 (
    const DataBlock & dataBlock ) [virtual]
```

Reimplemented from [BMeasureApi::BMeasureUnit](#).

7.8.2.3 sendMessageServe()

```
void BMeasureApi::BMeasureUnit1::sendMessageServe (
    const BUInt32 & source,
    const BString & message ) [virtual]
```

Reimplemented from [BMeasureApi::BMeasure](#).

7.8.2.4 serialNumber()

```
BString BMeasureApi::BMeasureUnit1::serialNumber ( )
```

7.8.2.5 setSerialNumber()

```
void BMeasureApi::BMeasureUnit1::setSerialNumber (
    BString serialNumber )
```

7.8.3 Member Data Documentation

7.8.3.1 oconnected

Bool BMeasureApi::BMeasureUnit1::oconnected

7.8.3.2 oenabled

Bool BMeasureApi::BMeasureUnit1::oenabled

7.8.3.3 omeasureUnits

[BMeasureUnits](#)& BMeasureApi::BMeasureUnit1::omeasureUnits

7.8.3.4 oorder

BUInt BMeasureApi::BMeasureUnit1::oorder

7.8.3.5 oserialNumber

BString BMeasureApi::BMeasureUnit1::oserialNumber

7.8.3.6 osource

BUInt BMeasureApi::BMeasureUnit1::osource

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

- [BMeasureUnits.h](#)
- [BMeasureUnits.cpp](#)

7.9 BMeasureApi::BMeasureUnitDevice Class Reference

```
#include <BMeasureUnit.h>
```

Public Member Functions

- [BMeasureUnitDevice](#) (**BString** serialNumber="", **BString** device="")

Public Attributes

- **BString** serialNumber
- **BString** device

7.9.1 Constructor & Destructor Documentation

7.9.1.1 BMeasureUnitDevice()

```
BMeasureApi::BMeasureUnitDevice::BMeasureUnitDevice (
    BString serialNumber = "",
    BString device = "" ) [inline]
```

7.9.2 Member Data Documentation

7.9.2.1 device

```
BString BMeasureApi::BMeasureUnitDevice::device
```

7.9.2.2 serialNumber

```
BString BMeasureApi::BMeasureUnitDevice::serialNumber
```

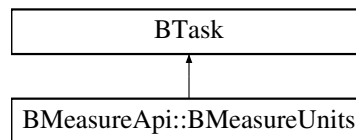
The documentation for this class was generated from the following file:

- [BMeasureUnit.h](#)

7.10 BMeasureApi::BMeasureUnits Class Reference

```
#include <BMeasureUnits.h>
```

Inheritance diagram for BMeasureApi::BMeasureUnits:



Public Member Functions

- [BMeasureUnits](#) (**Bool** threaded=0)
- virtual [~BMeasureUnits](#) ()
- void [clear](#) ()
- **BError** [unitsFind](#) ()
- **BError** [unitAdd](#) (**BString** serialNumber, **BString** device)
- **BError** [unitDelete](#) (**BString** device)
- **BUInt32** [unitsNum](#) ()
- **BUInt32** [unitsConnectedNum](#) ()
- [BMeasureUnit1](#) & [unit](#) (**BUInt** u)
- [BMeasureUnit1](#) & [unitMaster](#) ()
- **BError** [unitsConnect](#) ()
- **Bool** [unitsConnected](#) ()
- **BError** [unitsDisconnect](#) ()
- virtual void [disconnected](#) ()
- **BError** [unitSetOrder](#) (**BUInt** u, **BUInt** order, **Bool** move)
- **BError** [unitSetEnabled](#) (**BUInt** u, **Bool** enable)
- **BError** [dataSetNumStreams](#) (**BUInt** num)
 - *Set the number of data output channels.*
- void [dataStreamEnable](#) (**Bool** on)
 - *Enable the streaming of data.*
- void [dataClear](#) ()
- **BUInt** [dataAvailable](#) (**BUInt** stream)
- **BError** [dataWait](#) (**BUInt** stream, **BTimeout** timeoutUs= **BTimeoutForever**)
- virtual void [dataEvent](#) (**BUInt** stream)
- [DataBlock](#) * [dataRead](#) (**BUInt** stream)
- void [dataDone](#) (**BUInt** stream)
- virtual void [dataProcEvent](#) (**BUInt** stream)
- [DataBlockProc](#) * [dataProcRead](#) (**BUInt** stream)
- void [dataProcDone](#) (**BUInt** stream)
- void [run](#) ()
 - *Threaded run mode.*
- void [sendDataQueue](#) (const [DataBlock](#) &dataBlock)
- void [sendDataProcQueue](#) (const [DataBlock](#) &dataBlock)
- void [sendDataProcess](#) ()
- void [sendDataProcessTrigger](#) ()
- void [outputBlock](#) ([BMeasureUnitsDataBlock](#) *block)
- virtual **BError** [getNodeInfo](#) ([NodeInfo](#) &nodeInfo)
- virtual **BError** [login](#) (const **BString** &userid, const **BString** &password)

- virtual **BError** [logout](#) ()
- virtual **BError** [changePassword](#) (const **BString** &userid, const **BString** &oldPassword, const **BString** &newPassword)
- virtual **BUInt** [numChannels](#) ()
 - The number of channels of data.*
- virtual **BError** [setMode](#) (const **Mode** &mode)
 - Set the current operational mode.*
- virtual **BError** [getStatus](#) (**NodeStatus** &nodeStatus)
- virtual void [sendTime](#) (const **BTimeUs** &time)
 - Sends the current time.*
- virtual **BError** [getInformation](#) (**Information** &info)
- virtual **BError** [getInfoBlock](#) (**InfoBlock** &infoBlock)
- virtual **BError** [getChannelConfig](#) (const **BUInt8** &channelNumber, **ChannelConfig** &channelConfig)
- virtual **BError** [setChannelConfig](#) (const **BUInt8** &channelNumber, const **ChannelConfig** &channelConfig)
- virtual **BError** [getConfig](#) (**Configuration** &config)
 - Should we have this, not generic for different instruments ?*
- virtual **BError** [setConfig](#) (const **Configuration** &config)
 - Should we have this, not generic for different instruments ?*
- virtual **BError** [getMeasurementConfig](#) (const **Bool** &saved, **MeasurementConfig** &measurement)
 - Get measurement config.*
- virtual **BError** [setMeasurementConfig](#) (const **Bool** &save, const **MeasurementConfig** &measurement)
 - Set measurement config.*
- virtual **BError** [sendDataEnable](#) (const **DataSend** &dataSend)
 - Enables the sending of data.*
- virtual **BError** [getAwgConfig](#) (const **Bool** &saved, **AwgConfig** &awgConfig)
 - Get AWG Configuration.*
- virtual **BError** [setAwgConfig](#) (const **Bool** &save, const **AwgConfig** &awgConfig)
 - Configure AWG.*
- virtual **BError** [alarmsClear](#) (const **BUInt32** &bits)
 - Clear all alarms.*
- virtual void [sendDataServe1](#) (const **DataBlock** &dataBlock)
- virtual void [sendMessage](#) (**BUInt32** &source, **BString** &message)
- virtual void [sendMessageServe](#) (const **BUInt32** &source, const **BString** &message)
- void [debugPrint](#) ()

Private Member Functions

- **BMeasureUnitsDataBlock** * [getFreeBlock](#) (**BUInt** numSamples)

Private Attributes

- **BSemaphoreBool** [oprocEnable](#)
 - Enable processing.*
- **BSemaphoreBool** [oprocRunning](#)
 - Processing is running.*
- **BMutex** [olockUnits](#)
- **BList**< **BMeasureUnit1** * > [ounits](#)
- **BInt** [ounitMaster](#)
- **BUInt** [onumConnected](#)
- **BUInt** [onumChannels](#)
- **BUInt** [odataStreamNum](#)

- **BUInt32** *ofill*
- **BUInt** *onumBlocks*
- **BMutex** *olockInput*
- **BList**< [BMeasureUnitsDataBlock](#) * > *odataBlocksFree*
- **BList**< [BMeasureUnitsDataBlock](#) * > *odataBlocksIn*
- **BList**< [BMeasureUnitsDataBlock](#) * > *odataBlocksProcess*
- **BCondInt** *odataBlocksProcessNum*
- **BMutex** *olockOutput*
- **BList**< [BMeasureUnitsDataBlock](#) * > *odataBlocksOut* [2]
- **BCondInt** *odataBlocksOutCount* [2]
- **BMutex** *olockProInput*
- **BList**< [DataBlock](#) > *odataProcBlocks*
- [MeasurementConfig](#) *olocalTrigger*
- **Bool** *otriggered*
- **BUInt** *ostartSample*

Additional Inherited Members

7.10.1 Constructor & Destructor Documentation

7.10.1.1 BMeasureUnits()

```
BMeasureApi::BMeasureUnits::BMeasureUnits (
    Bool threaded = 0 )
```

7.10.1.2 ~BMeasureUnits()

```
BMeasureApi::BMeasureUnits::~~BMeasureUnits ( ) [virtual]
```

7.10.2 Member Function Documentation

7.10.2.1 alarmsClear()

```
BError BMeasureApi::BMeasureUnits::alarmsClear (
    const BUInt32 & bits ) [virtual]
```

Clear all alarms.

7.10.2.2 changePassword()

```
BError BMeasureApi::BMeasureUnits::changePassword (
    const BString & userid,
    const BString & oldPassword,
    const BString & newPassword ) [virtual]
```

7.10.2.3 clear()

```
void BMeasureApi::BMeasureUnits::clear ( )
```

7.10.2.4 dataAvailable()

```
BUInt BMeasureApi::BMeasureUnits::dataAvailable (
    BUInt stream )
```

7.10.2.5 dataClear()

```
void BMeasureApi::BMeasureUnits::dataClear ( )
```

7.10.2.6 dataDone()

```
void BMeasureApi::BMeasureUnits::dataDone (
    BUInt stream )
```

7.10.2.7 dataEvent()

```
void BMeasureApi::BMeasureUnits::dataEvent (
    BUInt stream ) [virtual]
```

7.10.2.8 dataProcDone()

```
void BMeasureApi::BMeasureUnits::dataProcDone (
    BUInt stream )
```

7.10.2.9 dataProcEvent()

```
void BMeasureApi::BMeasureUnits::dataProcEvent (
    BUInt stream ) [virtual]
```

7.10.2.10 dataProcRead()

```
DataBlockProc * BMeasureApi::BMeasureUnits::dataProcRead (
    BUInt stream )
```

7.10.2.11 dataRead()

```
DataBlock * BMeasureApi::BMeasureUnits::dataRead (
    BUInt stream )
```

7.10.2.12 dataSetNumStreams()

```
BError BMeasureApi::BMeasureUnits::dataSetNumStreams (
    BUInt num )
```

Set the number of data output channels.

7.10.2.13 dataStreamEnable()

```
void BMeasureApi::BMeasureUnits::dataStreamEnable (
    Bool on )
```

Enable the streaming of data.

7.10.2.14 dataWait()

```
BError BMeasureApi::BMeasureUnits::dataWait (
    BUInt stream,
    BTimeout timeoutUs = BTimeoutForever )
```

7.10.2.15 debugPrint()

```
void BMeasureApi::BMeasureUnits::debugPrint ( )
```

7.10.2.16 disconnected()

```
void BMeasureApi::BMeasureUnits::disconnected ( ) [virtual]
```

7.10.2.17 getAwgConfig()

```
BError BMeasureApi::BMeasureUnits::getAwgConfig (
    const Bool & saved,
    AwgConfig & awgConfig ) [virtual]
```

Get AWG [Configuration](#).

7.10.2.18 getChannelConfig()

```
BError BMeasureApi::BMeasureUnits::getChannelConfig (
    const BUInt8 & channelNumber,
    ChannelConfig & channelConfig ) [virtual]
```

7.10.2.19 getConfig()

```
BError BMeasureApi::BMeasureUnits::getConfig (
    Configuration & config ) [virtual]
```

Should we have this, not generic for different instruments ?

7.10.2.20 getFreeBlock()

```
BMeasureUnitsDataBlock * BMeasureApi::BMeasureUnits::getFreeBlock (
    BUInt numSamples ) [private]
```

7.10.2.21 getInfoBlock()

```
BError BMeasureApi::BMeasureUnits::getInfoBlock (
    InfoBlock & infoBlock ) [virtual]
```

7.10.2.22 getInformation()

```
BError BMeasureApi::BMeasureUnits::getInformation (
    Information & info ) [virtual]
```

7.10.2.23 getMeasurementConfig()

```
BError BMeasureApi::BMeasureUnits::getMeasurementConfig (
    const Bool & saved,
    MeasurementConfig & measurement ) [virtual]
```

Get measurement config.

7.10.2.24 getNodeInfo()

```
BError BMeasureApi::BMeasureUnits::getNodeInfo (
    NodeInfo & nodeInfo ) [virtual]
```

7.10.2.25 getStatus()

```
BError BMeasureApi::BMeasureUnits::getStatus (
    NodeStatus & nodeStatus ) [virtual]
```

7.10.2.26 login()

```
BError BMeasureApi::BMeasureUnits::login (
    const BString & userid,
    const BString & password ) [virtual]
```

7.10.2.27 logout()

```
BError BMeasureApi::BMeasureUnits::logout ( ) [virtual]
```

7.10.2.28 numChannels()

```
BUInt BMeasureApi::BMeasureUnits::numChannels ( ) [virtual]
```

The number of channels of data.

7.10.2.29 outputBlock()

```
void BMeasureApi::BMeasureUnits::outputBlock (
    BMeasureUnitsDataBlock * block )
```

7.10.2.30 run()

```
void BMeasureApi::BMeasureUnits::run ( ) [virtual]
```

Threaded run mode.

Reimplemented from **BTask**.

7.10.2.31 sendDataEnable()

```
BError BMeasureApi::BMeasureUnits::sendDataEnable (
    const DataSend & dataSend ) [virtual]
```

Enables the sending of data.

7.10.2.32 sendDataProcess()

```
void BMeasureApi::BMeasureUnits::sendDataProcess ( )
```

7.10.2.33 sendDataProcessTrigger()

```
void BMeasureApi::BMeasureUnits::sendDataProcessTrigger ( )
```

7.10.2.34 sendDataProcQueue()

```
void BMeasureApi::BMeasureUnits::sendDataProcQueue (
    const DataBlock & dataBlock )
```

7.10.2.35 sendDataQueue()

```
void BMeasureApi::BMeasureUnits::sendDataQueue (
    const DataBlock & dataBlock )
```

7.10.2.36 sendDataServe1()

```
void BMeasureApi::BMeasureUnits::sendDataServe1 (
    const DataBlock & dataBlock ) [virtual]
```

7.10.2.37 sendMessage()

```
void BMeasureApi::BMeasureUnits::sendMessage (
    BUInt32 & source,
    BString & message ) [virtual]
```

7.10.2.38 sendMessageServe()

```
void BMeasureApi::BMeasureUnits::sendMessageServe (
    const BUInt32 & source,
    const BString & message ) [virtual]
```

7.10.2.39 sendTime()

```
void BMeasureApi::BMeasureUnits::sendTime (
    const BTimeUs & time ) [virtual]
```

Sends the current time.

7.10.2.40 setAwgConfig()

```
BError BMeasureApi::BMeasureUnits::setAwgConfig (
    const Bool & save,
    const AwgConfig & awgConfig ) [virtual]
```

Configure AWG.

7.10.2.41 setChannelConfig()

```
BError BMeasureApi::BMeasureUnits::setChannelConfig (
    const BUInt8 & channelNumber,
    const ChannelConfig & channelConfig ) [virtual]
```

7.10.2.42 setConfig()

```
BError BMeasureApi::BMeasureUnits::setConfig (
    const Configuration & config ) [virtual]
```

Should we have this, not generic for different instruments ?

7.10.2.43 setMeasurementConfig()

```
BError BMeasureApi::BMeasureUnits::setMeasurementConfig (
    const Bool & save,
    const MeasurementConfig & measurement ) [virtual]
```

Set measurement config.

7.10.2.44 setMode()

```
BError BMeasureApi::BMeasureUnits::setMode (
    const Mode & mode ) [virtual]
```

Set the current operational mode.

7.10.2.45 unit()

```
BMeasureUnit1 & BMeasureApi::BMeasureUnits::unit (
    BUInt u )
```

7.10.2.46 unitAdd()

```
BError BMeasureApi::BMeasureUnits::unitAdd (
    BString serialNumber,
    BString device )
```

7.10.2.47 unitDelete()

```
BError BMeasureApi::BMeasureUnits::unitDelete (
    BString device )
```

7.10.2.48 unitMaster()

```
BMeasureUnit1 & BMeasureApi::BMeasureUnits::unitMaster ( )
```

7.10.2.49 unitsConnect()

```
BError BMeasureApi::BMeasureUnits::unitsConnect ( )
```

7.10.2.50 unitsConnected()

```
Bool BMeasureApi::BMeasureUnits::unitsConnected ( )
```

7.10.2.51 unitsConnectedNum()

```
BUInt BMeasureApi::BMeasureUnits::unitsConnectedNum ( )
```

7.10.2.52 unitsDisconnect()

```
BError BMeasureApi::BMeasureUnits::unitsDisconnect ( )
```

7.10.2.53 unitSetEnabled()

```
BError BMeasureApi::BMeasureUnits::unitSetEnabled (
    BUInt u,
    Bool enable )
```

7.10.2.54 unitSetOrder()

```
BError BMeasureApi::BMeasureUnits::unitSetOrder (
    BUInt u,
    BUInt order,
    Bool move )
```

7.10.2.55 unitsFind()

```
BError BMeasureApi::BMeasureUnits::unitsFind ( )
```

7.10.2.56 unitsNum()

```
BUInt BMeasureApi::BMeasureUnits::unitsNum ( )
```

7.10.3 Member Data Documentation

7.10.3.1 odataBlocksFree

```
BList<BMeasureUnitsDataBlock\*> BMeasureApi::BMeasureUnits::odataBlocksFree [private]
```

7.10.3.2 odataBlocksIn

BList<BMeasureUnitsDataBlock*> BMeasureApi::BMeasureUnits::odataBlocksIn [private]

7.10.3.3 odataBlocksOut

BList<BMeasureUnitsDataBlock*> BMeasureApi::BMeasureUnits::odataBlocksOut[2] [private]

7.10.3.4 odataBlocksOutCount

BCondInt BMeasureApi::BMeasureUnits::odataBlocksOutCount[2] [private]

7.10.3.5 odataBlocksProcess

BList<BMeasureUnitsDataBlock*> BMeasureApi::BMeasureUnits::odataBlocksProcess [private]

7.10.3.6 odataBlocksProcessNum

BCondInt BMeasureApi::BMeasureUnits::odataBlocksProcessNum [private]

7.10.3.7 odataProcBlocks

BList<DataBlock> BMeasureApi::BMeasureUnits::odataProcBlocks [private]

7.10.3.8 odataStreamNum

BUInt BMeasureApi::BMeasureUnits::odataStreamNum [private]

7.10.3.9 ofill

BUInt32 BMeasureApi::BMeasureUnits::ofill [private]

7.10.3.10 olocalTrigger

`MeasurementConfig` `BMeasureApi::BMeasureUnits::olocalTrigger` [private]

7.10.3.11 olockInput

`BMutex` `BMeasureApi::BMeasureUnits::olockInput` [private]

7.10.3.12 olockOutput

`BMutex` `BMeasureApi::BMeasureUnits::olockOutput` [private]

7.10.3.13 olockProcInput

`BMutex` `BMeasureApi::BMeasureUnits::olockProcInput` [private]

7.10.3.14 olockUnits

`BMutex` `BMeasureApi::BMeasureUnits::olockUnits` [private]

7.10.3.15 onumBlocks

`BUInt` `BMeasureApi::BMeasureUnits::onumBlocks` [private]

7.10.3.16 onumChannels

`BUInt` `BMeasureApi::BMeasureUnits::onumChannels` [private]

7.10.3.17 onumConnected

`BUInt` `BMeasureApi::BMeasureUnits::onumConnected` [private]

7.10.3.18 oprocEnable

BSemaphoreBool BMeasureApi::BMeasureUnits::oprocEnable [private]

Enable processing.

7.10.3.19 oprocRunning

BSemaphoreBool BMeasureApi::BMeasureUnits::oprocRunning [private]

Processing is running.

7.10.3.20 ostartSample

BUInt BMeasureApi::BMeasureUnits::ostartSample [private]

7.10.3.21 otriggered

Bool BMeasureApi::BMeasureUnits::ottriggered [private]

7.10.3.22 ounitMaster

BInt BMeasureApi::BMeasureUnits::ounitMaster [private]

7.10.3.23 ounits

BList<[BMeasureUnit1*](#)> BMeasureApi::BMeasureUnits::ounits [private]

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

- [BMeasureUnits.h](#)
- [BMeasureUnits.cpp](#)

7.11 BMeasureApi::BMeasureUnitsDataBlock Class Reference

```
#include <BMeasureUnits.h>
```

Public Member Functions

- [BMeasureUnitsDataBlock](#) (**BUInt** numChannels=0, **BUInt** numSamples=0)
- [~BMeasureUnitsDataBlock](#) ()
- void [init](#) (**BUInt** numChannels, **BUInt** numSamples)

Public Attributes

- [DataBlock](#) * [odataBlock](#)
- **BUInt32** [ofill](#)
- **BUInt** [oinUse](#)

7.11.1 Constructor & Destructor Documentation

7.11.1.1 BMeasureUnitsDataBlock()

```
BMeasureApi::BMeasureUnitsDataBlock::BMeasureUnitsDataBlock (
    BUInt numChannels = 0,
    BUInt numSamples = 0 )
```

7.11.1.2 ~BMeasureUnitsDataBlock()

```
BMeasureApi::BMeasureUnitsDataBlock::~~BMeasureUnitsDataBlock ( )
```

7.11.2 Member Function Documentation

7.11.2.1 init()

```
void BMeasureApi::BMeasureUnitsDataBlock::init (
    BUInt numChannels,
    BUInt numSamples )
```

7.11.3 Member Data Documentation

7.11.3.1 odataBlock

`DataBlock*` BMeasureApi::BMeasureUnitsDataBlock::odataBlock

7.11.3.2 ofill

`BUInt32` BMeasureApi::BMeasureUnitsDataBlock::ofill

7.11.3.3 oinUse

`BUInt` BMeasureApi::BMeasureUnitsDataBlock::oinUse

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

- [BMeasureUnits.h](#)
- [BMeasureUnits.cpp](#)

7.12 BMeasureApi::BoardConfig Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const `BObjMember` * [getMembers](#) ()

Public Attributes

- `BUInt32` [magic](#)
- `Version` [hardwareVersion](#)
- `BChar` [serialNumber](#) [12]
- `BTime` [buildTime](#)
- `BUInt8` [macAddress](#) [6]
- `BUInt8` [testMode](#)
- `BUInt8` [spare0](#)
- `BTime` [calibTime](#)
- `BFloat32` [calibTemp](#)
- `BFloat64` [calibDacOffsets](#) [2]
- `BFloat64` [calibDacScales](#) [2]
- `BFloat64` [calibAdcOffsets](#) [8]
- `BFloat64` [calibAdcScales](#) [8]
- `BFloat64` [calibAttenScales](#) [8]
- `BFloat64` [calibFiveVolts](#)
- `Version` [fpgaVersion](#)
- `Version` [wifiVersion](#)
- `BUInt32` [spare](#) [8]

7.12.1 Member Function Documentation

7.12.1.1 getMembers()

```
const BObjMember * BMeasureApi::BoardConfig::getMembers ( ) [static]
```

7.12.2 Member Data Documentation

7.12.2.1 buildTime

```
BTime BMeasureApi::BoardConfig::buildTime
```

7.12.2.2 calibAdcOffsets

```
BFloat64 BMeasureApi::BoardConfig::calibAdcOffsets[8]
```

7.12.2.3 calibAdcScales

```
BFloat64 BMeasureApi::BoardConfig::calibAdcScales[8]
```

7.12.2.4 calibAttenScales

```
BFloat64 BMeasureApi::BoardConfig::calibAttenScales[8]
```

7.12.2.5 calibDacOffsets

```
BFloat64 BMeasureApi::BoardConfig::calibDacOffsets[2]
```


7.12.2.6 calibDacScales

BFloat64 BMeasureApi::BoardConfig::calibDacScales[2]

7.12.2.7 calibFiveVolts

BFloat64 BMeasureApi::BoardConfig::calibFiveVolts

7.12.2.8 calibTemp

BFloat32 BMeasureApi::BoardConfig::calibTemp

7.12.2.9 calibTime

BTime BMeasureApi::BoardConfig::calibTime

7.12.2.10 fpgaVersion

Version BMeasureApi::BoardConfig::fpgaVersion

7.12.2.11 hardwareVersion

Version BMeasureApi::BoardConfig::hardwareVersion

7.12.2.12 macAddress

BUInt8 BMeasureApi::BoardConfig::macAddress[6]

7.12.2.13 magic

BUInt32 BMeasureApi::BoardConfig::magic

7.12.2.14 serialNumber

BChar BMeasureApi::BoardConfig::serialNumber[12]

7.12.2.15 spare

BUInt32 BMeasureApi::BoardConfig::spare[8]

7.12.2.16 spare0

BUInt8 BMeasureApi::BoardConfig::spare0

7.12.2.17 testMode

BUInt8 BMeasureApi::BoardConfig::testMode

7.12.2.18 wifiVersion

Version BMeasureApi::BoardConfig::wifiVersion

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.13 BMeasureApi::CalibrateInfo Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BUint32** [stage](#)
Stage to run.
- **BFloat64** [calibrateFrequency](#)
The Awg frequency for calibration.
- **BFloat64** [calibrateAmplitude](#)
The Awg amplitude for calibration.
- **BFloat64** [calibrateTime](#)
Number of seconds to calibrate over (synced to multiple AWG cycles)
- **BFloat64** [value](#)
Target/Set Value.

7.13.1 Member Function Documentation

7.13.1.1 getMembers()

```
const BObjMember * BMeasureApi::CalibrateInfo::getMembers ( ) [static]
```

7.13.2 Member Data Documentation

7.13.2.1 calibrateAmplitude

```
BFloat64 BMeasureApi::CalibrateInfo::calibrateAmplitude
```

The Awg amplitude for calibration.

7.13.2.2 calibrateFrequency

```
BFloat64 BMeasureApi::CalibrateInfo::calibrateFrequency
```

The Awg frequency for calibration.

7.13.2.3 calibrateTime

```
BFloat64 BMeasureApi::CalibrateInfo::calibrateTime
```

Number of seconds to calibrate over (synced to multiple AWG cycles)

7.13.2.4 stage

BUInt32 BMeasureApi::CalibrateInfo::stage

Stage to run.

7.13.2.5 value

BFloat64 BMeasureApi::CalibrateInfo::value

Target/Set Value.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.14 BMeasureApi::ChannelConfig Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BUInt8** [number](#)
The channel number.
- **BUInt8** [enabled](#)
Channel is enabled.
- **BUInt8** [attenuator](#)
Attenuator number in use.
- [ChannelType](#) [type](#)
The channel type.
- [SampleType](#) [sampleType](#)
The sample type.
- **BUInt8** [spare0](#) [3]
- **BUInt32** [dataChannel](#)
Data channel.
- **BChar** [id](#) [16]
- **BChar** [name](#) [16]
- **BChar** [siUnits](#) [8]
- **BFloat64** [calibOffset](#)
The calibration data offset.
- **BFloat64** [calibScale](#)

- The calibration data scale factor to volts.*
- **BFloat64** [calibScaleAtten1](#)
Attenuator 1 scaling.
- **BFloat64** [pgaGain](#)
The PGA gain.
- **BFloat64** [scale](#)
The user data scale factor.
- **BFloat64** [offset](#)
The user data offset.
- **BChar** [process](#) [32]

7.14.1 Member Function Documentation

7.14.1.1 getMembers()

```
const BObjMember * BMeasureApi::ChannelConfig::getMembers ( ) [static]
```

7.14.2 Member Data Documentation

7.14.2.1 attenuator

```
BUInt8 BMeasureApi::ChannelConfig::attenuator
```

Attenuator number in use.

7.14.2.2 calibOffset

```
BFloat64 BMeasureApi::ChannelConfig::calibOffset
```

The calibration data offset.

7.14.2.3 calibScale

```
BFloat64 BMeasureApi::ChannelConfig::calibScale
```

The calibration data scale factor to volts.

7.14.2.4 calibScaleAtten1

BFloat64 BMeasureApi::ChannelConfig::calibScaleAtten1

Attenuator 1 scaling.

7.14.2.5 dataChannel

BUInt32 BMeasureApi::ChannelConfig::dataChannel

Data channel.

7.14.2.6 enabled

BUInt8 BMeasureApi::ChannelConfig::enabled

Channel is enabled.

7.14.2.7 id

BChar BMeasureApi::ChannelConfig::id[16]

7.14.2.8 name

BChar BMeasureApi::ChannelConfig::name[16]

7.14.2.9 number

BUInt8 BMeasureApi::ChannelConfig::number

The channel number.

7.14.2.10 offset

BFloat64 BMeasureApi::ChannelConfig::offset

The user data offset.

7.14.2.11 pgaGain

BFloat64 BMeasureApi::ChannelConfig::pgaGain

The PGA gain.

7.14.2.12 process

BChar BMeasureApi::ChannelConfig::process[32]

7.14.2.13 sampleType

[SampleType](#) BMeasureApi::ChannelConfig::sampleType

The sample type.

7.14.2.14 scale

BFloat64 BMeasureApi::ChannelConfig::scale

The user data scale factor.

7.14.2.15 siUnits

BChar BMeasureApi::ChannelConfig::siUnits[8]

7.14.2.16 spare0

BUInt8 BMeasureApi::ChannelConfig::spare0[3]

7.14.2.17 type

`ChannelType` `BMeasureApi::ChannelConfig::type`

The channel type.

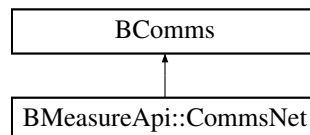
The documentation for this class was generated from the following files:

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.15 BMeasureApi::CommsNet Class Reference

```
#include <CommsNet.h>
```

Inheritance diagram for `BMeasureApi::CommsNet`:



Public Member Functions

- [CommsNet](#) (`BUInt` rxFifoSize=1024, `BUInt` txFifoSize=1024)
- [~CommsNet](#) ()
- [BError](#) [init](#) ()
- [BError](#) [connect](#) (`BString` host, `BUInt16` port)
- [BError](#) [disconnect](#) ()
- `BUInt` [readAvailable](#) ()
- [BError](#) [wait](#) (`BUInt32` eventSet, `BTimeout` timeout=-1, `BUInt32` num=1)
- [BError](#) [read](#) (void * `data`, `BUInt32` num, `BUInt32` &nt)
- `BUInt` [writeAvailable](#) ()
- [BError](#) [write](#) (const void * `data`, `BUInt32` nBytes, `BUInt32` &nt)
- [BError](#) [writeChunks](#) (const `BDataChunk` *chunks, `BUInt` nChunks, `BUInt32` &nt)

Protected Attributes

- `BSocket` [osocket](#)
- `Bool` [oinWait](#)
- `Bool` [oterminating](#)

Additional Inherited Members

7.15.1 Constructor & Destructor Documentation

7.15.1.1 CommsNet()

```
BMeasureApi::CommsNet::CommsNet (
    BUInt rxFifoSize = 1024,
    BUInt txFifoSize = 1024 )
```

7.15.1.2 ~CommsNet()

```
BMeasureApi::CommsNet::~CommsNet ( )
```

7.15.2 Member Function Documentation

7.15.2.1 connect()

```
BError BMeasureApi::CommsNet::connect (
    BString host,
    BUInt16 port )
```

7.15.2.2 disconnect()

```
BError BMeasureApi::CommsNet::disconnect ( ) [virtual]
```

Reimplemented from **BComms**.

7.15.2.3 init()

```
BError BMeasureApi::CommsNet::init ( ) [virtual]
```

Reimplemented from **BComms**.

7.15.2.4 read()

```
BError BMeasureApi::CommsNet::read (
    void * data,
    BUInt32 num,
    BUInt32 & nt ) [virtual]
```

Implements **BComms**.

7.15.2.5 readAvailable()

```
BUInt BMeasureApi::CommsNet::readAvailable ( ) [virtual]
```

Reimplemented from **BComms**.

7.15.2.6 wait()

```
BError BMeasureApi::CommsNet::wait (
    BUInt32 eventSet,
    BTimeout timeout = -1,
    BUInt32 num = 1 )
```

7.15.2.7 write()

```
BError BMeasureApi::CommsNet::write (
    const void * data,
    BUInt32 nBytes,
    BUInt32 & nt ) [virtual]
```

Implements **BComms**.

7.15.2.8 writeAvailable()

```
BUInt BMeasureApi::CommsNet::writeAvailable ( ) [virtual]
```

Reimplemented from **BComms**.

7.15.2.9 writeChunks()

```
BError BMeasureApi::CommsNet::writeChunks (
    const BDataChunk * chunks,
    BUInt nChunks,
    BUInt32 & nt ) [virtual]
```

Reimplemented from **BComms**.

7.15.3 Member Data Documentation

7.15.3.1 oinWait

Bool BMeasureApi::CommsNet::oinWait [protected]

7.15.3.2 osocket

BSocket BMeasureApi::CommsNet::osocket [protected]

7.15.3.3 oterminating

Bool BMeasureApi::CommsNet::oterminating [protected]

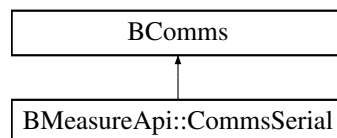
The documentation for this class was generated from the following files:

- [CommsNet.h](#)
- [CommsNet.cpp](#)

7.16 BMeasureApi::CommsSerial Class Reference

```
#include <CommsSerial.h>
```

Inheritance diagram for BMeasureApi::CommsSerial:



Public Member Functions

- [CommsSerial](#) ()
- [~CommsSerial](#) ()
- **BError** [connect](#) (**BString** device)
- **BError** [disconnect](#) ()
- **BUInt** [readAvailable](#) ()
- **BError** [read](#) (void * **data**, **BUInt32** num, **BUInt32** &nTrans)
- **BError** [write](#) (const void * **data**, **BUInt32** num, **BUInt32** &nTrans)
- **BError** [wait](#) (**BUInt32** eventSet, **BTimeout** timeout=-1, **BUInt32** num=1)

Private Attributes

- **BString** [odevice](#)
- int [oserialPort](#)

Additional Inherited Members

7.16.1 Constructor & Destructor Documentation

7.16.1.1 CommsSerial()

```
BMeasureApi::CommsSerial::CommsSerial ( )
```

7.16.1.2 ~CommsSerial()

```
BMeasureApi::CommsSerial::~~CommsSerial ( )
```

7.16.2 Member Function Documentation

7.16.2.1 connect()

```
BError BMeasureApi::CommsSerial::connect (
    BString device )
```

7.16.2.2 disconnect()

```
BError BMeasureApi::CommsSerial::disconnect ( ) [virtual]
```

Reimplemented from **BComms**.

7.16.2.3 read()

```
BError BMeasureApi::CommsSerial::read (
    void * data,
    BUInt32 num,
    BUInt32 & nTrans ) [virtual]
```

Implements **BComms**.

7.16.2.4 readAvailable()

```
BUInt BMeasureApi::CommsSerial::readAvailable ( ) [virtual]
```

Reimplemented from **BComms**.

7.16.2.5 wait()

```
BError BMeasureApi::CommsSerial::wait (
    BUInt32 eventSet,
    BTimeout timeout = -1,
    BUInt32 num = 1 )
```

7.16.2.6 write()

```
BError BMeasureApi::CommsSerial::write (
    const void * data,
    BUInt32 num,
    BUInt32 & nTrans ) [virtual]
```

Implements **BComms**.

7.16.3 Member Data Documentation

7.16.3.1 odevice

```
BString BMeasureApi::CommsSerial::odevice [private]
```

7.16.3.2 oserialPort

```
int BMeasureApi::CommsSerial::oserialPort [private]
```

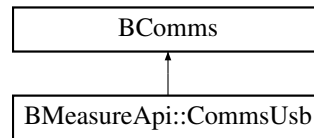
The documentation for this class was generated from the following file:

- [CommsSerial.h](#)

7.17 BMeasureApi::CommsUsb Class Reference

```
#include <CommsUsb.h>
```

Inheritance diagram for BMeasureApi::CommsUsb:



Public Member Functions

- [CommsUsb](#) ()
- [~CommsUsb](#) ()
- **BError** [connect](#) (**BString** device)
- **BError** [disconnect](#) ()
- **BUInt** [readAvailable](#) ()
- **BError** [read](#) (void * **data**, **BUInt32** num, **BUInt32** &nTrans)
- **BError** [write](#) (const void * **data**, **BUInt32** num, **BUInt32** &nTrans)
- **BError** [wait](#) (**BUInt32** eventSet, **BTimeout** timeout=-1, **BUInt32** num=1)

Private Member Functions

- **BError** [readChunk](#) ()

Private Attributes

- **BString** [odevice](#)
- libusb_context * [ocontext](#)
- libusb_device_handle * [odev](#)
- char [obuffer](#) [102400]
- **BUInt** [onum](#)
- **Bool** [oterminated](#)
- **Bool** [oterminating](#)

Additional Inherited Members

7.17.1 Constructor & Destructor Documentation

7.17.1.1 CommsUsb()

```
BMeasureApi::CommsUsb::CommsUsb ( )
```

7.17.1.2 ~CommsUsb()

BMeasureApi::CommsUsb::~~CommsUsb ()

7.17.2 Member Function Documentation

7.17.2.1 connect()

BError BMeasureApi::CommsUsb::connect (
 BString *device*)

7.17.2.2 disconnect()

BError BMeasureApi::CommsUsb::disconnect () [virtual]

Reimplemented from **BComms**.

7.17.2.3 read()

BError BMeasureApi::CommsUsb::read (
 void * *data*,
 BUInt32 *num*,
 BUInt32 & *nTrans*) [virtual]

Implements **BComms**.

7.17.2.4 readAvailable()

BUInt BMeasureApi::CommsUsb::readAvailable () [virtual]

Reimplemented from **BComms**.

7.17.2.5 readChunk()

BError BMeasureApi::CommsUsb::readChunk () [private]

7.17.2.6 wait()

```
BError BMeasureApi::CommsUsb::wait (
    BUInt32 eventSet,
    BTimeout timeout = -1,
    BUInt32 num = 1 )
```

7.17.2.7 write()

```
BError BMeasureApi::CommsUsb::write (
    const void * data,
    BUInt32 num,
    BUInt32 & nTrans ) [virtual]
```

Implements **BComms**.

7.17.3 Member Data Documentation

7.17.3.1 obuffer

```
char BMeasureApi::CommsUsb::obuffer[102400] [private]
```

7.17.3.2 ocontext

```
libusb_context* BMeasureApi::CommsUsb::ocontext [private]
```

7.17.3.3 odev

```
libusb_device_handle* BMeasureApi::CommsUsb::odev [private]
```

7.17.3.4 odevice

```
BString BMeasureApi::CommsUsb::odevice [private]
```


7.17.3.5 onum

BUInt BMeasureApi::CommsUsb::onum [private]

7.17.3.6 oterminated

Bool BMeasureApi::CommsUsb::oterminated [private]

7.17.3.7 oterminating

Bool BMeasureApi::CommsUsb::oterminating [private]

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

- [CommsUsb.h](#)
- [CommsUsb.cpp](#)

7.18 BMeasureApi::ConfigItem Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BChar** [name](#) [16]
- **BUInt8** [type](#)
The type of data.
- **BUInt8** [spare](#) [3]
- **BChar** [value](#) [16]

7.18.1 Member Function Documentation

7.18.1.1 getMembers()

```
const BObjMember * BMeasureApi::ConfigItem::getMembers ( ) [static]
```

7.18.2 Member Data Documentation

7.18.2.1 name

BChar BMeasureApi::ConfigItem::name[16]

7.18.2.2 spare

BUInt8 BMeasureApi::ConfigItem::spare[3]

7.18.2.3 type

BUInt8 BMeasureApi::ConfigItem::type

The type of data.

7.18.2.4 value

BChar BMeasureApi::ConfigItem::value[16]

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.19 BMeasureApi::Configuration Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BUInt32** [version](#)
The configuration version.
- **BChar** [name](#) [16]
- **BChar** [location](#) [16]
- **Mode** [mode](#)
The boot run mode.
- **SecurityMode** [securityMode](#)
The security mode.
- **BUInt8** [logData](#)
Log the data.
- **BUInt8** [logDataMode](#)
Log data mode.
- **BUInt8** [logDataDevice](#)
The device to store data.
- **BUInt8** [source](#)
The source number if multiple units are in use.
- **BUInt8** [sampleFrequencyMode](#)
The base sample frequency mode.
- **DigitalMode** [digitalMode](#)
The digital mode.
- **BUInt8** [digitalPins](#) [8]
- **NetworkMode** [networkMode](#)
The network mode (0 - off, 1 - dhcp, 2 - manual)
- **BUInt8** [spare1](#) [3]
- **BUInt32** [networkAddress](#)
The network IP address.
- **BUInt32** [networkMask](#)
The network netmask.
- **BUInt32** [networkGateway](#)
The network gateway.
- **BUInt32** [networkNameServer0](#)
The network nameserver.
- **BUInt32** [networkTimeServer](#)
The network timeserver.
- **Rs485Mode** [rs485Mode](#)
The RS485 mode.
- **BUInt8** [rs485Bits](#)
The RS485 number of bits.
- **BUInt8** [rs485StopBits](#)
The RS485 stop bits.
- **BUInt8** [spare2](#)
- **BUInt32** [rs485BaudRate](#)
The RS485 baud rate.
- **WifiMode** [wifiMode](#)
The wifi mode.
- **BUInt8** [spare3](#) [3]
- **BChar** [wifiAp0](#) [32]
- **BChar** [spare4](#) [32]
- **AlarmConfig** [alarms](#) [16]
- **EventMode** [mqttMode](#)

MQTT mode.

- **BUInt8** [spare5](#) [3]
- **BChar** [mqttServer](#) [32]
- **BUInt32** [mqttPort](#)

The MQTT port.

- [EventMode](#) [emailMode](#)

Email mode.

- **BUInt8** [spare6](#) [3]
- **BChar** [emailAddress](#) [32]
- **BChar** [program](#) [32]

7.19.1 Member Function Documentation

7.19.1.1 [getMembers\(\)](#)

```
const BObjMember * BMeasureApi::Configuration::getMembers ( ) [static]
```

7.19.2 Member Data Documentation

7.19.2.1 [alarms](#)

```
AlarmConfig BMeasureApi::Configuration::alarms[16]
```

7.19.2.2 [digitalMode](#)

```
DigitalMode BMeasureApi::Configuration::digitalMode
```

The digital mode.

7.19.2.3 [digitalPins](#)

```
BUInt8 BMeasureApi::Configuration::digitalPins[8]
```

7.19.2.4 emailAddress

BChar BMeasureApi::Configuration::emailAddress[32]

7.19.2.5 emailMode

EventMode BMeasureApi::Configuration::emailMode

Email mode.

7.19.2.6 location

BChar BMeasureApi::Configuration::location[16]

7.19.2.7 logData

BUInt8 BMeasureApi::Configuration::logData

Log the data.

7.19.2.8 logDataDevice

BUInt8 BMeasureApi::Configuration::logDataDevice

The device to store data.

7.19.2.9 logDataMode

BUInt8 BMeasureApi::Configuration::logDataMode

Log data mode.

7.19.2.10 mode

`Mode` `BMeasureApi::Configuration::mode`

The boot run mode.

7.19.2.11 mqttMode

`EventMode` `BMeasureApi::Configuration::mqttMode`

MQTT mode.

7.19.2.12 mqttPort

`BUInt32` `BMeasureApi::Configuration::mqttPort`

The MQTT port.

7.19.2.13 mqttServer

`BChar` `BMeasureApi::Configuration::mqttServer[32]`

7.19.2.14 name

`BChar` `BMeasureApi::Configuration::name[16]`

7.19.2.15 networkAddress

`BUInt32` `BMeasureApi::Configuration::networkAddress`

The network IP address.

7.19.2.16 networkGateway

BUInt32 BMeasureApi::Configuration::networkGateway

The network gateway.

7.19.2.17 networkMask

BUInt32 BMeasureApi::Configuration::networkMask

The network netmask.

7.19.2.18 networkMode

NetworkMode BMeasureApi::Configuration::networkMode

The network mode (0 - off, 1 - dhcp, 2 - manual)

7.19.2.19 networkNameServer0

BUInt32 BMeasureApi::Configuration::networkNameServer0

The network nameserver.

7.19.2.20 networkTimeServer

BUInt32 BMeasureApi::Configuration::networkTimeServer

The network timeserver.

7.19.2.21 program

BChar BMeasureApi::Configuration::program[32]

7.19.2.22 rs485BaudRate

BUInt32 BMeasureApi::Configuration::rs485BaudRate

The RS485 baud rate.

7.19.2.23 rs485Bits

BUInt8 BMeasureApi::Configuration::rs485Bits

The RS485 number of bits.

7.19.2.24 rs485Mode

[Rs485Mode](#) BMeasureApi::Configuration::rs485Mode

The RS485 mode.

7.19.2.25 rs485StopBits

BUInt8 BMeasureApi::Configuration::rs485StopBits

The RS485 stop bits.

7.19.2.26 sampleFrequencyMode

BUInt8 BMeasureApi::Configuration::sampleFrequencyMode

The base sample frequency mode.

7.19.2.27 securityMode

[SecurityMode](#) BMeasureApi::Configuration::securityMode

The security mode.

7.19.2.28 source

BUInt8 BMeasureApi::Configuration::source

The source number if multiple units are in use.

7.19.2.29 spare1

BUInt8 BMeasureApi::Configuration::spare1[3]

7.19.2.30 spare2

BUInt8 BMeasureApi::Configuration::spare2

7.19.2.31 spare3

BUInt8 BMeasureApi::Configuration::spare3[3]

7.19.2.32 spare4

BChar BMeasureApi::Configuration::spare4[32]

7.19.2.33 spare5

BUInt8 BMeasureApi::Configuration::spare5[3]

7.19.2.34 spare6

BUInt8 BMeasureApi::Configuration::spare6[3]

7.19.2.35 version

BUInt32 `BMeasureApi::Configuration::version`

The configuration version.

7.19.2.36 wifiAp0

BChar `BMeasureApi::Configuration::wifiAp0[32]`

7.19.2.37 wifiMode

WifiMode `BMeasureApi::Configuration::wifiMode`

The wifi mode.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.20 BMeasureApi::DataBlock Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BUInt64** [time](#)
The time in microseconds since 1970-01-01 to TAI.
- **BUInt16** [source](#)
The source unit,.
- **BUInt16** [status](#)
- **BUInt16** [numChannels](#)
The number of data channels.
- **BUInt16** [numSamples](#)
The number of samples.
- **BUInt32** [sequence](#)
The sequence number.
- **DataType** [type](#)
The type of data block.
- **BUInt8** [spare](#) [7]
- **BFloat32** [data](#) [117]

7.20.1 Member Function Documentation

7.20.1.1 getMembers()

```
const BObjMember * BMeasureApi::DataBlock::getMembers ( ) [static]
```

7.20.2 Member Data Documentation

7.20.2.1 data

```
BFloat32 BMeasureApi::DataBlock::data[117]
```

7.20.2.2 numChannels

```
BUInt16 BMeasureApi::DataBlock::numChannels
```

The number of data channels.

7.20.2.3 numSamples

```
BUInt16 BMeasureApi::DataBlock::numSamples
```

The number of samples.

7.20.2.4 sequence

```
BUInt32 BMeasureApi::DataBlock::sequence
```

The sequence number.

7.20.2.5 source

```
BUInt16 BMeasureApi::DataBlock::source
```

The source unit.

7.20.2.6 spare

```
BUInt8 BMeasureApi::DataBlock::spare[7]
```

7.20.2.7 status

```
BUInt16 BMeasureApi::DataBlock::status
```

7.20.2.8 time

```
BUInt64 BMeasureApi::DataBlock::time
```

The time in microseconds since 1970-01-01 to TAI.

7.20.2.9 type

```
DataType BMeasureApi::DataBlock::type
```

The type of data block.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.21 BMeasureApi::DataBlockProc Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BUInt64** [time](#)
The time in microseconds since 1970-01-01 to TAI.
- **BUInt16** [source](#)
The source unit.
- **BUInt16** [status](#)
- **BUInt16** [numChannels](#)
The number of data channels.
- **BUInt16** [numSamples](#)
The number of samples.
- **BUInt32** [sequence](#)
The sequence number.
- [DataType](#) [type](#)
The type of data block.
- **BUInt8** [spare](#) [7]
- [DataProc](#) [analogueData](#) [8]
- **BUInt32** [digitalData](#)
Digital channel data.

7.21.1 Member Function Documentation

7.21.1.1 [getMembers\(\)](#)

```
const BObjMember * BMeasureApi::DataBlockProc::getMembers ( ) [static]
```

7.21.2 Member Data Documentation

7.21.2.1 [analogueData](#)

```
DataProc BMeasureApi::DataBlockProc::analogueData [8]
```

7.21.2.2 [digitalData](#)

```
BUInt32 BMeasureApi::DataBlockProc::digitalData
```

Digital channel data.

7.21.2.3 numChannels

BUInt16 BMeasureApi::DataBlockProc::numChannels

The number of data channels.

7.21.2.4 numSamples

BUInt16 BMeasureApi::DataBlockProc::numSamples

The number of samples.

7.21.2.5 sequence

BUInt32 BMeasureApi::DataBlockProc::sequence

The sequence number.

7.21.2.6 source

BUInt16 BMeasureApi::DataBlockProc::source

The source unit.

7.21.2.7 spare

BUInt8 BMeasureApi::DataBlockProc::spare[7]

7.21.2.8 status

BUInt16 BMeasureApi::DataBlockProc::status

7.21.2.9 time

BUInt64 BMeasureApi::DataBlockProc::time

The time in microseconds since 1970-01-01 to TAI.

7.21.2.10 type

DataType BMeasureApi::DataBlockProc::type

The type of data block.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.22 BMeasureApi::DataFile Class Reference

```
#include <DataFile.h>
```

Public Member Functions

- [DataFile](#) ()
- [~DataFile](#) ()
- void [init](#) ()
Initialise.
- **BError** [open](#) (**BString** fileName, **BString** mode, **BString** format="")
Open the file for read or write.
- **BError** [close](#) ()
Close the file.
- **BString** [getFileName](#) ()
Return the file name.
- **BError** [writeInfo](#) (const [InfoBlock](#) &infoBlock, const [ChannelConfigs](#) &channels)
- **BError** [writeData](#) ([DataBlock](#) * data)
Write a block of data.
- **BError** [writeData](#) ([DataBlockProc](#) * data)
Write a block of processed data.
- **BError** [writeEnd](#) ()
- **BError** [readInfo](#) (**BString** &format, [InfoBlock](#) &infoBlock, [ChannelConfigs](#) &channels)
- **BError** [readData](#) ([DataBlock](#) * data)
Read a block of data.

Private Member Functions

- **BError** `validateFormat` (**BString** format)
- **BError** `writeInfoCsv` (const **InfoBlock** &infoBlock, const **ChannelConfigs** &channels)
- **BError** `writeInfoTdms` (const **InfoBlock** &infoBlock, const **ChannelConfigs** &channels)
- **BError** `writeInfoBMeas` (const **InfoBlock** &infoBlock, const **ChannelConfigs** &channels)

Private Attributes

- **BString** `ofilename`
- **BString** `omode`
- **BString** `oformat`
- **BFile** `ofile`
- **BUint32** `opacketLen`
- **BoapMc1Packet** * `opacket`

7.22.1 Constructor & Destructor Documentation

7.22.1.1 `DataFile()`

```
BMeasureApi::DataFile::DataFile ( )
```

7.22.1.2 `~DataFile()`

```
BMeasureApi::DataFile::~~DataFile ( )
```

7.22.2 Member Function Documentation

7.22.2.1 `close()`

```
BError BMeasureApi::DataFile::close ( )
```

Close the file.

7.22.2.2 getFileName()

```
BString BMeasureApi::DataFile::getFileName ( )
```

Return the file name.

7.22.2.3 init()

```
void BMeasureApi::DataFile::init ( )
```

Initialise.

7.22.2.4 open()

```
BError BMeasureApi::DataFile::open (
    BString fileName,
    BString mode,
    BString format = "" )
```

Open the file for read or write.

7.22.2.5 readData()

```
BError BMeasureApi::DataFile::readData (
    DataBlock * data )
```

Read a block of data.

7.22.2.6 readInfo()

```
BError BMeasureApi::DataFile::readInfo (
    BString & format,
    InfoBlock & infoBlock,
    ChannelConfigs & channels )
```

7.22.2.7 validateFormat()

```
BError BMeasureApi::DataFile::validateFormat (
    BString format ) [private]
```

7.22.2.8 writeData() [1/2]

```
BError BMeasureApi::DataFile::writeData (  
    DataBlock * data )
```

Write a block of data.

7.22.2.9 writeData() [2/2]

```
BError BMeasureApi::DataFile::writeData (  
    DataBlockProc * data )
```

Write a block of processed data.

7.22.2.10 writeEnd()

```
BError BMeasureApi::DataFile::writeEnd ( )
```

7.22.2.11 writeInfo()

```
BError BMeasureApi::DataFile::writeInfo (  
    const InfoBlock & infoBlock,  
    const ChannelConfigs & channels )
```

7.22.2.12 writeInfoBMeas()

```
BError BMeasureApi::DataFile::writeInfoBMeas (  
    const InfoBlock & infoBlock,  
    const ChannelConfigs & channels ) [private]
```

7.22.2.13 writeInfoCsv()

```
BError BMeasureApi::DataFile::writeInfoCsv (  
    const InfoBlock & infoBlock,  
    const ChannelConfigs & channels ) [private]
```

7.22.2.14 writeInfoTdms()

```
BError BMeasureApi::DataFile::writeInfoTdms (
    const InfoBlock & infoBlock,
    const ChannelConfigs & channels ) [private]
```

7.22.3 Member Data Documentation

7.22.3.1 ofile

```
BFile BMeasureApi::DataFile::ofile [private]
```

7.22.3.2 ofileName

```
BString BMeasureApi::DataFile::ofileName [private]
```

7.22.3.3 oformat

```
BString BMeasureApi::DataFile::oformat [private]
```

7.22.3.4 omode

```
BString BMeasureApi::DataFile::omode [private]
```

7.22.3.5 opacket

```
BoapMc1Packet* BMeasureApi::DataFile::opacket [private]
```

7.22.3.6 opacketLen

```
BUInt32 BMeasureApi::DataFile::opacketLen [private]
```

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

- [DataFile.h](#)
- [DataFile.cpp](#)

7.23 BMeasureApi::DataProc Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BFloat32** [mean](#)
Processed data mean.
- **BFloat32** [rms](#)
Processed data RMS.
- **BFloat32** [peakHigh](#)
Processed data highest peak.
- **BFloat32** [peakLow](#)
Processed data lowest peak.
- **Bool** [alarm](#)
An alarm condition on this channel.
- **BUInt8** [spare](#) [3]

7.23.1 Member Function Documentation

7.23.1.1 [getMembers\(\)](#)

```
const BObjMember * BMeasureApi::DataProc::getMembers ( ) [static]
```

7.23.2 Member Data Documentation

7.23.2.1 [alarm](#)

```
Bool BMeasureApi::DataProc::alarm
```

An alarm condition on this channel.

7.23.2.2 mean

```
BFloat32 BMeasureApi::DataProc::mean
```

Processed data mean.

7.23.2.3 peakHigh

```
BFloat32 BMeasureApi::DataProc::peakHigh
```

Processed data highest peak.

7.23.2.4 peakLow

```
BFloat32 BMeasureApi::DataProc::peakLow
```

Processed data lowest peak.

7.23.2.5 rms

```
BFloat32 BMeasureApi::DataProc::rms
```

Processed data RMS.

7.23.2.6 spare

```
BUInt8 BMeasureApi::DataProc::spare[3]
```

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.24 Dfu Class Reference

The [Dfu](#) access class.

```
#include <Dfu.h>
```

Public Member Functions

- [Dfu](#) ()
- [~Dfu](#) ()
- **BError** [init](#) (**Bool** verbose)
 - Initialise.*
- **BError** [detectDevice](#) ()
 - Check if DFU devuce exists.*
- **BError** [validateFile](#) (**BString** filename, **BUInt** type, **BString** &version)
 - Check if file is valid firmware.*
- **BError** [connect](#) ()
 - Connect to USB DFU device.*
- **BError** [disconnect](#) ()
 - Disconnect from USB DFU device.*
- **BError** [reset](#) ()
 - Reset.*
- **BError** [clearStatus](#) ()
- **BError** [getStatus](#) ([DfuStatus](#) &status)
- **BError** [upload](#) (**BString** filename, **BUInt** type)
 - Upload a file.*
- **BError** [upload_cmd](#) (**BUInt8** cmd, **BUInt32** address)

Private Attributes

- **Bool** [overbose](#)
- **Bool** [oconnected](#)
- `libusb_context *` [ocontext](#)
- `libusb_device_handle *` [odev](#)

7.24.1 Detailed Description

The [Dfu](#) access class.

7.24.2 Constructor & Destructor Documentation

7.24.2.1 [Dfu](#)()

```
Dfu::Dfu ( )
```

7.24.2.2 [~Dfu](#)()

```
Dfu::~Dfu ( )
```

7.24.3 Member Function Documentation

7.24.3.1 clearStatus()

```
BError Dfu::clearStatus ( )
```

7.24.3.2 connect()

```
BError Dfu::connect ( )
```

Connect to USB DFU device.

7.24.3.3 detectDevice()

```
BError Dfu::detectDevice ( )
```

Check if DFU devuce exists.

7.24.3.4 disconnect()

```
BError Dfu::disconnect ( )
```

Disconnect from USB DFU device.

7.24.3.5 getStatus()

```
BError Dfu::getStatus (
    DfuStatus & status )
```

7.24.3.6 init()

```
BError Dfu::init (
    Bool verbose )
```

Initialise.

7.24.3.7 reset()

```
BError Dfu::reset ( )
```

Reset.

7.24.3.8 upload()

```
BError Dfu::upload (
    BString filename,
    BUInt type )
```

Upload a file.

7.24.3.9 upload_cmd()

```
BError Dfu::upload_cmd (
    BUInt8 cmd,
    BUInt32 address )
```

7.24.3.10 validateFile()

```
BError Dfu::validateFile (
    BString filename,
    BUInt type,
    BString & version )
```

Check if file is valid firmware.

7.24.4 Member Data Documentation

7.24.4.1 oconnected

```
Bool Dfu::occonnected [private]
```


7.24.4.2 ocontext

```
libusb_context* Dfu::ocontext [private]
```

7.24.4.3 odev

```
libusb_device_handle* Dfu::odev [private]
```

7.24.4.4 overbose

```
bool Dfu::overbose [private]
```

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

- [Dfu.h](#)
- [Dfu.cpp](#)

7.25 DfuStatus Struct Reference

```
#include <Dfu.h>
```

Public Attributes

- **BUInt8** [status](#)
- **BUInt** [pollTimeout](#)
- **BUInt8** [state](#)
- **BUInt8** [iString](#)

7.25.1 Member Data Documentation

7.25.1.1 iString

```
BUInt8 DfuStatus::iString
```

7.25.1.2 pollTimeout

BUInt DfuStatus::pollTimeout

7.25.1.3 state

BUInt8 DfuStatus::state

7.25.1.4 status

BUInt8 DfuStatus::status

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

- [Dfu.h](#)

7.26 BMeasureApi::FileData Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BUInt32** [length](#)
The data length.
- **BUInt8** [data](#) [512]

7.26.1 Member Function Documentation

7.26.1.1 getMembers()

```
const BObjMember * BMeasureApi::FileData::getMembers ( ) [static]
```

7.26.2 Member Data Documentation

7.26.2.1 data

BUInt8 BMeasureApi::FileData::data[512]

7.26.2.2 length

BUInt32 BMeasureApi::FileData::length

The data length.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.27 BMeasureApi::FileInfo Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BChar** [name](#) [128]
- **BTime** [time](#)
The file date/time.
- **FileType** [fileType](#)
The file type.
- **BUInt8** [spare](#) [3]
- **BUInt64** [fileLength](#)
The file length.

7.27.1 Member Function Documentation

7.27.1.1 getMembers()

```
const BObjMember * BMeasureApi::FileInfo::getMembers ( ) [static]
```

7.27.2 Member Data Documentation

7.27.2.1 fileLength

```
BUInt64 BMeasureApi::FileInfo::fileLength
```

The file length.

7.27.2.2 fileType

```
FileType BMeasureApi::FileInfo::fileType
```

The file type.

7.27.2.3 name

```
BChar BMeasureApi::FileInfo::name[128]
```

7.27.2.4 spare

```
BUInt8 BMeasureApi::FileInfo::spare[3]
```

7.27.2.5 time

```
BTime BMeasureApi::FileInfo::time
```

The file date/time.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.28 BMeasureApi::FilesysInfo Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BChar** [name](#) [128]
- **BUInt64** [size](#)
The store size.
- **BUInt64** [free](#)
The store free space.

7.28.1 Member Function Documentation

7.28.1.1 [getMembers\(\)](#)

```
const BObjMember * BMeasureApi::FilesysInfo::getMembers ( ) [static]
```

7.28.2 Member Data Documentation

7.28.2.1 [free](#)

```
BUInt64 BMeasureApi::FilesysInfo::free
```

The store free space.

7.28.2.2 [name](#)

```
BChar BMeasureApi::FilesysInfo::name[128]
```

7.28.2.3 size

BUInt64 BMeasureApi::FilesysInfo::size

The store size.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.29 BMeasureApi::InfoBlock Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BUInt64** [time](#)
The time in microseconds since 1970-01-01 to TAI.
- **BUInt16** [source](#)
The source unit.
- **BUInt16** [numChannels](#)
The number of data channels.
- **BUInt16** [version](#)
The info/data version.
- **BMeasFileType** [fileType](#)
The file structure type.
- **DataType** [dataType](#)
The data type file contents.
- **BChar** [name](#) [16]
- **BChar** [location](#) [16]
- **NodeInfo** [nodeInfo](#)
Information on the unit.
- **MeasurementConfig** [measureConfig](#)
The measurement configuration.

7.29.1 Member Function Documentation

7.29.1.1 getMembers()

```
const BObjMember * BMeasureApi::InfoBlock::getMembers ( ) [static]
```

7.29.2 Member Data Documentation

7.29.2.1 dataType

[DataType](#) BMeasureApi::InfoBlock::dataType

The data type file contents.

7.29.2.2 fileType

[BMeasFileType](#) BMeasureApi::InfoBlock::fileType

The file structure type.

7.29.2.3 location

BChar BMeasureApi::InfoBlock::location[16]

7.29.2.4 measureConfig

[MeasurementConfig](#) BMeasureApi::InfoBlock::measureConfig

The measurement configuration.

7.29.2.5 name

BChar BMeasureApi::InfoBlock::name[16]

7.29.2.6 nodeInfo

[NodeInfo](#) BMeasureApi::InfoBlock::nodeInfo

[Information](#) on the unit.

7.29.2.7 numChannels

BUInt16 BMeasureApi::InfoBlock::numChannels

The number of data channels.

7.29.2.8 source

BUInt16 BMeasureApi::InfoBlock::source

The source unit.

7.29.2.9 time

BUInt64 BMeasureApi::InfoBlock::time

The time in microseconds since 1970-01-01 to TAI.

7.29.2.10 version

BUInt16 BMeasureApi::InfoBlock::version

The info/data version.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.30 BMeasureApi::Information Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- [NodeInfo](#) `nodeInfo`
- **BUInt8** `numConfigItems`
The number of config items.
- **BUInt8** `numChannels`
The number of channels.
- **BUInt8** `spare0` [2]
- **BTimeUs** `time`
The system time.
- **BUInt32** `networkMode`
The network Mode.
- **BUInt8** `networkMacAddress` [6]
- **BUInt8** `spare1` [2]
- **BUInt32** `networkAddress`
The network IP address.
- **BUInt32** `networkMask`
The network netmask.
- **BUInt32** `networkGateway`
The network gateway.
- **BUInt32** `networkNameServer0`
The network nameserver.
- **BUInt32** `networkTimeServer`
The network time server.
- **BUInt32** `wifiMode`
The Wifi mode.
- **BUInt8** `wifiMacAddress` [6]
- **BUInt8** `spare2` [2]
- **BUInt32** `wifiAddress`
The Wifi IP address.
- **BUInt32** `wifiMask`
The Wifi netmask.
- **BUInt32** `wifiGateway`
The Wifi gateway.
- **BTime** `calibTime`
The last calibration time.
- **BUInt8** `spare3` [28]

7.30.1 Member Function Documentation

7.30.1.1 `getMembers()`

```
const BObjMember * BMeasureApi::Information::getMembers ( ) [static]
```

7.30.2 Member Data Documentation

7.30.2.1 calibTime

BTime BMeasureApi::Information::calibTime

The last calibration time.

7.30.2.2 networkAddress

BUInt32 BMeasureApi::Information::networkAddress

The network IP address.

7.30.2.3 networkGateway

BUInt32 BMeasureApi::Information::networkGateway

The network gateway.

7.30.2.4 networkMacAddress

BUInt8 BMeasureApi::Information::networkMacAddress[6]

7.30.2.5 networkMask

BUInt32 BMeasureApi::Information::networkMask

The network netmask.

7.30.2.6 networkMode

BUInt32 BMeasureApi::Information::networkMode

The network Mode.

7.30.2.7 networkNameServer0

BUInt32 BMeasureApi::Information::networkNameServer0

The network nameserver.

7.30.2.8 networkTimeServer

BUInt32 BMeasureApi::Information::networkTimeServer

The network time server.

7.30.2.9 nodeInfo

[NodeInfo](#) BMeasureApi::Information::nodeInfo

7.30.2.10 numChannels

BUInt8 BMeasureApi::Information::numChannels

The number of channels.

7.30.2.11 numConfigItems

BUInt8 BMeasureApi::Information::numConfigItems

The number of config items.

7.30.2.12 spare0

BUInt8 BMeasureApi::Information::spare0[2]

7.30.2.13 spare1

BUInt8 BMeasureApi::Information::spare1[2]

7.30.2.14 spare2

BUInt8 BMeasureApi::Information::spare2[2]

7.30.2.15 spare3

BUInt8 BMeasureApi::Information::spare3[28]

7.30.2.16 time

BTimeUs BMeasureApi::Information::time

The system time.

7.30.2.17 wifiAddress

BUInt32 BMeasureApi::Information::wifiAddress

The Wifi IP address.

7.30.2.18 wifiGateway

BUInt32 BMeasureApi::Information::wifiGateway

The Wifi gateway.

7.30.2.19 wifiMacAddress

BUInt8 BMeasureApi::Information::wifiMacAddress[6]

7.30.2.20 wifiMask

BUInt32 BMeasureApi::Information::wifiMask

The Wifi netmask.

7.30.2.21 wifiMode

BUInt32 BMeasureApi::Information::wifiMode

The Wifi mode.

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.31 BMeasureApi::MeasurementConfig Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- [MeasureMode](#) `measureMode`
- **BUInt8** `measureOptions`
Measure option bit set.
- **BUInt8** `peakFilter`
Peak filtering number of samples.
- **BUInt8** `spare1`
- [TriggerMode](#) `triggerMode`
- [TriggerConfig](#) `triggerConfig`
Trigger config including direction, filters etc.
- **BUInt8** `triggerChannel`
- **BUInt8** `spare2`
- **BFloat64** `triggerLevel`
- **BInt32** `triggerDelay`
Trigger delay in samples.
- **BFloat64** `sampleRate`
- **BUInt32** `numSamples0`
The number of samples in a chunk for display and/or repeat.
- **BUInt32** `numSamples1`
The number of samples per each data processing cycle. 0 disables this processing.
- **BUInt32** `numSamples2`
The number of samples to capture. 0 is continuous.
- **BUInt32** `measurePeriod`
Time in seconds between measurement sample bursts. 0 is continuous.
- **BUInt32** `numSamplesBlock`
The number of samples per block. 0 is default setting.
- **BChar** `description` [64]

7.31.1 Member Function Documentation

7.31.1.1 getMembers()

```
const BObjMember * BMeasureApi::MeasurementConfig::getMembers ( ) [static]
```

7.31.2 Member Data Documentation

7.31.2.1 description

```
BChar BMeasureApi::MeasurementConfig::description[64]
```

7.31.2.2 measureMode

```
MeasureMode BMeasureApi::MeasurementConfig::measureMode
```

7.31.2.3 measureOptions

```
BUInt8 BMeasureApi::MeasurementConfig::measureOptions
```

Measure option bit set.

7.31.2.4 measurePeriod

```
BUInt32 BMeasureApi::MeasurementConfig::measurePeriod
```

Time in seconds between measurement sample bursts. 0 is continuous.

7.31.2.5 numSamples0

```
BUInt32 BMeasureApi::MeasurementConfig::numSamples0
```

The number of samples in a chunk for display and/or repeat.

7.31.2.6 numSamples1

BUInt32 BMeasureApi::MeasurementConfig::numSamples1

The number of samples per each data processing cycle. 0 disables this processing.

7.31.2.7 numSamples2

BUInt32 BMeasureApi::MeasurementConfig::numSamples2

The number of samples to capture. 0 is continuous.

7.31.2.8 numSamplesBlock

BUInt32 BMeasureApi::MeasurementConfig::numSamplesBlock

The number of samples per block. 0 is default setting.

7.31.2.9 peakFilter

BUInt8 BMeasureApi::MeasurementConfig::peakFilter

Peak filtering number of samples.

7.31.2.10 sampleRate

BFloat64 BMeasureApi::MeasurementConfig::sampleRate

7.31.2.11 spare1

BUInt8 BMeasureApi::MeasurementConfig::spare1

7.31.2.12 spare2

BUInt8 BMeasureApi::MeasurementConfig::spare2

7.31.2.13 triggerChannel

BUInt8 BMeasureApi::MeasurementConfig::triggerChannel

7.31.2.14 triggerConfig

[TriggerConfig](#) BMeasureApi::MeasurementConfig::triggerConfig

Trigger config including direction, filters etc.

7.31.2.15 triggerDelay

BInt32 BMeasureApi::MeasurementConfig::triggerDelay

Trigger delay in samples.

7.31.2.16 triggerLevel

BFloat64 BMeasureApi::MeasurementConfig::triggerLevel

7.31.2.17 triggerMode

[TriggerMode](#) BMeasureApi::MeasurementConfig::triggerMode

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.32 BMeasureApi::NodeInfo Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- [BUInt32](#) `apiVersion`
- [Version](#) `hardwareVersion`
- [Version](#) `fpgaVersion`
- [Version](#) `wifiVersion`
- [Version](#) `softwareVersion`
- [BChar](#) `variant` [12]
- [BChar](#) `serialNumber` [12]
- [SecurityMode](#) `securityMode`
- [BUInt8](#) `spare` [7]

7.32.1 Member Function Documentation

7.32.1.1 `getMembers()`

```
const BObjMember * BMeasureApi::NodeInfo::getMembers ( ) [static]
```

7.32.2 Member Data Documentation

7.32.2.1 `apiVersion`

```
BUInt32 BMeasureApi::NodeInfo::apiVersion
```

7.32.2.2 `fpgaVersion`

```
Version BMeasureApi::NodeInfo::fpgaVersion
```

7.32.2.3 `hardwareVersion`

```
Version BMeasureApi::NodeInfo::hardwareVersion
```

7.32.2.4 `securityMode`

```
SecurityMode BMeasureApi::NodeInfo::securityMode
```

7.32.2.5 serialNumber

BChar BMeasureApi::NodeInfo::serialNumber[12]

7.32.2.6 softwareVersion

Version BMeasureApi::NodeInfo::softwareVersion

7.32.2.7 spare

BUInt8 BMeasureApi::NodeInfo::spare[7]

7.32.2.8 variant

BChar BMeasureApi::NodeInfo::variant[12]

7.32.2.9 wifiVersion

Version BMeasureApi::NodeInfo::wifiVersion

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.33 BMeasureApi::NodeStatus Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- [BTimeUs](#) `time`
- [BUInt32](#) `status`
- [BUInt32](#) `error`
- [BChar](#) `errorStr` [32]
- [Mode](#) `mode`
- [BUInt8](#) `ethernetStatus`
- [BUInt8](#) `wifiStatus`
- [BUInt8](#) `spare` [9]

7.33.1 Member Function Documentation

7.33.1.1 `getMembers()`

```
const BObjMember * BMeasureApi::NodeStatus::getMembers ( ) [static]
```

7.33.2 Member Data Documentation

7.33.2.1 `error`

```
BUInt32 BMeasureApi::NodeStatus::error
```

7.33.2.2 `errorStr`

```
BChar BMeasureApi::NodeStatus::errorStr[32]
```

7.33.2.3 `ethernetStatus`

```
BUInt8 BMeasureApi::NodeStatus::ethernetStatus
```

7.33.2.4 `mode`

```
Mode BMeasureApi::NodeStatus::mode
```

7.33.2.5 spare

BUInt8 BMeasureApi::NodeStatus::spare[9]

7.33.2.6 status

BUInt32 BMeasureApi::NodeStatus::status

7.33.2.7 time

BTimeUs BMeasureApi::NodeStatus::time

7.33.2.8 wifiStatus

BUInt8 BMeasureApi::NodeStatus::wifiStatus

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

7.34 BMeasureApi::Version Class Reference

```
#include <BMeasureD.h>
```

Static Public Member Functions

- static const **BObjMember** * [getMembers](#) ()

Public Attributes

- **BUInt8** [type](#)
- **BUInt8** [ver0](#)
- **BUInt8** [ver1](#)
- **BUInt8** [ver2](#)

7.34.1 Member Function Documentation

7.34.1.1 getMembers()

```
const BObjMember * BMeasureApi::Version::getMembers ( ) [static]
```

7.34.2 Member Data Documentation

7.34.2.1 type

```
BUInt8 BMeasureApi::Version::type
```

7.34.2.2 ver0

```
BUInt8 BMeasureApi::Version::ver0
```

7.34.2.3 ver1

```
BUInt8 BMeasureApi::Version::ver1
```

7.34.2.4 ver2

```
BUInt8 BMeasureApi::Version::ver2
```

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

- [BMeasureD.h](#)
- [BMeasureD.cpp](#)

Chapter 8

File Documentation

8.1 BMdns.cpp File Reference

```
#include <BMdns.h>
#include <BDebug.h>
#include <stdio.h>
#include <errno.h>
#include <sys/ioctl.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <net/if.h>
```

Macros

- `#define BDEBUGL1 0`

Enumerations

- enum `MdnsRecordType` {
 `MDNS_RECORDTYPE_IGNORE` = 0, `MDNS_RECORDTYPE_A` = 1, `MDNS_RECORDTYPE_PTR` = 12,
 `MDNS_RECORDTYPE_TXT` = 16,
 `MDNS_RECORDTYPE_AAAA` = 28, `MDNS_RECORDTYPE_SRV` = 33 }
- enum `MdnsEntryType` { `MDNS_ENTRYTYPE_ANSWER` = 1, `MDNS_ENTRYTYPE_AUTHORITY` = 2,
 `MDNS_ENTRYTYPE_ADDITIONAL` = 3 }
- enum `MdnsClass` { `MDNS_CLASS_IN` = 1 }

Functions

- static int `mdns_write_string` (`BUInt8` *buffer, `BUInt8` *p, `BString` str)
- static int `mdns_read_string` (void *buffer, `BUInt8` *p, `BString` &str)
- static int `mdns_read_strings` (void *buffer, `BUInt8` *p, `BString` &str)

8.1.1 Macro Definition Documentation

8.1.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.1.2 Enumeration Type Documentation

8.1.2.1 MdnsClass

```
enum MdnsClass
```

Enumerator

MDNS_CLASS_IN	
---------------	--

8.1.2.2 MdnsEntryType

```
enum MdnsEntryType
```

Enumerator

MDNS_ENTRYTYPE_ANSWER	
MDNS_ENTRYTYPE_AUTHORITY	
MDNS_ENTRYTYPE_ADDITIONAL	

8.1.2.3 MdnsRecordType

```
enum MdnsRecordType
```

Enumerator

MDNS_RECORDTYPE_IGNORE	
MDNS_RECORDTYPE_A	
MDNS_RECORDTYPE_PTR	
MDNS_RECORDTYPE_TXT	
MDNS_RECORDTYPE_AAAA	
MDNS_RECORDTYPE_SRV	

8.1.3 Function Documentation

8.1.3.1 mdns_read_string()

```
static int mdns_read_string (  
    void * buffer,  
    BUInt8 * p,  
    BString & str ) [static]
```

8.1.3.2 mdns_read_strings()

```
static int mdns_read_strings (  
    void * buffer,  
    BUInt8 * p,  
    BString & str ) [static]
```

8.1.3.3 mdns_write_string()

```
static int mdns_write_string (  
    BUInt8 * buffer,  
    BUInt8 * p,  
    BString str ) [static]
```

8.2 BMdns.h File Reference

```
#include <BSocket.h>
```

Classes

- class [BMdnsService](#)
- class [BMdns](#)

8.3 BMeasureB-1.cpp File Reference

```
#include <BMeasureB.h>  
#include <string.h>
```

Namespaces

- [BMeasureApi](#)

8.4 BMeasureB.cpp File Reference

```
#include <BMeasureB.h>
#include <string.h>
```

Namespaces

- [BMeasureApi](#)

8.5 BMeasureB.h File Reference

```
#include <BTypes.h>
#include <BComplex.h>
#include <BoapMcl.h>
#include <BMeasureD.h>
```

Classes

- class [BMeasureApi::BMeasure](#)

Namespaces

- [BMeasureApi](#)

Variables

- const **BUInt32** [BMeasureApi::apiVersion](#) = 0

8.6 BMeasureD.cpp File Reference

```
#include <BMeasureD.h>
```

Namespaces

- [BMeasureApi](#)

Macros

- #define `boffsetof(T, F) ((BUInt)((char*)&((T*)0L)->F - (char*)0L))`

Functions

- `BString BMeasureApi::toBString (ErrorNum v)`
- `BError BMeasureApi::fromBString (BString str, ErrorNum &v)`
- `BString BMeasureApi::toBStringJson (BString n, ErrorNum v)`
- `BString BMeasureApi::toBString (NodeType v)`
- `BError BMeasureApi::fromBString (BString str, NodeType &v)`
- `BString BMeasureApi::toBStringJson (BString n, NodeType v)`
- `BString BMeasureApi::toBString (SecurityMode v)`
- `BError BMeasureApi::fromBString (BString str, SecurityMode &v)`
- `BString BMeasureApi::toBStringJson (BString n, SecurityMode v)`
- `BString BMeasureApi::toBString (Status v)`
- `BError BMeasureApi::fromBString (BString str, Status &v)`
- `BString BMeasureApi::toBStringJson (BString n, Status v)`
- `BString BMeasureApi::toBString (Mode v)`
- `BError BMeasureApi::fromBString (BString str, Mode &v)`
- `BString BMeasureApi::toBStringJson (BString n, Mode v)`
- `BString BMeasureApi::toBString (BlockTypes v)`
- `BError BMeasureApi::fromBString (BString str, BlockTypes &v)`
- `BString BMeasureApi::toBStringJson (BString n, BlockTypes v)`
- `BString BMeasureApi::toBString (ChannelType v)`
- `BError BMeasureApi::fromBString (BString str, ChannelType &v)`
- `BString BMeasureApi::toBStringJson (BString n, ChannelType v)`
- `BString BMeasureApi::toBString (SampleType v)`
- `BError BMeasureApi::fromBString (BString str, SampleType &v)`
- `BString BMeasureApi::toBStringJson (BString n, SampleType v)`
- `BString BMeasureApi::toBString (SyncMode v)`
- `BError BMeasureApi::fromBString (BString str, SyncMode &v)`
- `BString BMeasureApi::toBStringJson (BString n, SyncMode v)`
- `BString BMeasureApi::toBString (MeasureMode v)`
- `BError BMeasureApi::fromBString (BString str, MeasureMode &v)`
- `BString BMeasureApi::toBStringJson (BString n, MeasureMode v)`
- `BString BMeasureApi::toBString (MeasureOption v)`
- `BError BMeasureApi::fromBString (BString str, MeasureOption &v)`
- `BString BMeasureApi::toBStringJson (BString n, MeasureOption v)`
- `BString BMeasureApi::toBString (TriggerMode v)`
- `BError BMeasureApi::fromBString (BString str, TriggerMode &v)`
- `BString BMeasureApi::toBStringJson (BString n, TriggerMode v)`
- `BString BMeasureApi::toBString (TriggerConfig v)`
- `BError BMeasureApi::fromBString (BString str, TriggerConfig &v)`
- `BString BMeasureApi::toBStringJson (BString n, TriggerConfig v)`
- `BString BMeasureApi::toBString (DigitalMode v)`
- `BError BMeasureApi::fromBString (BString str, DigitalMode &v)`
- `BString BMeasureApi::toBStringJson (BString n, DigitalMode v)`
- `BString BMeasureApi::toBString (AwgMode v)`
- `BError BMeasureApi::fromBString (BString str, AwgMode &v)`
- `BString BMeasureApi::toBStringJson (BString n, AwgMode v)`
- `BString BMeasureApi::toBString (AwgOutput v)`
- `BError BMeasureApi::fromBString (BString str, AwgOutput &v)`
- `BString BMeasureApi::toBStringJson (BString n, AwgOutput v)`

- **BString** [BMeasureApi::toBString](#) (FileType v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, FileType &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, FileType v)
- **BString** [BMeasureApi::toBString](#) (FilesysDeleteType v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, FilesysDeleteType &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, FilesysDeleteType v)
- **BString** [BMeasureApi::toBString](#) (LogData v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, LogData &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, LogData v)
- **BString** [BMeasureApi::toBString](#) (LogDataMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, LogDataMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, LogDataMode v)
- **BString** [BMeasureApi::toBString](#) (DataType v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, DataType &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, DataType v)
- **BString** [BMeasureApi::toBString](#) (DataSend v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, DataSend &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, DataSend v)
- **BString** [BMeasureApi::toBString](#) (CalibrateStage v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, CalibrateStage &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, CalibrateStage v)
- **BString** [BMeasureApi::toBString](#) (MessageSource v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, MessageSource &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, MessageSource v)
- **BString** [BMeasureApi::toBString](#) (NetworkMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, NetworkMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, NetworkMode v)
- **BString** [BMeasureApi::toBString](#) (WifiMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, WifiMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, WifiMode v)
- **BString** [BMeasureApi::toBString](#) (AlarmMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, AlarmMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, AlarmMode v)
- **BString** [BMeasureApi::toBString](#) (AlarmOutput v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, AlarmOutput &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, AlarmOutput v)
- **BString** [BMeasureApi::toBString](#) (EventMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, EventMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, EventMode v)
- **BString** [BMeasureApi::toBString](#) (Rs485Mode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, Rs485Mode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, Rs485Mode v)
- **BString** [BMeasureApi::toBString](#) (BMeasFileType v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, BMeasFileType &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, BMeasFileType v)

8.6.1 Macro Definition Documentation

8.6.1.1 boffsetof

```
#define boffsetof(  
    T,  
    F ) (( BUInt ) ((char*)&((T*)0L)->F - (char*)0L))
```

8.7 BMeasureD.h File Reference

```
#include <BTypes.h>  
#include <BObj.h>  
#include <BTime.h>  
#include <BTimeUs.h>  
#include <BArray.h>  
#include <BComplex.h>  
#include <BoapMc.h>
```

Classes

- class [BMeasureApi::Version](#)
- class [BMeasureApi::NodeInfo](#)
- class [BMeasureApi::NodeStatus](#)
- class [BMeasureApi::BoardConfig](#)
- class [BMeasureApi::ChannelConfig](#)
- class [BMeasureApi::Information](#)
- class [BMeasureApi::AlarmConfig](#)
- class [BMeasureApi::Configuration](#)
- class [BMeasureApi::ConfigItem](#)
- class [BMeasureApi::MeasurementConfig](#)
- class [BMeasureApi::DataBlock](#)
- class [BMeasureApi::DataProc](#)
- class [BMeasureApi::DataBlockProc](#)
- class [BMeasureApi::InfoBlock](#)
- class [BMeasureApi::AwgConfig](#)
- class [BMeasureApi::FilesysInfo](#)
- class [BMeasureApi::FileInfo](#)
- class [BMeasureApi::FileData](#)
- class [BMeasureApi::CalibrateInfo](#)

Namespaces

- [BMeasureApi](#)

Enumerations

- enum `BMeasureApi::ErrorNum` { `BMeasureApi::ErrorNumSystem` = 64, `BMeasureApi::ErrorNumDataOvrrun` = 65, `BMeasureApi::ErrorNumToFast` = 66 }
- enum `BMeasureApi::NodeType` { `BMeasureApi::NodeTypeNone` = 0, `BMeasureApi::NodeTypeBMeasure1` = 1 }
- enum `BMeasureApi::SecurityMode` { `BMeasureApi::SecurityModeBasic`, `BMeasureApi::SecurityModeConfig`, `BMeasureApi::SecurityModeFull` }
- enum `BMeasureApi::Status` { `BMeasureApi::StatusNone` = 0x00, `BMeasureApi::StatusError` = 0x01, `BMeasureApi::StatusWarning` = 0x02, `BMeasureApi::StatusRun` = 0x04, `BMeasureApi::StatusTriggerWait` = 0x08, `BMeasureApi::StatusEnd0` = 0x10, `BMeasureApi::StatusEnd1` = 0x20, `BMeasureApi::StatusDataOvrrun` = 0x40, `BMeasureApi::StatusFpgaOvrrun` = 0x80, `BMeasureApi::StatusAlarm` = 0x0100 }
- enum `BMeasureApi::Mode` { `BMeasureApi::Modeldle` = 0, `BMeasureApi::ModeRun` = 1, `BMeasureApi::ModeRunProgram` = 2, `BMeasureApi::ModelInternal` = 3, `BMeasureApi::ModeSleep` = 4, `BMeasureApi::ModeDemo1` = 5 }
- enum `BMeasureApi::BlockTypes` { `BMeasureApi::BlockTypeInfo` = 0x424E4531, `BMeasureApi::BlockTypeData` = 0x424E4532 }
- enum `BMeasureApi::ChannelType` { `BMeasureApi::ChannelTypeNone` = 0, `BMeasureApi::ChannelTypeAnalogueIn` = 1, `BMeasureApi::ChannelTypeAnalogueOut` = 2, `BMeasureApi::ChannelTypeDigitalIn` = 3, `BMeasureApi::ChannelTypeDigitalOut` = 4 }
- enum `BMeasureApi::SampleType` { `BMeasureApi::SampleTypeNone` = 0, `BMeasureApi::SampleTypeBool` = 1, `BMeasureApi::SampleTypeInt8` = 2, `BMeasureApi::SampleTypeInt16` = 3, `BMeasureApi::SampleTypeInt32` = 4, `BMeasureApi::SampleTypeFloat32` = 5, `BMeasureApi::SampleTypeFloat64` = 6 }
- enum `BMeasureApi::SyncMode` { `BMeasureApi::SyncModeOff` = 0, `BMeasureApi::SyncModeMaster` = 1, `BMeasureApi::SyncModeSlave` = 2 }
- enum `BMeasureApi::MeasureMode` { `BMeasureApi::MeasureModeOff` = 0, `BMeasureApi::MeasureModeOneShot` = 1, `BMeasureApi::MeasureModeRepeat` = 2, `BMeasureApi::MeasureModeContinuous` = 3 }
- enum `BMeasureApi::MeasureOption` { `BMeasureApi::MeasureOptionNone` = 0, `BMeasureApi::MeasureOptionProcess` = 0x01 }
- enum `BMeasureApi::TriggerMode` { `BMeasureApi::TriggerModeOff` = 0, `BMeasureApi::TriggerModePositive` = 1, `BMeasureApi::TriggerModeNegative` = 2 }
- enum `BMeasureApi::TriggerConfig` { `BMeasureApi::TriggerConfigNone` = 0 }
- enum `BMeasureApi::DigitalMode` { `BMeasureApi::DigitalModeInput` = 0, `BMeasureApi::DigitalModeOutput` = 1, `BMeasureApi::DigitalModeInOut` = 2, `BMeasureApi::DigitalModeSyncMaster` = 3, `BMeasureApi::DigitalModeSyncSlave` = 4 }
- enum `BMeasureApi::AwgMode` { `BMeasureApi::AwgModeNone`, `BMeasureApi::AwgModeDc`, `BMeasureApi::AwgModeSine`, `BMeasureApi::AwgModeSquare`, `BMeasureApi::AwgModeTriangle`, `BMeasureApi::AwgModeNoise`, `BMeasureApi::AwgModeTrackRms`, `BMeasureApi::AwgModeTrackMean`, `BMeasureApi::AwgModeArbitrary` }
- enum `BMeasureApi::AwgOutput` { `BMeasureApi::AwgOutputNone`, `BMeasureApi::AwgOutputAO0`, `BMeasureApi::AwgOutputAO1`, `BMeasureApi::AwgOutputAO01` }
- enum `BMeasureApi::FileType` { `BMeasureApi::FileTypeNone`, `BMeasureApi::FileTypeFile`, `BMeasureApi::FileTypeDir` }
- enum `BMeasureApi::FilesysDeleteType` { `BMeasureApi::FilesysDeleteTypeNone`, `BMeasureApi::FilesysDeleteTypeData`, `BMeasureApi::FilesysDeleteTypeFormat` }
- enum `BMeasureApi::LogData` { `BMeasureApi::LogDataOff`, `BMeasureApi::LogDataRaw` = 0x01, `BMeasureApi::LogDataProcess` = 0x02 }
- enum `BMeasureApi::LogDataMode` { `BMeasureApi::LogDataModeNormal`, `BMeasureApi::LogDataModeDeleteOld` }

- enum `BMeasureApi::DataType` { `BMeasureApi::DataTypeFloat32`, `BMeasureApi::DataType125i`, `BMeasureApi::DataTypeProc` }
- enum `BMeasureApi::DataSend` { `BMeasureApi::DataSendOff`, `BMeasureApi::DataSendStatus` = 0x01, `BMeasureApi::DataSendRaw` = 0x02, `BMeasureApi::DataSendProcessed` = 0x04 }
- enum `BMeasureApi::CalibrateStage` { `BMeasureApi::CalibrateStageNone` = 0, `BMeasureApi::CalibrateStageClear` = 1, `BMeasureApi::CalibrateStageSettle` = 2, `BMeasureApi::CalibrateStageAdcOffsets` = 3, `BMeasureApi::CalibrateStageDacOffsets` = 4, `BMeasureApi::CalibrateStageDacScaling0` = 5, `BMeasureApi::CalibrateStageDacScaling1` = 6, `BMeasureApi::CalibrateStageAdcScaling` = 7, `BMeasureApi::CalibrateStageAttenScaling` = 8, `BMeasureApi::CalibrationStageFiveVolts` = 9 }
- enum `BMeasureApi::MessageSource` { `BMeasureApi::MessageSourceGeneral` = 0, `BMeasureApi::MessageSourceDebug` = 1, `BMeasureApi::MessageSourceTest` = 2, `BMeasureApi::MessageSourceWifi` = 3, `BMeasureApi::MessageSourceWifiTest` = 4 }
- enum `BMeasureApi::NetworkMode` { `BMeasureApi::NetworkModeOff` = 0, `BMeasureApi::NetworkModeDhcp` = 1, `BMeasureApi::NetworkModeManual` = 2 }
- enum `BMeasureApi::WifiMode` { `BMeasureApi::WifiModeOff`, `BMeasureApi::WifiModeClient`, `BMeasureApi::WifiModeAp` }
- enum `BMeasureApi::AlarmMode` { `BMeasureApi::AlarmModeOff`, `BMeasureApi::AlarmModeHigh`, `BMeasureApi::AlarmModeLow`, `BMeasureApi::AlarmModeRange` }
- enum `BMeasureApi::AlarmOutput` { `BMeasureApi::AlarmOutputOff`, `BMeasureApi::AlarmOutputDioHigh`, `BMeasureApi::AlarmOutputDioLow`, `BMeasureApi::AlarmOutputRelayOn`, `BMeasureApi::AlarmOutputRelayOff` }
- enum `BMeasureApi::EventMode` { `BMeasureApi::EventModeOff`, `BMeasureApi::EventModeAlarm`, `BMeasureApi::EventModeSecond` }
- enum `BMeasureApi::Rs485Mode` { `BMeasureApi::Rs485ModeOff`, `BMeasureApi::Rs485ModeBoap` }
- enum `BMeasureApi::BMeasFileType` { `BMeasureApi::BMeasFileTypeBlock512`, `BMeasureApi::BMeasFileTypeStream` }

Functions

- `BString` `BMeasureApi::toBString` (ErrorNum v)
- `BError` `BMeasureApi::fromBString` (`BString` str, ErrorNum &v)
- `BString` `BMeasureApi::toBStringJson` (`BString` n, ErrorNum v)
- `BString` `BMeasureApi::toBString` (NodeType v)
- `BError` `BMeasureApi::fromBString` (`BString` str, NodeType &v)
- `BString` `BMeasureApi::toBStringJson` (`BString` n, NodeType v)
- `BString` `BMeasureApi::toBString` (SecurityMode v)
- `BError` `BMeasureApi::fromBString` (`BString` str, SecurityMode &v)
- `BString` `BMeasureApi::toBStringJson` (`BString` n, SecurityMode v)
- `BString` `BMeasureApi::toBString` (Status v)
- `BError` `BMeasureApi::fromBString` (`BString` str, Status &v)
- `BString` `BMeasureApi::toBStringJson` (`BString` n, Status v)
- `BString` `BMeasureApi::toBString` (Mode v)
- `BError` `BMeasureApi::fromBString` (`BString` str, Mode &v)
- `BString` `BMeasureApi::toBStringJson` (`BString` n, Mode v)
- `BString` `BMeasureApi::toBString` (BlockTypes v)
- `BError` `BMeasureApi::fromBString` (`BString` str, BlockTypes &v)
- `BString` `BMeasureApi::toBStringJson` (`BString` n, BlockTypes v)
- `BString` `BMeasureApi::toBString` (ChannelType v)
- `BError` `BMeasureApi::fromBString` (`BString` str, ChannelType &v)
- `BString` `BMeasureApi::toBStringJson` (`BString` n, ChannelType v)
- `BString` `BMeasureApi::toBString` (SampleType v)
- `BError` `BMeasureApi::fromBString` (`BString` str, SampleType &v)

- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, SampleType v)
- **BString** [BMeasureApi::toBString](#) (SyncMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, SyncMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, SyncMode v)
- **BString** [BMeasureApi::toBString](#) (MeasureMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, MeasureMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, MeasureMode v)
- **BString** [BMeasureApi::toBString](#) (MeasureOption v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, MeasureOption &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, MeasureOption v)
- **BString** [BMeasureApi::toBString](#) (TriggerMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, TriggerMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, TriggerMode v)
- **BString** [BMeasureApi::toBString](#) (TriggerConfig v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, TriggerConfig &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, TriggerConfig v)
- **BString** [BMeasureApi::toBString](#) (DigitalMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, DigitalMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, DigitalMode v)
- **BString** [BMeasureApi::toBString](#) (AvgMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, AvgMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, AvgMode v)
- **BString** [BMeasureApi::toBString](#) (AvgOutput v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, AvgOutput &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, AvgOutput v)
- **BString** [BMeasureApi::toBString](#) (FileType v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, FileType &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, FileType v)
- **BString** [BMeasureApi::toBString](#) (FilesysDeleteType v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, FilesysDeleteType &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, FilesysDeleteType v)
- **BString** [BMeasureApi::toBString](#) (LogData v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, LogData &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, LogData v)
- **BString** [BMeasureApi::toBString](#) (LogDataMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, LogDataMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, LogDataMode v)
- **BString** [BMeasureApi::toBString](#) (DataType v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, DataType &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, DataType v)
- **BString** [BMeasureApi::toBString](#) (DataSend v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, DataSend &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, DataSend v)
- **BString** [BMeasureApi::toBString](#) (CalibrateStage v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, CalibrateStage &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, CalibrateStage v)
- **BString** [BMeasureApi::toBString](#) (MessageSource v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, MessageSource &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, MessageSource v)
- **BString** [BMeasureApi::toBString](#) (NetworkMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, NetworkMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, NetworkMode v)
- **BString** [BMeasureApi::toBString](#) (WifiMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, WifiMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, WifiMode v)

- **BString** [BMeasureApi::toBString](#) (AlarmMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, AlarmMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, AlarmMode v)
- **BString** [BMeasureApi::toBString](#) (AlarmOutput v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, AlarmOutput &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, AlarmOutput v)
- **BString** [BMeasureApi::toBString](#) (EventMode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, EventMode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, EventMode v)
- **BString** [BMeasureApi::toBString](#) (Rs485Mode v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, Rs485Mode &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, Rs485Mode v)
- **BString** [BMeasureApi::toBString](#) (BMeasFileType v)
- **BError** [BMeasureApi::fromBString](#) (**BString** str, BMeasFileType &v)
- **BString** [BMeasureApi::toBStringJson](#) (**BString** n, BMeasFileType v)

8.8 BMeasureLib.cpp File Reference

```
#include <BMeasureLib.h>
#include <BObjStringFormat.h>
#include <BDebug.h>
```

Namespaces

- [BMeasureApi](#)

Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0

Functions

- **BString** [toBStringJson](#) (**BString** n, [BMeasureApi::Version](#) v)
- **BString** [toBStringJson](#) (**BString** n, [BMeasureApi::AlarmConfig](#) v)
- **BString** [toBStringJson](#) (**BString** n, [BMeasureApi::NodeInfo](#) v)

8.8.1 Macro Definition Documentation

8.8.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.8.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.8.2 Function Documentation

8.8.2.1 toBStringJson() [1/3]

```
BString toBStringJson (  
    BString n,  
    BMeasureApi::Version v )
```

8.8.2.2 toBStringJson() [2/3]

```
BString toBStringJson (  
    BString n,  
    BMeasureApi::AlarmConfig v )
```

8.8.2.3 toBStringJson() [3/3]

```
BString toBStringJson (  
    BString n,  
    BMeasureApi::NodeInfo v )
```

8.9 BMeasureLib.h File Reference

```
#include <BMeasureD.h>
```

Namespaces

- [BMeasureApi](#)

Typedefs

- typedef **BArray**< ChannelConfig > [BMeasureApi::ChannelConfigs](#)

Functions

- [BString toBStringJson](#) ([BString](#) n, [BMeasureApi::Version](#) v)
- [BString toBStringJson](#) ([BString](#) n, [BMeasureApi::AlarmConfig](#) v)
- [BString toBStringJson](#) ([BString](#) n, [BMeasureApi::NodeInfo](#) v)

8.9.1 Function Documentation

8.9.1.1 toBStringJson() [1/3]

```
BString toBStringJson (  
    BString n,  
    BMeasureApi::Version v )
```

8.9.1.2 toBStringJson() [2/3]

```
BString toBStringJson (  
    BString n,  
    BMeasureApi::AlarmConfig v )
```

8.9.1.3 toBStringJson() [3/3]

```
BString toBStringJson (  
    BString n,  
    BMeasureApi::NodeInfo v )
```

8.10 BMeasureS.cpp File Reference

```
#include <BMeasureS.h>  
#include <string.h>
```

Namespaces

- [BMeasureApi](#)

8.11 BMeasureUnit.cpp File Reference

```
#include <BMeasureUnit.h>
#include <CommsSerial.h>
#include <CommsNet.h>
#include <CommsUsb.h>
#include <BDir.h>
#include <BSys.h>
#include <libusb-1.0/libusb.h>
#include <BMdns.h>
#include <BDebug.h>
#include <unistd.h>
```

Namespaces

- [BMeasureApi](#)

Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0
- #define [BDEBUGL3](#) 0
- #define [CONVERT_FLOAT](#) 0
Convert to floating point.

Functions

- const char * [BMeasureApi::channelTypeString](#) (ChannelType type)
- const char * [BMeasureApi::sampleTypeString](#) (SampleType type)
- **BFloat32** [BMeasureApi::toFloat](#) (**BUInt32** v)

8.11.1 Macro Definition Documentation

8.11.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.11.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.11.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

8.11.1.4 CONVERT_FLOAT

```
#define CONVERT_FLOAT 0
```

Convert to floating point.

8.12 BMeasureUnit.h File Reference

```
#include <BMeasureD.h>  
#include <BMeasureB.h>  
#include <BTask.h>
```

Classes

- class [BMeasureApi::BMeasureUnitDevice](#)
- class [BMeasureApi::BMeasureUnit](#)

Namespaces

- [BMeasureApi](#)

Functions

- const char * [BMeasureApi::channelTypeString](#) (ChannelType type)
- const char * [BMeasureApi::sampleTypeString](#) (SampleType type)

8.13 BMeasureUnits.cpp File Reference

```
#include <BMeasureUnits.h>  
#include <BDebug.h>  
#include <unistd.h>
```

Namespaces

- [BMeasureApi](#)

Macros

- `#define BDEBUGL1 0`
- `#define BDEBUGL2 0`
- `#define BDEBUGL3 0`

Functions

- `static int BMeasureApi::unitSort (BMeasureUnit1 *&u1, BMeasureUnit1 *&u2)`

8.13.1 Macro Definition Documentation

8.13.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.13.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.13.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

8.14 BMeasureUnits.h File Reference

```
#include <BMeasureUnit.h>  
#include <BMutex.h>  
#include <BSemaphore.h>
```

Classes

- class [BMeasureApi::BMeasureUnit1](#)
- class [BMeasureApi::BMeasureUnitsDataBlock](#)
- class [BMeasureApi::BMeasureUnits](#)

Namespaces

- [BMeasureApi](#)

8.15 CommsNet.cpp File Reference

```
#include <CommsNet.h>
#include <BPoll.h>
#include <BSys.h>
#include <BDebug.h>
#include <string.h>
```

Namespaces

- [BMeasureApi](#)

Macros

- `#define BDEBUGL1 0`
- `#define BDEBUGL2 0`
- `#define BDEBUGL3 0`

8.15.1 Macro Definition Documentation

8.15.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.15.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.15.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

8.16 CommsNet.h File Reference

```
#include <BComms.h>
#include <BSocket.h>
```

Classes

- class [BMeasureApi::CommsNet](#)

Namespaces

- [BMeasureApi](#)

8.17 CommsSerial.cpp File Reference

8.18 CommsSerial.h File Reference

```
#include <BComms.h>
```

Classes

- class [BMeasureApi::CommsSerial](#)

Namespaces

- [BMeasureApi](#)

8.19 CommsUsb.cpp File Reference

```
#include <CommsUsb.h>
#include <BSys.h>
#include <libusb-1.0/libusb.h>
#include <stdio.h>
#include <stdlib.h>
#include <BDebug.h>
```

Namespaces

- [BMeasureApi](#)

Macros

- `#define BDEBUGL1 0`
- `#define BDEBUGL2 0`

Functions

- static `BUInt32 BMeasureApi::roundDown512 (BUInt32 size)`

8.19.1 Macro Definition Documentation

8.19.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.19.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.20 CommsUsb.h File Reference

```
#include <BComms.h>  
#include <BMutex.h>  
#include <libusb-1.0/libusb.h>
```

Classes

- class `BMeasureApi::CommsUsb`

Namespaces

- `BMeasureApi`

8.21 DataFile.cpp File Reference

```
#include <DataFile.h>  
#include <BoapMcl.h>  
#include <BBuffer.h>  
#include <BDebug.h>  
#include <errno.h>
```

Namespaces

- [BMeasureApi](#)

Macros

- `#define BDEBUGL1 0`
- `#define BDEBUGL2 0`

Enumerations

- enum `BMeasureApi::TdsDataType` {
[BMeasureApi::TdsTypeVoid](#), [BMeasureApi::TdsTypeI8](#), [BMeasureApi::TdsTypeI16](#), [BMeasureApi::TdsTypeI32](#),
[BMeasureApi::TdsTypeI64](#), [BMeasureApi::TdsTypeU8](#), [BMeasureApi::TdsTypeU16](#), [BMeasureApi::TdsTypeU32](#),
[BMeasureApi::TdsTypeU64](#), [BMeasureApi::TdsTypeSingleFloat](#), [BMeasureApi::TdsTypeDoubleFloat](#),
[BMeasureApi::TdsTypeExtendedFloat](#),
[BMeasureApi::TdsTypeSingleFloatWithUnit](#) =0x19, [BMeasureApi::TdsTypeDoubleFloatWithUnit](#), [BMeasureApi::TdsTypeExtendedFloatWithUnit](#) =0x20,
[BMeasureApi::TdsTypeBoolean](#) =0x21, [BMeasureApi::TdsTypeTimeStamp](#) =0x44, [BMeasureApi::TdsTypeFixedPoint](#)
=0x4F, [BMeasureApi::TdsTypeComplexSingleFloat](#) =0x08000c,
[BMeasureApi::TdsTypeComplexDoubleFloat](#) =0x10000d, [BMeasureApi::TdsTypeDAQmxRawData](#) =0xFF↔
FFFFFF }

Functions

- const `BUInt32 BMeasureApi::TocMetaData` (1<< 1)
- const `BUInt32 BMeasureApi::TocNewObjList` (1<< 2)
- const `BUInt32 BMeasureApi::TocRawData` (1<< 3)
- const `BUInt32 BMeasureApi::TocInterleavedData` (1<< 5)
- const `BUInt32 BMeasureApi::TocBigEndian` (1<< 6)
- const `BUInt32 BMeasureApi::TocDaqRawData` (1<< 7)
- `BUInt32 BMeasureApi::round512` (`BUInt32 s`)

8.21.1 Macro Definition Documentation

8.21.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.21.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.22 DataFile.h File Reference

```
#include <BString.h>
#include <BFile.h>
#include <BMeasureLib.h>
#include <BoapMcl.h>
```

Classes

- class [BMeasureApi::DataFile](#)

Namespaces

- [BMeasureApi](#)

8.23 Dfu.cpp File Reference

```
#include <Dfu.h>
#include <BFile.h>
#include <BDebug.h>
#include <unistd.h>
```

Classes

- struct [BFirmwareInfo](#)

Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0
- #define [STATE_APP_IDLE](#) 0x00
- #define [STATE_APP_DETACH](#) 0x01
- #define [STATE_DFU_IDLE](#) 0x02
- #define [STATE_DFU_DOWNLOAD_SYNC](#) 0x03
- #define [STATE_DFU_DOWNLOAD_BUSY](#) 0x04
- #define [STATE_DFU_DOWNLOAD_IDLE](#) 0x05
- #define [STATE_DFU_MANIFEST_SYNC](#) 0x06
- #define [STATE_DFU_MANIFEST](#) 0x07
- #define [STATE_DFU_MANIFEST_WAIT_RESET](#) 0x08
- #define [STATE_DFU_UPLOAD_IDLE](#) 0x09
- #define [STATE_DFU_ERROR](#) 0x0a
- #define [DFU_STATUS_OK](#) 0x00
- #define [DFU_STATUS_ERROR_TARGET](#) 0x01
- #define [DFU_STATUS_ERROR_FILE](#) 0x02
- #define [DFU_STATUS_ERROR_WRITE](#) 0x03
- #define [DFU_STATUS_ERROR_ERASE](#) 0x04

- #define `DFU_STATUS_ERROR_CHECK_ERASED` 0x05
- #define `DFU_STATUS_ERROR_PROG` 0x06
- #define `DFU_STATUS_ERROR_VERIFY` 0x07
- #define `DFU_STATUS_ERROR_ADDRESS` 0x08
- #define `DFU_STATUS_ERROR_NOTDONE` 0x09
- #define `DFU_STATUS_ERROR_FIRMWARE` 0x0a
- #define `DFU_STATUS_ERROR_VENDOR` 0x0b
- #define `DFU_STATUS_ERROR_USBR` 0x0c
- #define `DFU_STATUS_ERROR_POR` 0x0d
- #define `DFU_STATUS_ERROR_UNKNOWN` 0x0e
- #define `DFU_STATUS_ERROR_STALLEDPKT` 0x0f
- #define `DFU_DETACH` 0
- #define `DFU_DNLOAD` 1
- #define `DFU_UPLOAD` 2
- #define `DFU_GETSTATUS` 3
- #define `DFU_CLRSTATUS` 4
- #define `DFU_GETSTATE` 5
- #define `DFU_ABORT` 6
- #define `DFU_IFF_DFU` 0x0001 /* DFU Mode, (not Runtime) */
- #define `DFU_IFF_VENDOR` 0x0100
- #define `DFU_IFF_PRODUCT` 0x0200
- #define `DFU_IFF_CONFIG` 0x0400
- #define `DFU_IFF_IFACE` 0x0800
- #define `DFU_IFF_ALT` 0x1000
- #define `DFU_IFF_DEVNUM` 0x2000
- #define `DFU_IFF_PATH` 0x4000

Enumerations

- enum `dfuse_command` { `SET_ADDRESS`, `ERASE_PAGE`, `MASS_ERASE`, `READ_UNPROTECT` }

Functions

- static `BInt32` `pageNumber` (`BUInt32` address)
- static `BUInt32` `pageAddress` (`BUInt32` page)

Variables

- const `BUInt32` `BFirmwareInfoMagic` = 0xBBEEAA00
- const `BUInt8` `BFirmwareInfoEncrypt1` = 0x40

8.23.1 Macro Definition Documentation

8.23.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.23.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.23.1.3 DFU_ABORT

```
#define DFU_ABORT 6
```

8.23.1.4 DFU_CLRSTATUS

```
#define DFU_CLRSTATUS 4
```

8.23.1.5 DFU_DETACH

```
#define DFU_DETACH 0
```

8.23.1.6 DFU_DNLOAD

```
#define DFU_DNLOAD 1
```

8.23.1.7 DFU_GETSTATE

```
#define DFU_GETSTATE 5
```

8.23.1.8 DFU_GETSTATUS

```
#define DFU_GETSTATUS 3
```

8.23.1.9 DFU_IFF_ALT

```
#define DFU_IFF_ALT 0x1000
```

8.23.1.10 DFU_IFF_CONFIG

```
#define DFU_IFF_CONFIG 0x0400
```

8.23.1.11 DFU_IFF_DEVNUM

```
#define DFU_IFF_DEVNUM 0x2000
```

8.23.1.12 DFU_IFF_DFU

```
#define DFU_IFF_DFU 0x0001 /* DFU Mode, (not Runtime) */
```

8.23.1.13 DFU_IFF_IFACE

```
#define DFU_IFF_IFACE 0x0800
```

8.23.1.14 DFU_IFF_PATH

```
#define DFU_IFF_PATH 0x4000
```

8.23.1.15 DFU_IFF_PRODUCT

```
#define DFU_IFF_PRODUCT 0x0200
```

8.23.1.16 DFU_IFF_VENDOR

```
#define DFU_IFF_VENDOR 0x0100
```

8.23.1.17 DFU_STATUS_ERROR_ADDRESS

```
#define DFU_STATUS_ERROR_ADDRESS 0x08
```

8.23.1.18 DFU_STATUS_ERROR_CHECK_ERASED

```
#define DFU_STATUS_ERROR_CHECK_ERASED 0x05
```

8.23.1.19 DFU_STATUS_ERROR_ERASE

```
#define DFU_STATUS_ERROR_ERASE 0x04
```

8.23.1.20 DFU_STATUS_ERROR_FILE

```
#define DFU_STATUS_ERROR_FILE 0x02
```

8.23.1.21 DFU_STATUS_ERROR_FIRMWARE

```
#define DFU_STATUS_ERROR_FIRMWARE 0x0a
```

8.23.1.22 DFU_STATUS_ERROR_NOTDONE

```
#define DFU_STATUS_ERROR_NOTDONE 0x09
```

8.23.1.23 DFU_STATUS_ERROR_POR

```
#define DFU_STATUS_ERROR_POR 0x0d
```

8.23.1.24 DFU_STATUS_ERROR_PROG

```
#define DFU_STATUS_ERROR_PROG 0x06
```

8.23.1.25 DFU_STATUS_ERROR_STALLEDPKT

```
#define DFU_STATUS_ERROR_STALLEDPKT 0x0f
```

8.23.1.26 DFU_STATUS_ERROR_TARGET

```
#define DFU_STATUS_ERROR_TARGET 0x01
```

8.23.1.27 DFU_STATUS_ERROR_UNKNOWN

```
#define DFU_STATUS_ERROR_UNKNOWN 0x0e
```

8.23.1.28 DFU_STATUS_ERROR_USBR

```
#define DFU_STATUS_ERROR_USBR 0x0c
```

8.23.1.29 DFU_STATUS_ERROR_VENDOR

```
#define DFU_STATUS_ERROR_VENDOR 0x0b
```

8.23.1.30 DFU_STATUS_ERROR_VERIFY

```
#define DFU_STATUS_ERROR_VERIFY 0x07
```

8.23.1.31 DFU_STATUS_ERROR_WRITE

```
#define DFU_STATUS_ERROR_WRITE 0x03
```

8.23.1.32 DFU_STATUS_OK

```
#define DFU_STATUS_OK 0x00
```

8.23.1.33 DFU_UPLOAD

```
#define DFU_UPLOAD 2
```


8.23.1.34 STATE_APP_DETACH

```
#define STATE_APP_DETACH 0x01
```

8.23.1.35 STATE_APP_IDLE

```
#define STATE_APP_IDLE 0x00
```

8.23.1.36 STATE_DFU_DOWNLOAD_BUSY

```
#define STATE_DFU_DOWNLOAD_BUSY 0x04
```

8.23.1.37 STATE_DFU_DOWNLOAD_IDLE

```
#define STATE_DFU_DOWNLOAD_IDLE 0x05
```

8.23.1.38 STATE_DFU_DOWNLOAD_SYNC

```
#define STATE_DFU_DOWNLOAD_SYNC 0x03
```

8.23.1.39 STATE_DFU_ERROR

```
#define STATE_DFU_ERROR 0x0a
```

8.23.1.40 STATE_DFU_IDLE

```
#define STATE_DFU_IDLE 0x02
```

8.23.1.41 STATE_DFU_MANIFEST

```
#define STATE_DFU_MANIFEST 0x07
```

8.23.1.42 STATE_DFU_MANIFEST_SYNC

```
#define STATE_DFU_MANIFEST_SYNC 0x06
```

8.23.1.43 STATE_DFU_MANIFEST_WAIT_RESET

```
#define STATE_DFU_MANIFEST_WAIT_RESET 0x08
```

8.23.1.44 STATE_DFU_UPLOAD_IDLE

```
#define STATE_DFU_UPLOAD_IDLE 0x09
```

8.23.2 Enumeration Type Documentation

8.23.2.1 dfuse_command

```
enum dfuse_command
```

Enumerator

SET_ADDRESS	
ERASE_PAGE	
MASS_ERASE	
READ_UNPROTECT	

8.23.3 Function Documentation

8.23.3.1 pageAddress()

```
static BUInt32 pageAddress (  
    BUInt32 page ) [static]
```

8.23.3.2 pageNumber()

```
static BInt32 pageNumber (
    BUInt32 address ) [static]
```

8.23.4 Variable Documentation

8.23.4.1 BFirmwareInfoEncrypt1

```
const BUInt8 BFirmwareInfoEncrypt1 = 0x40
```

8.23.4.2 BFirmwareInfoMagic

```
const BUInt32 BFirmwareInfoMagic = 0xBBEEAA00
```

8.24 Dfu.h File Reference

```
#include <BError.h>
#include <libusb-1.0/libusb.h>
```

Classes

- struct [DfuStatus](#)
- class [Dfu](#)

The [Dfu](#) access class.

8.25 overview.dox File Reference

Index

- ~BMdns
 - BMdns, [53](#)
- ~BMeasureUnit
 - BMeasureApi::BMeasureUnit, [76](#)
- ~BMeasureUnits
 - BMeasureApi::BMeasureUnits, [87](#)
- ~BMeasureUnitsDataBlock
 - BMeasureApi::BMeasureUnitsDataBlock, [100](#)
- ~CommsNet
 - BMeasureApi::CommsNet, [111](#)
- ~CommsSerial
 - BMeasureApi::CommsSerial, [114](#)
- ~CommsUsb
 - BMeasureApi::CommsUsb, [116](#)
- ~DataFile
 - BMeasureApi::DataFile, [134](#)
- ~Dfu
 - Dfu, [140](#)
- address
 - BMdnsService, [54](#)
- alarm
 - BMeasureApi::DataProc, [138](#)
- AlarmMode
 - BMeasureApi, [19](#)
- AlarmModeHigh
 - BMeasureApi, [19](#)
- AlarmModeLow
 - BMeasureApi, [19](#)
- AlarmModeOff
 - BMeasureApi, [19](#)
- AlarmModeRange
 - BMeasureApi, [19](#)
- AlarmOutput
 - BMeasureApi, [19](#)
- AlarmOutputDioHigh
 - BMeasureApi, [20](#)
- AlarmOutputDioLow
 - BMeasureApi, [20](#)
- AlarmOutputOff
 - BMeasureApi, [20](#)
- AlarmOutputRelayOff
 - BMeasureApi, [20](#)
- AlarmOutputRelayOn
 - BMeasureApi, [20](#)
- alarms
 - BMeasureApi::Configuration, [122](#)
- alarmsClear
 - BMeasureApi::BMeasure, [58](#)
 - BMeasureApi::BMeasureUnits, [87](#)
- alarmsClearServe
 - BMeasureApi::BMeasure, [59](#)
- amplitude
 - BMeasureApi::AwgConfig, [50](#)
- analogueData
 - BMeasureApi::DataBlockProc, [131](#)
- apiVersion
 - BMeasureApi, [45](#)
 - BMeasureApi::NodeInfo, [159](#)
- attenuator
 - BMeasureApi::ChannelConfig, [107](#)
- AwgMode
 - BMeasureApi, [20](#)
- AwgModeArbitrary
 - BMeasureApi, [20](#)
- AwgModeDc
 - BMeasureApi, [20](#)
- AwgModeNoise
 - BMeasureApi, [20](#)
- AwgModeNone
 - BMeasureApi, [20](#)
- AwgModeSine
 - BMeasureApi, [20](#)
- AwgModeSquare
 - BMeasureApi, [20](#)
- AwgModeTrackMean
 - BMeasureApi, [20](#)
- AwgModeTrackRms
 - BMeasureApi, [20](#)
- AwgModeTriangle
 - BMeasureApi, [20](#)
- AwgOutput
 - BMeasureApi, [20](#)
- AwgOutputAO0
 - BMeasureApi, [20](#)
- AwgOutputAO01
 - BMeasureApi, [20](#)
- AwgOutputAO1
 - BMeasureApi, [20](#)
- AwgOutputNone
 - BMeasureApi, [20](#)
- BDEBUGL1
 - BMdns.cpp, [166](#)
 - BMeasureLib.cpp, [175](#)
 - BMeasureUnit.cpp, [178](#)
 - BMeasureUnits.cpp, [180](#)
 - CommsNet.cpp, [181](#)
 - CommsUsb.cpp, [183](#)
 - DataFile.cpp, [184](#)

- Dfu.cpp, 186
- BDEBUGL2
 - BMeasureLib.cpp, 175
 - BMeasureUnit.cpp, 178
 - BMeasureUnits.cpp, 180
 - CommsNet.cpp, 181
 - CommsUsb.cpp, 183
 - DataFile.cpp, 184
 - Dfu.cpp, 186
- BDEBUGL3
 - BMeasureUnit.cpp, 178
 - BMeasureUnits.cpp, 180
 - CommsNet.cpp, 181
- BFirmwareInfo, 51
 - checksum, 51
 - length, 52
 - magic, 52
 - type, 52
 - ver0, 52
 - ver1, 52
 - ver2, 52
- BFirmwareInfoEncrypt1
 - Dfu.cpp, 193
- BFirmwareInfoMagic
 - Dfu.cpp, 193
- blockNumChannels
 - BMeasureApi::BMeasureUnit, 79
- blockNumSamples
 - BMeasureApi::BMeasureUnit, 79
- BlockTypeData
 - BMeasureApi, 21
- BlockTypeInfo
 - BMeasureApi, 21
- BlockTypes
 - BMeasureApi, 20
- BMdns, 53
 - ~BMdns, 53
 - BMdns, 53
 - findServices, 53
 - init, 53
 - osocket, 54
 - otransactionId, 54
- BMdns.cpp, 165
 - BDEBUGL1, 166
 - MDNS_CLASS_IN, 166
 - MDNS_ENTRYTYPE_ADDITIONAL, 166
 - MDNS_ENTRYTYPE_ANSWER, 166
 - MDNS_ENTRYTYPE_AUTHORITY, 166
 - mdns_read_string, 167
 - mdns_read_strings, 167
 - MDNS_RECORDTYPE_A, 166
 - MDNS_RECORDTYPE_AAAA, 166
 - MDNS_RECORDTYPE_IGNORE, 166
 - MDNS_RECORDTYPE_PTR, 166
 - MDNS_RECORDTYPE_SRV, 166
 - MDNS_RECORDTYPE_TXT, 166
 - mdns_write_string, 167
 - MdnsClass, 166
 - MdnsEntryType, 166
 - MdnsRecordType, 166
- BMdns.h, 167
- BMdnsService, 54
 - address, 54
 - extra, 54
 - hostname, 55
 - name, 55
- BMeasFileType
 - BMeasureApi, 21
- BMeasFileTypeBlock512
 - BMeasureApi, 21
- BMeasFileTypeStream
 - BMeasureApi, 21
- BMeasure
 - BMeasureApi::BMeasure, 58
- BMeasureApi, 15
 - AlarmMode, 19
 - AlarmModeHigh, 19
 - AlarmModeLow, 19
 - AlarmModeOff, 19
 - AlarmModeRange, 19
 - AlarmOutput, 19
 - AlarmOutputDioHigh, 20
 - AlarmOutputDioLow, 20
 - AlarmOutputOff, 20
 - AlarmOutputRelayOff, 20
 - AlarmOutputRelayOn, 20
 - apiVersion, 45
 - AwgMode, 20
 - AwgModeArbitrary, 20
 - AwgModeDc, 20
 - AwgModeNoise, 20
 - AwgModeNone, 20
 - AwgModeSine, 20
 - AwgModeSquare, 20
 - AwgModeTrackMean, 20
 - AwgModeTrackRms, 20
 - AwgModeTriangle, 20
 - AwgOutput, 20
 - AwgOutputAO0, 20
 - AwgOutputAO01, 20
 - AwgOutputAO1, 20
 - AwgOutputNone, 20
 - BlockTypeData, 21
 - BlockTypeInfo, 21
 - BlockTypes, 20
 - BMeasFileType, 21
 - BMeasFileTypeBlock512, 21
 - BMeasFileTypeStream, 21
 - CalibrateStage, 21
 - CalibrateStageAdcOffsets, 21
 - CalibrateStageAdcScaling, 21
 - CalibrateStageAttenScaling, 21
 - CalibrateStageClear, 21
 - CalibrateStageDacOffsets, 21
 - CalibrateStageDacScaling0, 21
 - CalibrateStageDacScaling1, 21

CalibrateStageNone, 21
CalibrateStageSettle, 21
CalibrationStageFiveVolts, 21
ChannelConfigs, 19
ChannelType, 21
ChannelTypeAnalogueIn, 21
ChannelTypeAnalogueOut, 21
ChannelTypeDigitalIn, 21
ChannelTypeDigitalOut, 21
ChannelTypeNone, 21
channelTypeString, 29
DataSend, 22
DataSendOff, 22
DataSendProcessed, 22
DataSendRaw, 22
DataSendStatus, 22
DataType, 22
DataType125i, 22
DataTypeFloat32, 22
DataTypeProc, 22
DigitalMode, 22
DigitalModeInOut, 22
DigitalModeInput, 22
DigitalModeOutput, 22
DigitalModeSyncMaster, 22
DigitalModeSyncSlave, 22
ErrorNum, 22
ErrorNumDataOverrun, 23
ErrorNumSystem, 23
ErrorNumToFast, 23
EventMode, 23
EventModeAlarm, 23
EventModeOff, 23
EventModeSecond, 23
FilesysDeleteType, 23
FilesysDeleteTypeData, 23
FilesysDeleteTypeFormat, 23
FilesysDeleteTypeNone, 23
FileType, 23
FileTypeDir, 23
FileTypeFile, 23
FileTypeNone, 23
fromBString, 29–34
LogData, 23
LogDataMode, 24
LogDataModeDeleteOld, 24
LogDataModeNormal, 24
LogDataOff, 24
LogDataProcessed, 24
LogDataRaw, 24
MeasureMode, 24
MeasureModeContinuous, 24
MeasureModeOff, 24
MeasureModeOneShot, 24
MeasureModeRepeat, 24
MeasureOption, 24
MeasureOptionNone, 24
MeasureOptionProcess, 24
MessageSource, 24
MessageSourceDebug, 25
MessageSourceGeneral, 25
MessageSourceTest, 25
MessageSourceWifi, 25
MessageSourceWifiTest, 25
Mode, 25
ModeDemo1, 25
ModeIdle, 25
ModeInternal, 25
ModeRun, 25
ModeRunProgram, 25
ModeSleep, 25
NetworkMode, 25
NetworkModeDhcp, 25
NetworkModeManual, 25
NetworkModeOff, 25
NodeType, 25
NodeTypeBMeasure1, 26
NodeTypeNone, 26
round512, 34
roundDown512, 34
Rs485Mode, 26
Rs485ModeBoap, 26
Rs485ModeOff, 26
SampleType, 26
SampleTypeBool, 26
SampleTypeFloat32, 26
SampleTypeFloat64, 26
SampleTypeInt16, 26
SampleTypeInt32, 26
SampleTypeInt8, 26
SampleTypeNone, 26
sampleTypeString, 34
SecurityMode, 26
SecurityModeBasic, 26
SecurityModeConfig, 26
SecurityModeFull, 26
Status, 26
StatusAlarm, 27
StatusDataOverrun, 27
StatusEnd0, 27
StatusEnd1, 27
StatusError, 27
StatusFpgaOverrun, 27
StatusNone, 27
StatusRun, 27
StatusTriggerWait, 27
StatusWarning, 27
SyncMode, 27
SyncModeMaster, 27
SyncModeOff, 27
SyncModeSlave, 27
TdsDataType, 27
TdsTypeBoolean, 28
TdsTypeComplexDoubleFloat, 28
TdsTypeComplexSingleFloat, 28
TdsTypeDAQmxRawData, 28

- TdsTypeDoubleFloat, 27
- TdsTypeDoubleFloatWithUnit, 28
- TdsTypeExtendedFloat, 28
- TdsTypeExtendedFloatWithUnit, 28
- TdsTypeFixedPoint, 28
- TdsTypeInt16, 27
- TdsTypeInt32, 27
- TdsTypeInt64, 27
- TdsTypeInt8, 27
- TdsTypeSingleFloat, 27
- TdsTypeSingleFloatWithUnit, 28
- TdsTypeString, 28
- TdsTypeTimeStamp, 28
- TdsTypeUInt16, 27
- TdsTypeUInt32, 27
- TdsTypeUInt64, 27
- TdsTypeUInt8, 27
- TdsTypeVoid, 27
- toBString, 34–38
- toBStringJson, 39–44
- TocBigEndian, 44
- TocDaqRawData, 44
- TocInterleavedData, 44
- TocMetaData, 44
- TocNewObjList, 44
- TocRawData, 44
- toFloat, 45
- TriggerConfig, 28
- TriggerConfigNone, 28
- TriggerMode, 28
- TriggerModeNegative, 28
- TriggerModeOff, 28
- TriggerModePositive, 28
- unitSort, 45
- WifiMode, 28
- WifiModeAp, 28
- WifiModeClient, 28
- WifiModeOff, 28
- BMeasureApi::AlarmConfig, 47
 - getMembers, 47
 - levelHigh, 48
 - levelLow, 48
 - mode, 48
 - output, 48
 - outputChannel, 48
 - spare, 48
- BMeasureApi::AwgConfig, 49
 - amplitude, 50
 - duty, 50
 - frequency, 50
 - getMembers, 49
 - mode, 50
 - numSamples, 50
 - offset, 50
 - output, 51
 - spare, 51
 - trackChannel, 51
- BMeasureApi::BMeasure, 55
 - alarmsClear, 58
 - alarmsClearServe, 59
 - BMeasure, 58
 - calibrate, 59
 - calibrateServe, 59
 - changePassword, 59
 - changePasswordServe, 59
 - factoryReset, 59
 - factoryResetServe, 60
 - fileClose, 60
 - fileCloseServe, 60
 - fileDelete, 60
 - fileDeleteServe, 60
 - fileList, 60
 - fileListServe, 61
 - fileOpen, 61
 - fileOpenServe, 61
 - fileRead, 61
 - fileReadServe, 61
 - fileSysDelete, 62
 - fileSysDeleteServe, 62
 - fileSysInfo, 62
 - fileSysInfoServe, 62
 - fileWrite, 62
 - fileWriteServe, 62
 - functionUnLock, 63
 - functionUnLockServe, 63
 - getAwgConfig, 63
 - getAwgConfigServe, 63
 - getBoardConfig, 63
 - getBoardConfigServe, 63
 - getChannelConfig, 64
 - getChannelConfigServe, 64
 - getConfig, 64
 - getConfigServe, 64
 - getDigital, 64
 - getDigitalServe, 64
 - getInfoBlock, 65
 - getInfoBlockServe, 65
 - getInformation, 65
 - getInformationServe, 65
 - getMeasurementConfig, 65
 - getMeasurementConfigServe, 65
 - getNodeInfo, 66
 - getNodeInfoServe, 66
 - getStatus, 66
 - getStatusServe, 66
 - getSwitch, 66
 - getSwitchServe, 66
 - login, 67
 - loginServe, 67
 - logout, 67
 - logoutServe, 67
 - measure, 67
 - measureServe, 67
 - processRequest, 68
 - runBoardTest, 68
 - runBoardTestServe, 68

- sendChannelConfig, 68
- sendChannelConfigServe, 68
- sendData, 68
- sendDataEnable, 69
- sendDataEnableServe, 69
- sendDataServe, 69
- sendInfo, 69
- sendInfoServe, 69
- sendMessage, 69
- sendMessageServe, 70
- sendStatus, 70
- sendStatusServe, 70
- sendTime, 70
- sendTimeServe, 70
- setAnalogueOut, 70
- setAnalogueOutServe, 71
- setAwgConfig, 71
- setAwgConfigServe, 71
- setAwgWaveform, 71
- setAwgWaveformServe, 71
- setBoardConfig, 72
- setBoardConfigServe, 72
- setChannelConfig, 72
- setChannelConfigFull, 72
- setChannelConfigFullServe, 72
- setChannelConfigServe, 72
- setConfig, 73
- setConfigServe, 73
- setDigital, 73
- setDigitalServe, 73
- setMeasurementConfig, 73
- setMeasurementConfigServe, 73
- setMode, 74
- setModeServe, 74
- setRelay, 74
- setRelayServe, 74
- BMeasureApi::BMeasureUnit, 75
 - ~BMeasureUnit, 76
 - blockNumChannels, 79
 - blockNumSamples, 79
 - BMeasureUnit, 76
 - connect, 76
 - device, 76
 - disconnect, 77
 - disconnected, 77
 - findDevices, 77
 - findDevicesNetwork, 77
 - findDevicesUsb, 77
 - info, 77
 - numChannels, 78
 - oblockCount, 79
 - ochannels, 79
 - oconfigMeasurement, 79
 - odataBlock, 80
 - odevice, 80
 - odisconnecting, 80
 - oinfo, 80
 - onodeInfo, 80
 - osampleCount, 80
 - osequenceNext, 80
 - processdataBlock, 78
 - run, 78
 - sendDataServe, 78
 - sendDataServe1, 78
 - serialNumber, 78
 - setChannelConfig, 79
 - setMeasurementConfig, 79
- BMeasureApi::BMeasureUnit1, 81
 - BMeasureUnit1, 81
 - disconnected, 82
 - oconnected, 83
 - oenabled, 83
 - omeasureUnits, 83
 - oorder, 83
 - oserialNumber, 83
 - osource, 83
 - sendDataServe1, 82
 - sendMessageServe, 82
 - serialNumber, 82
 - setSerialNumber, 82
- BMeasureApi::BMeasureUnitDevice, 84
 - BMeasureUnitDevice, 84
 - device, 84
 - serialNumber, 84
- BMeasureApi::BMeasureUnits, 85
 - ~BMeasureUnits, 87
 - alarmsClear, 87
 - BMeasureUnits, 87
 - changePassword, 87
 - clear, 88
 - dataAvailable, 88
 - dataClear, 88
 - dataDone, 88
 - dataEvent, 88
 - dataProcDone, 88
 - dataProcEvent, 88
 - dataProcRead, 89
 - dataRead, 89
 - dataSetNumStreams, 89
 - dataStreamEnable, 89
 - dataWait, 89
 - debugPrint, 89
 - disconnected, 90
 - getAwgConfig, 90
 - getChannelConfig, 90
 - getConfig, 90
 - getFreeBlock, 90
 - getInfoBlock, 90
 - getInformation, 91
 - getMeasurementConfig, 91
 - getNodeInfo, 91
 - getStatus, 91
 - login, 91
 - logout, 91
 - numChannels, 92
 - odataBlocksFree, 96

- odataBlocksIn, 96
- odataBlocksOut, 97
- odataBlocksOutCount, 97
- odataBlocksProcess, 97
- odataBlocksProcessNum, 97
- odataProcBlocks, 97
- odataStreamNum, 97
- ofill, 97
- olocalTrigger, 97
- olockInput, 98
- olockOutput, 98
- olockProInput, 98
- olockUnits, 98
- onumBlocks, 98
- onumChannels, 98
- onumConnected, 98
- oprocEnable, 98
- oprocRunning, 99
- ostartSample, 99
- otriggered, 99
- ounitMaster, 99
- ounits, 99
- outputBlock, 92
- run, 92
- sendDataEnable, 92
- sendDataProcess, 92
- sendDataProcessTrigger, 92
- sendDataProcQueue, 93
- sendDataQueue, 93
- sendDataServe1, 93
- sendMessage, 93
- sendMessageServe, 93
- sendTime, 93
- setAwgConfig, 93
- setChannelConfig, 94
- setConfig, 94
- setMeasurementConfig, 94
- setMode, 94
- unit, 94
- unitAdd, 95
- unitDelete, 95
- unitMaster, 95
- unitsConnect, 95
- unitsConnected, 95
- unitsConnectedNum, 95
- unitsDisconnect, 95
- unitSetEnabled, 96
- unitSetOrder, 96
- unitsFind, 96
- unitsNum, 96
- BMeasureApi::BMeasureUnitsDataBlock, 99
 - ~BMeasureUnitsDataBlock, 100
 - BMeasureUnitsDataBlock, 100
 - init, 100
 - odataBlock, 100
 - ofill, 101
 - oinUse, 101
- BMeasureApi::BoardConfig, 101
 - buildTime, 102
 - calibAdcOffsets, 102
 - calibAdcScales, 102
 - calibAttenScales, 102
 - calibDacOffsets, 102
 - calibDacScales, 102
 - calibFiveVolts, 103
 - calibTemp, 103
 - calibTime, 103
 - fpgaVersion, 103
 - getMembers, 102
 - hardwareVersion, 103
 - macAddress, 103
 - magic, 103
 - serialNumber, 103
 - spare, 104
 - spare0, 104
 - testMode, 104
 - wifiVersion, 104
- BMeasureApi::CalibrateInfo, 104
 - calibrateAmplitude, 105
 - calibrateFrequency, 105
 - calibrateTime, 105
 - getMembers, 105
 - stage, 105
 - value, 106
- BMeasureApi::ChannelConfig, 106
 - attenuator, 107
 - calibOffset, 107
 - calibScale, 107
 - calibScaleAtten1, 107
 - dataChannel, 108
 - enabled, 108
 - getMembers, 107
 - id, 108
 - name, 108
 - number, 108
 - offset, 108
 - pgaGain, 109
 - process, 109
 - sampleType, 109
 - scale, 109
 - siUnits, 109
 - spare0, 109
 - type, 109
- BMeasureApi::CommsNet, 110
 - ~CommsNet, 111
 - CommsNet, 110
 - connect, 111
 - disconnect, 111
 - init, 111
 - oinWait, 112
 - osocket, 113
 - oterminating, 113
 - read, 111
 - readAvailable, 111
 - wait, 112
 - write, 112

- writeAvailable, 112
- writeChunks, 112
- BMeasureApi::CommsSerial, 113
 - ~CommsSerial, 114
 - CommsSerial, 114
 - connect, 114
 - disconnect, 114
 - odevice, 115
 - oserialPort, 115
 - read, 114
 - readAvailable, 114
 - wait, 115
 - write, 115
- BMeasureApi::CommsUsb, 116
 - ~CommsUsb, 116
 - CommsUsb, 116
 - connect, 117
 - disconnect, 117
 - obuffer, 118
 - ocontext, 118
 - odev, 118
 - odevice, 118
 - onum, 118
 - oterminated, 119
 - oterminating, 119
 - read, 117
 - readAvailable, 117
 - readChunk, 117
 - wait, 117
 - write, 118
- BMeasureApi::ConfigItem, 119
 - getMembers, 119
 - name, 120
 - spare, 120
 - type, 120
 - value, 120
- BMeasureApi::Configuration, 120
 - alarms, 122
 - digitalMode, 122
 - digitalPins, 122
 - emailAddress, 122
 - emailMode, 123
 - getMembers, 122
 - location, 123
 - logData, 123
 - logDataDevice, 123
 - logDataMode, 123
 - mode, 123
 - mqttMode, 124
 - mqttPort, 124
 - mqttServer, 124
 - name, 124
 - networkAddress, 124
 - networkGateway, 124
 - networkMask, 125
 - networkMode, 125
 - networkNameServer0, 125
 - networkTimeServer, 125
 - program, 125
 - rs485BaudRate, 125
 - rs485Bits, 126
 - rs485Mode, 126
 - rs485StopBits, 126
 - sampleFrequencyMode, 126
 - securityMode, 126
 - source, 126
 - spare1, 127
 - spare2, 127
 - spare3, 127
 - spare4, 127
 - spare5, 127
 - spare6, 127
 - version, 127
 - wifiAp0, 128
 - wifiMode, 128
- BMeasureApi::DataBlock, 128
 - data, 129
 - getMembers, 129
 - numChannels, 129
 - numSamples, 129
 - sequence, 129
 - source, 129
 - spare, 130
 - status, 130
 - time, 130
 - type, 130
- BMeasureApi::DataBlockProc, 130
 - analogueData, 131
 - digitalData, 131
 - getMembers, 131
 - numChannels, 131
 - numSamples, 132
 - sequence, 132
 - source, 132
 - spare, 132
 - status, 132
 - time, 132
 - type, 133
- BMeasureApi::DataFile, 133
 - ~DataFile, 134
 - close, 134
 - DataFile, 134
 - getFileName, 134
 - init, 135
 - ofile, 137
 - ofilename, 137
 - offormat, 137
 - omode, 137
 - opacket, 137
 - opacketLen, 137
 - open, 135
 - readData, 135
 - readInfo, 135
 - validateFormat, 135
 - writeData, 135, 136
 - writeEnd, 136

- writeInfo, 136
- writeInfoBMeas, 136
- writeInfoCsv, 136
- writeInfoTdms, 136
- BMeasureApi::DataProc, 138
 - alarm, 138
 - getMembers, 138
 - mean, 138
 - peakHigh, 139
 - peakLow, 139
 - rms, 139
 - spare, 139
- BMeasureApi::FileData, 144
 - data, 145
 - getMembers, 144
 - length, 145
- BMeasureApi::FileInfo, 145
 - fileLength, 146
 - fileType, 146
 - getMembers, 145
 - name, 146
 - spare, 146
 - time, 146
- BMeasureApi::FilesysInfo, 147
 - free, 147
 - getMembers, 147
 - name, 147
 - size, 147
- BMeasureApi::InfoBlock, 148
 - dataType, 149
 - fileType, 149
 - getMembers, 148
 - location, 149
 - measureConfig, 149
 - name, 149
 - nodeInfo, 149
 - numChannels, 149
 - source, 150
 - time, 150
 - version, 150
- BMeasureApi::Information, 150
 - calibTime, 151
 - getMembers, 151
 - networkAddress, 152
 - networkGateway, 152
 - networkMacAddress, 152
 - networkMask, 152
 - networkMode, 152
 - networkNameServer0, 152
 - networkTimeServer, 153
 - nodeInfo, 153
 - numChannels, 153
 - numConfigItems, 153
 - spare0, 153
 - spare1, 153
 - spare2, 153
 - spare3, 154
 - time, 154
 - wifiAddress, 154
 - wifiGateway, 154
 - wifiMacAddress, 154
 - wifiMask, 154
 - wifiMode, 154
- BMeasureApi::MeasurementConfig, 155
 - description, 156
 - getMembers, 156
 - measureMode, 156
 - measureOptions, 156
 - measurePeriod, 156
 - numSamples0, 156
 - numSamples1, 156
 - numSamples2, 157
 - numSamplesBlock, 157
 - peakFilter, 157
 - sampleRate, 157
 - spare1, 157
 - spare2, 157
 - triggerChannel, 157
 - triggerConfig, 158
 - triggerDelay, 158
 - triggerLevel, 158
 - triggerMode, 158
- BMeasureApi::NodeInfo, 158
 - apiVersion, 159
 - fpgaVersion, 159
 - getMembers, 159
 - hardwareVersion, 159
 - securityMode, 159
 - serialNumber, 159
 - softwareVersion, 160
 - spare, 160
 - variant, 160
 - wifiVersion, 160
- BMeasureApi::NodeStatus, 160
 - error, 161
 - errorStr, 161
 - ethernetStatus, 161
 - getMembers, 161
 - mode, 161
 - spare, 161
 - status, 162
 - time, 162
 - wifiStatus, 162
- BMeasureApi::Version, 162
 - getMembers, 162
 - type, 163
 - ver0, 163
 - ver1, 163
 - ver2, 163
- BMeasureB-1.cpp, 167
- BMeasureB.cpp, 168
- BMeasureB.h, 168
- BMeasureD.cpp, 168
 - boffsetof, 170
- BMeasureD.h, 171
- BMeasureLib.cpp, 175

- BDEBUGL1, [175](#)
- BDEBUGL2, [175](#)
- toBStringJson, [176](#)
- BMeasureLib.h, [176](#)
- toBStringJson, [177](#)
- BMeasureS.cpp, [177](#)
- BMeasureUnit
 - BMeasureApi::BMeasureUnit, [76](#)
- BMeasureUnit.cpp, [178](#)
- BDEBUGL1, [178](#)
- BDEBUGL2, [178](#)
- BDEBUGL3, [178](#)
- CONVERT_FLOAT, [179](#)
- BMeasureUnit.h, [179](#)
- BMeasureUnit1
 - BMeasureApi::BMeasureUnit1, [81](#)
- BMeasureUnitDevice
 - BMeasureApi::BMeasureUnitDevice, [84](#)
- BMeasureUnits
 - BMeasureApi::BMeasureUnits, [87](#)
- BMeasureUnits.cpp, [179](#)
- BDEBUGL1, [180](#)
- BDEBUGL2, [180](#)
- BDEBUGL3, [180](#)
- BMeasureUnits.h, [180](#)
- BMeasureUnitsDataBlock
 - BMeasureApi::BMeasureUnitsDataBlock, [100](#)
- boffsetof
 - BMeasureD.cpp, [170](#)
- buildTime
 - BMeasureApi::BoardConfig, [102](#)
- calibAdcOffsets
 - BMeasureApi::BoardConfig, [102](#)
- calibAdcScales
 - BMeasureApi::BoardConfig, [102](#)
- calibAttenScales
 - BMeasureApi::BoardConfig, [102](#)
- calibDacOffsets
 - BMeasureApi::BoardConfig, [102](#)
- calibDacScales
 - BMeasureApi::BoardConfig, [102](#)
- calibFiveVolts
 - BMeasureApi::BoardConfig, [103](#)
- calibOffset
 - BMeasureApi::ChannelConfig, [107](#)
- calibrate
 - BMeasureApi::BMeasure, [59](#)
- calibrateAmplitude
 - BMeasureApi::CalibrateInfo, [105](#)
- calibrateFrequency
 - BMeasureApi::CalibrateInfo, [105](#)
- calibrateServe
 - BMeasureApi::BMeasure, [59](#)
- CalibrateStage
 - BMeasureApi, [21](#)
- CalibrateStageAdcOffsets
 - BMeasureApi, [21](#)
- CalibrateStageAdcScaling
 - BMeasureApi, [21](#)
- CalibrateStageAttenScaling
 - BMeasureApi, [21](#)
- CalibrateStageClear
 - BMeasureApi, [21](#)
- CalibrateStageDacOffsets
 - BMeasureApi, [21](#)
- CalibrateStageDacScaling0
 - BMeasureApi, [21](#)
- CalibrateStageDacScaling1
 - BMeasureApi, [21](#)
- CalibrateStageNone
 - BMeasureApi, [21](#)
- CalibrateStageSettle
 - BMeasureApi, [21](#)
- calibrateTime
 - BMeasureApi::CalibrateInfo, [105](#)
- CalibrationStageFiveVolts
 - BMeasureApi, [21](#)
- calibScale
 - BMeasureApi::ChannelConfig, [107](#)
- calibScaleAtten1
 - BMeasureApi::ChannelConfig, [107](#)
- calibTemp
 - BMeasureApi::BoardConfig, [103](#)
- calibTime
 - BMeasureApi::BoardConfig, [103](#)
 - BMeasureApi::Information, [151](#)
- changePassword
 - BMeasureApi::BMeasure, [59](#)
 - BMeasureApi::BMeasureUnits, [87](#)
- changePasswordServe
 - BMeasureApi::BMeasure, [59](#)
- ChannelConfigs
 - BMeasureApi, [19](#)
- ChannelType
 - BMeasureApi, [21](#)
- ChannelTypeAnalogueIn
 - BMeasureApi, [21](#)
- ChannelTypeAnalogueOut
 - BMeasureApi, [21](#)
- ChannelTypeDigitalIn
 - BMeasureApi, [21](#)
- ChannelTypeDigitalOut
 - BMeasureApi, [21](#)
- ChannelTypeNone
 - BMeasureApi, [21](#)
- channelTypeString
 - BMeasureApi, [29](#)
- checksum
 - BFirmwareInfo, [51](#)
- clear
 - BMeasureApi::BMeasureUnits, [88](#)
- clearStatus
 - Dfu, [141](#)
- close
 - BMeasureApi::DataFile, [134](#)
- CommsNet

- BMeasureApi::CommsNet, 110
- CommsNet.cpp, 181
 - BDEBUGL1, 181
 - BDEBUGL2, 181
 - BDEBUGL3, 181
- CommsNet.h, 182
- CommsSerial
 - BMeasureApi::CommsSerial, 114
- CommsSerial.cpp, 182
- CommsSerial.h, 182
- CommsUsb
 - BMeasureApi::CommsUsb, 116
- CommsUsb.cpp, 182
 - BDEBUGL1, 183
 - BDEBUGL2, 183
- CommsUsb.h, 183
- connect
 - BMeasureApi::BMeasureUnit, 76
 - BMeasureApi::CommsNet, 111
 - BMeasureApi::CommsSerial, 114
 - BMeasureApi::CommsUsb, 117
 - Dfu, 141
- CONVERT_FLOAT
 - BMeasureUnit.cpp, 179
- data
 - BMeasureApi::DataBlock, 129
 - BMeasureApi::FileData, 145
- dataAvailable
 - BMeasureApi::BMeasureUnits, 88
- dataChannel
 - BMeasureApi::ChannelConfig, 108
- dataClear
 - BMeasureApi::BMeasureUnits, 88
- dataDone
 - BMeasureApi::BMeasureUnits, 88
- dataEvent
 - BMeasureApi::BMeasureUnits, 88
- DataFile
 - BMeasureApi::DataFile, 134
- DataFile.cpp, 183
 - BDEBUGL1, 184
 - BDEBUGL2, 184
- DataFile.h, 185
- dataProcDone
 - BMeasureApi::BMeasureUnits, 88
- dataProcEvent
 - BMeasureApi::BMeasureUnits, 88
- dataProcRead
 - BMeasureApi::BMeasureUnits, 89
- dataRead
 - BMeasureApi::BMeasureUnits, 89
- DataSend
 - BMeasureApi, 22
- DataSendOff
 - BMeasureApi, 22
- DataSendProcessed
 - BMeasureApi, 22
- DataSendRaw
 - BMeasureApi, 22
- DataSendStatus
 - BMeasureApi, 22
- dataSetNumStreams
 - BMeasureApi::BMeasureUnits, 89
- dataStreamEnable
 - BMeasureApi::BMeasureUnits, 89
- DataType
 - BMeasureApi, 22
- dataType
 - BMeasureApi::InfoBlock, 149
- DataType125i
 - BMeasureApi, 22
- DataTypeFloat32
 - BMeasureApi, 22
- DataTypeProc
 - BMeasureApi, 22
- dataWait
 - BMeasureApi::BMeasureUnits, 89
- debugPrint
 - BMeasureApi::BMeasureUnits, 89
- description
 - BMeasureApi::MeasurementConfig, 156
- detectDevice
 - Dfu, 141
- device
 - BMeasureApi::BMeasureUnit, 76
 - BMeasureApi::BMeasureUnitDevice, 84
- Dfu, 139
 - ~Dfu, 140
 - clearStatus, 141
 - connect, 141
 - detectDevice, 141
 - Dfu, 140
 - disconnect, 141
 - getStatus, 141
 - init, 141
 - oconnected, 142
 - ocontext, 142
 - odev, 143
 - overbose, 143
 - reset, 141
 - upload, 142
 - upload_cmd, 142
 - validateFile, 142
- Dfu.cpp, 185
 - BDEBUGL1, 186
 - BDEBUGL2, 186
 - BFirmwareInfoEncrypt1, 193
 - BFirmwareInfoMagic, 193
 - DFU_ABORT, 187
 - DFU_CLRSTATUS, 187
 - DFU_DETACH, 187
 - DFU_DNLOAD, 187
 - DFU_GETSTATE, 187
 - DFU_GETSTATUS, 187
 - DFU_IFF_ALT, 187
 - DFU_IFF_CONFIG, 187

- DFU_IFF_DEVNUM, [188](#)
- DFU_IFF_DFU, [188](#)
- DFU_IFF_IFACE, [188](#)
- DFU_IFF_PATH, [188](#)
- DFU_IFF_PRODUCT, [188](#)
- DFU_IFF_VENDOR, [188](#)
- DFU_STATUS_ERROR_ADDRESS, [188](#)
- DFU_STATUS_ERROR_CHECK_ERASED, [188](#)
- DFU_STATUS_ERROR_ERASE, [189](#)
- DFU_STATUS_ERROR_FILE, [189](#)
- DFU_STATUS_ERROR_FIRMWARE, [189](#)
- DFU_STATUS_ERROR_NOTDONE, [189](#)
- DFU_STATUS_ERROR_POR, [189](#)
- DFU_STATUS_ERROR_PROG, [189](#)
- DFU_STATUS_ERROR_STALLEDPKT, [189](#)
- DFU_STATUS_ERROR_TARGET, [189](#)
- DFU_STATUS_ERROR_UNKNOWN, [190](#)
- DFU_STATUS_ERROR_USBR, [190](#)
- DFU_STATUS_ERROR_VENDOR, [190](#)
- DFU_STATUS_ERROR_VERIFY, [190](#)
- DFU_STATUS_ERROR_WRITE, [190](#)
- DFU_STATUS_OK, [190](#)
- DFU_UPLOAD, [190](#)
- dfuse_command, [192](#)
- ERASE_PAGE, [192](#)
- MASS_ERASE, [192](#)
- pageAddress, [192](#)
- pageNumber, [192](#)
- READ_UNPROTECT, [192](#)
- SET_ADDRESS, [192](#)
- STATE_APP_DETACH, [190](#)
- STATE_APP_IDLE, [191](#)
- STATE_DFU_DOWNLOAD_BUSY, [191](#)
- STATE_DFU_DOWNLOAD_IDLE, [191](#)
- STATE_DFU_DOWNLOAD_SYNC, [191](#)
- STATE_DFU_ERROR, [191](#)
- STATE_DFU_IDLE, [191](#)
- STATE_DFU_MANIFEST, [191](#)
- STATE_DFU_MANIFEST_SYNC, [191](#)
- STATE_DFU_MANIFEST_WAIT_RESET, [192](#)
- STATE_DFU_UPLOAD_IDLE, [192](#)
- Dfu.h, [193](#)
- DFU_ABORT
 - Dfu.cpp, [187](#)
- DFU_CLRSTATUS
 - Dfu.cpp, [187](#)
- DFU_DETACH
 - Dfu.cpp, [187](#)
- DFU_DNLOAD
 - Dfu.cpp, [187](#)
- DFU_GETSTATE
 - Dfu.cpp, [187](#)
- DFU_GETSTATUS
 - Dfu.cpp, [187](#)
- DFU_IFF_ALT
 - Dfu.cpp, [187](#)
- DFU_IFF_CONFIG
 - Dfu.cpp, [187](#)
- DFU_IFF_DEVNUM
 - Dfu.cpp, [188](#)
- DFU_IFF_DFU
 - Dfu.cpp, [188](#)
- DFU_IFF_IFACE
 - Dfu.cpp, [188](#)
- DFU_IFF_PATH
 - Dfu.cpp, [188](#)
- DFU_IFF_PRODUCT
 - Dfu.cpp, [188](#)
- DFU_IFF_VENDOR
 - Dfu.cpp, [188](#)
- DFU_STATUS_ERROR_ADDRESS
 - Dfu.cpp, [188](#)
- DFU_STATUS_ERROR_CHECK_ERASED
 - Dfu.cpp, [188](#)
- DFU_STATUS_ERROR_ERASE
 - Dfu.cpp, [189](#)
- DFU_STATUS_ERROR_FILE
 - Dfu.cpp, [189](#)
- DFU_STATUS_ERROR_FIRMWARE
 - Dfu.cpp, [189](#)
- DFU_STATUS_ERROR_NOTDONE
 - Dfu.cpp, [189](#)
- DFU_STATUS_ERROR_POR
 - Dfu.cpp, [189](#)
- DFU_STATUS_ERROR_PROG
 - Dfu.cpp, [189](#)
- DFU_STATUS_ERROR_STALLEDPKT
 - Dfu.cpp, [189](#)
- DFU_STATUS_ERROR_TARGET
 - Dfu.cpp, [189](#)
- DFU_STATUS_ERROR_UNKNOWN
 - Dfu.cpp, [190](#)
- DFU_STATUS_ERROR_USBR
 - Dfu.cpp, [190](#)
- DFU_STATUS_ERROR_VENDOR
 - Dfu.cpp, [190](#)
- DFU_STATUS_ERROR_VERIFY
 - Dfu.cpp, [190](#)
- DFU_STATUS_ERROR_WRITE
 - Dfu.cpp, [190](#)
- DFU_STATUS_OK
 - Dfu.cpp, [190](#)
- DFU_UPLOAD
 - Dfu.cpp, [190](#)
- dfuse_command
 - Dfu.cpp, [192](#)
- DfuStatus, [143](#)
 - iString, [143](#)
 - pollTimeout, [143](#)
 - state, [144](#)
 - status, [144](#)
- digitalData
 - BMeasureApi::DataBlockProc, [131](#)
- DigitalMode
 - BMeasureApi, [22](#)
- digitalMode

- BMeasureApi::Configuration, [122](#)
- DigitalModelInOut
 - BMeasureApi, [22](#)
- DigitalModelInput
 - BMeasureApi, [22](#)
- DigitalModeOutput
 - BMeasureApi, [22](#)
- DigitalModeSyncMaster
 - BMeasureApi, [22](#)
- DigitalModeSyncSlave
 - BMeasureApi, [22](#)
- digitalPins
 - BMeasureApi::Configuration, [122](#)
- disconnect
 - BMeasureApi::BMeasureUnit, [77](#)
 - BMeasureApi::CommsNet, [111](#)
 - BMeasureApi::CommsSerial, [114](#)
 - BMeasureApi::CommsUsb, [117](#)
 - Dfu, [141](#)
- disconnected
 - BMeasureApi::BMeasureUnit, [77](#)
 - BMeasureApi::BMeasureUnit1, [82](#)
 - BMeasureApi::BMeasureUnits, [90](#)
- duty
 - BMeasureApi::AwgConfig, [50](#)
- emailAddress
 - BMeasureApi::Configuration, [122](#)
- emailMode
 - BMeasureApi::Configuration, [123](#)
- enabled
 - BMeasureApi::ChannelConfig, [108](#)
- ERASE_PAGE
 - Dfu.cpp, [192](#)
- error
 - BMeasureApi::NodeStatus, [161](#)
- ErrorNum
 - BMeasureApi, [22](#)
- ErrorNumDataOverrun
 - BMeasureApi, [23](#)
- ErrorNumSystem
 - BMeasureApi, [23](#)
- ErrorNumToFast
 - BMeasureApi, [23](#)
- errorStr
 - BMeasureApi::NodeStatus, [161](#)
- ethernetStatus
 - BMeasureApi::NodeStatus, [161](#)
- EventMode
 - BMeasureApi, [23](#)
- EventModeAlarm
 - BMeasureApi, [23](#)
- EventModeOff
 - BMeasureApi, [23](#)
- EventModeSecond
 - BMeasureApi, [23](#)
- extra
 - BMdnsService, [54](#)
- factoryReset
 - BMeasureApi::BMeasure, [59](#)
- factoryResetServe
 - BMeasureApi::BMeasure, [60](#)
- fileClose
 - BMeasureApi::BMeasure, [60](#)
- fileCloseServe
 - BMeasureApi::BMeasure, [60](#)
- fileDelete
 - BMeasureApi::BMeasure, [60](#)
- fileDeleteServe
 - BMeasureApi::BMeasure, [60](#)
- fileLength
 - BMeasureApi::FileInfo, [146](#)
- fileList
 - BMeasureApi::BMeasure, [60](#)
- fileListServe
 - BMeasureApi::BMeasure, [61](#)
- fileOpen
 - BMeasureApi::BMeasure, [61](#)
- fileOpenServe
 - BMeasureApi::BMeasure, [61](#)
- fileRead
 - BMeasureApi::BMeasure, [61](#)
- fileReadServe
 - BMeasureApi::BMeasure, [61](#)
- filesysDelete
 - BMeasureApi::BMeasure, [62](#)
- filesysDeleteServe
 - BMeasureApi::BMeasure, [62](#)
- FilesysDeleteType
 - BMeasureApi, [23](#)
- FilesysDeleteTypeData
 - BMeasureApi, [23](#)
- FilesysDeleteTypeFormat
 - BMeasureApi, [23](#)
- FilesysDeleteTypeNone
 - BMeasureApi, [23](#)
- filesysInfo
 - BMeasureApi::BMeasure, [62](#)
- filesysInfoServe
 - BMeasureApi::BMeasure, [62](#)
- FileType
 - BMeasureApi, [23](#)
- fileType
 - BMeasureApi::FileInfo, [146](#)
 - BMeasureApi::InfoBlock, [149](#)
- FileTypeDir
 - BMeasureApi, [23](#)
- FileTypeFile
 - BMeasureApi, [23](#)
- FileTypeNone
 - BMeasureApi, [23](#)
- fileWrite
 - BMeasureApi::BMeasure, [62](#)
- fileWriteServe
 - BMeasureApi::BMeasure, [62](#)
- findDevices

- BMeasureApi::BMeasureUnit, 77
- findDevicesNetwork
 - BMeasureApi::BMeasureUnit, 77
- findDevicesUsb
 - BMeasureApi::BMeasureUnit, 77
- findServices
 - BMdns, 53
- fpgaVersion
 - BMeasureApi::BoardConfig, 103
 - BMeasureApi::NodeInfo, 159
- free
 - BMeasureApi::FilesysInfo, 147
- frequency
 - BMeasureApi::AwgConfig, 50
- fromBString
 - BMeasureApi, 29–34
- functionUnLock
 - BMeasureApi::BMeasure, 63
- functionUnLockServe
 - BMeasureApi::BMeasure, 63
- getAwgConfig
 - BMeasureApi::BMeasure, 63
 - BMeasureApi::BMeasureUnits, 90
- getAwgConfigServe
 - BMeasureApi::BMeasure, 63
- getBoardConfig
 - BMeasureApi::BMeasure, 63
- getBoardConfigServe
 - BMeasureApi::BMeasure, 63
- getChannelConfig
 - BMeasureApi::BMeasure, 64
 - BMeasureApi::BMeasureUnits, 90
- getChannelConfigServe
 - BMeasureApi::BMeasure, 64
- getConfig
 - BMeasureApi::BMeasure, 64
 - BMeasureApi::BMeasureUnits, 90
- getConfigServe
 - BMeasureApi::BMeasure, 64
- getDigital
 - BMeasureApi::BMeasure, 64
- getDigitalServe
 - BMeasureApi::BMeasure, 64
- getFileName
 - BMeasureApi::DataFile, 134
- getFreeBlock
 - BMeasureApi::BMeasureUnits, 90
- getInfoBlock
 - BMeasureApi::BMeasure, 65
 - BMeasureApi::BMeasureUnits, 90
- getInfoBlockServe
 - BMeasureApi::BMeasure, 65
- getInformation
 - BMeasureApi::BMeasure, 65
 - BMeasureApi::BMeasureUnits, 91
- getInformationServe
 - BMeasureApi::BMeasure, 65
- getMeasurementConfig
 - BMeasureApi::BMeasure, 65
 - BMeasureApi::BMeasureUnits, 91
- getMeasurementConfigServe
 - BMeasureApi::BMeasure, 65
- getMembers
 - BMeasureApi::AlarmConfig, 47
 - BMeasureApi::AwgConfig, 49
 - BMeasureApi::BoardConfig, 102
 - BMeasureApi::CalibrateInfo, 105
 - BMeasureApi::ChannelConfig, 107
 - BMeasureApi::ConfigItem, 119
 - BMeasureApi::Configuration, 122
 - BMeasureApi::DataBlock, 129
 - BMeasureApi::DataBlockProc, 131
 - BMeasureApi::DataProc, 138
 - BMeasureApi::FileData, 144
 - BMeasureApi::FileInfo, 145
 - BMeasureApi::FilesysInfo, 147
 - BMeasureApi::InfoBlock, 148
 - BMeasureApi::Information, 151
 - BMeasureApi::MeasurementConfig, 156
 - BMeasureApi::NodeInfo, 159
 - BMeasureApi::NodeStatus, 161
 - BMeasureApi::Version, 162
- getNodeInfo
 - BMeasureApi::BMeasure, 66
 - BMeasureApi::BMeasureUnits, 91
- getNodeInfoServe
 - BMeasureApi::BMeasure, 66
- getStatus
 - BMeasureApi::BMeasure, 66
 - BMeasureApi::BMeasureUnits, 91
 - Dfu, 141
- getStatusServe
 - BMeasureApi::BMeasure, 66
- getSwitch
 - BMeasureApi::BMeasure, 66
- getSwitchServe
 - BMeasureApi::BMeasure, 66
- hardwareVersion
 - BMeasureApi::BoardConfig, 103
 - BMeasureApi::NodeInfo, 159
- hostname
 - BMdnsService, 55
- id
 - BMeasureApi::ChannelConfig, 108
- info
 - BMeasureApi::BMeasureUnit, 77
- init
 - BMdns, 53
 - BMeasureApi::BMeasureUnitsDataBlock, 100
 - BMeasureApi::CommsNet, 111
 - BMeasureApi::DataFile, 135
 - Dfu, 141
- iString
 - DfuStatus, 143

- length
 - BFirmwareInfo, 52
 - BMeasureApi::FileData, 145
- levelHigh
 - BMeasureApi::AlarmConfig, 48
- levelLow
 - BMeasureApi::AlarmConfig, 48
- location
 - BMeasureApi::Configuration, 123
 - BMeasureApi::InfoBlock, 149
- LogData
 - BMeasureApi, 23
- logData
 - BMeasureApi::Configuration, 123
- logDataDevice
 - BMeasureApi::Configuration, 123
- LogDataMode
 - BMeasureApi, 24
- logDataMode
 - BMeasureApi::Configuration, 123
- LogDataModeDeleteOld
 - BMeasureApi, 24
- LogDataModeNormal
 - BMeasureApi, 24
- LogDataOff
 - BMeasureApi, 24
- LogDataProcessed
 - BMeasureApi, 24
- LogDataRaw
 - BMeasureApi, 24
- login
 - BMeasureApi::BMeasure, 67
 - BMeasureApi::BMeasureUnits, 91
- loginServe
 - BMeasureApi::BMeasure, 67
- logout
 - BMeasureApi::BMeasure, 67
 - BMeasureApi::BMeasureUnits, 91
- logoutServe
 - BMeasureApi::BMeasure, 67
- macAddress
 - BMeasureApi::BoardConfig, 103
- magic
 - BFirmwareInfo, 52
 - BMeasureApi::BoardConfig, 103
- MASS_ERASE
 - Dfu.cpp, 192
- MDNS_CLASS_IN
 - BMdns.cpp, 166
- MDNS_ENTRYTYPE_ADDITIONAL
 - BMdns.cpp, 166
- MDNS_ENTRYTYPE_ANSWER
 - BMdns.cpp, 166
- MDNS_ENTRYTYPE_AUTHORITY
 - BMdns.cpp, 166
- mdns_read_string
 - BMdns.cpp, 167
- mdns_read_strings
 - BMdns.cpp, 167
- MDNS_RECORDTYPE_A
 - BMdns.cpp, 166
- MDNS_RECORDTYPE_AAAA
 - BMdns.cpp, 166
- MDNS_RECORDTYPE_IGNORE
 - BMdns.cpp, 166
- MDNS_RECORDTYPE_PTR
 - BMdns.cpp, 166
- MDNS_RECORDTYPE_SRV
 - BMdns.cpp, 166
- MDNS_RECORDTYPE_TXT
 - BMdns.cpp, 166
- mdns_write_string
 - BMdns.cpp, 167
- MdnsClass
 - BMdns.cpp, 166
- MdnsEntryType
 - BMdns.cpp, 166
- MdnsRecordType
 - BMdns.cpp, 166
- mean
 - BMeasureApi::DataProc, 138
- measure
 - BMeasureApi::BMeasure, 67
- measureConfig
 - BMeasureApi::InfoBlock, 149
- MeasureMode
 - BMeasureApi, 24
- measureMode
 - BMeasureApi::MeasurementConfig, 156
- MeasureModeContinuous
 - BMeasureApi, 24
- MeasureModeOff
 - BMeasureApi, 24
- MeasureModeOneShot
 - BMeasureApi, 24
- MeasureModeRepeat
 - BMeasureApi, 24
- MeasureOption
 - BMeasureApi, 24
- MeasureOptionNone
 - BMeasureApi, 24
- MeasureOptionProcess
 - BMeasureApi, 24
- measureOptions
 - BMeasureApi::MeasurementConfig, 156
- measurePeriod
 - BMeasureApi::MeasurementConfig, 156
- measureServe
 - BMeasureApi::BMeasure, 67
- MessageSource
 - BMeasureApi, 24
- MessageSourceDebug
 - BMeasureApi, 25
- MessageSourceGeneral
 - BMeasureApi, 25
- MessageSourceTest

- BMeasureApi, 25
- MessageSourceWifi
 - BMeasureApi, 25
- MessageSourceWifiTest
 - BMeasureApi, 25
- Mode
 - BMeasureApi, 25
- mode
 - BMeasureApi::AlarmConfig, 48
 - BMeasureApi::AwgConfig, 50
 - BMeasureApi::Configuration, 123
 - BMeasureApi::NodeStatus, 161
- ModeDemo1
 - BMeasureApi, 25
- ModelIdle
 - BMeasureApi, 25
- ModelInternal
 - BMeasureApi, 25
- ModeRun
 - BMeasureApi, 25
- ModeRunProgram
 - BMeasureApi, 25
- ModeSleep
 - BMeasureApi, 25
- mqttMode
 - BMeasureApi::Configuration, 124
- mqttPort
 - BMeasureApi::Configuration, 124
- mqttServer
 - BMeasureApi::Configuration, 124
- name
 - BmdnsService, 55
 - BMeasureApi::ChannelConfig, 108
 - BMeasureApi::ConfigItem, 120
 - BMeasureApi::Configuration, 124
 - BMeasureApi::FileInfo, 146
 - BMeasureApi::FilesysInfo, 147
 - BMeasureApi::InfoBlock, 149
- networkAddress
 - BMeasureApi::Configuration, 124
 - BMeasureApi::Information, 152
- networkGateway
 - BMeasureApi::Configuration, 124
 - BMeasureApi::Information, 152
- networkMacAddress
 - BMeasureApi::Information, 152
- networkMask
 - BMeasureApi::Configuration, 125
 - BMeasureApi::Information, 152
- NetworkMode
 - BMeasureApi, 25
- networkMode
 - BMeasureApi::Configuration, 125
 - BMeasureApi::Information, 152
- NetworkModeDhcp
 - BMeasureApi, 25
- NetworkModeManual
 - BMeasureApi, 25
- NetworkModeOff
 - BMeasureApi, 25
- networkNameServer0
 - BMeasureApi::Configuration, 125
 - BMeasureApi::Information, 152
- networkTimeServer
 - BMeasureApi::Configuration, 125
 - BMeasureApi::Information, 153
- nodeInfo
 - BMeasureApi::InfoBlock, 149
 - BMeasureApi::Information, 153
- NodeType
 - BMeasureApi, 25
- NodeTypeBMeasure1
 - BMeasureApi, 26
- NodeTypeNone
 - BMeasureApi, 26
- number
 - BMeasureApi::ChannelConfig, 108
- numChannels
 - BMeasureApi::BMeasureUnit, 78
 - BMeasureApi::BMeasureUnits, 92
 - BMeasureApi::DataBlock, 129
 - BMeasureApi::DataBlockProc, 131
 - BMeasureApi::InfoBlock, 149
 - BMeasureApi::Information, 153
- numConfigItems
 - BMeasureApi::Information, 153
- numSamples
 - BMeasureApi::AwgConfig, 50
 - BMeasureApi::DataBlock, 129
 - BMeasureApi::DataBlockProc, 132
- numSamples0
 - BMeasureApi::MeasurementConfig, 156
- numSamples1
 - BMeasureApi::MeasurementConfig, 156
- numSamples2
 - BMeasureApi::MeasurementConfig, 157
- numSamplesBlock
 - BMeasureApi::MeasurementConfig, 157
- oblockCount
 - BMeasureApi::BMeasureUnit, 79
- obuffer
 - BMeasureApi::CommsUsb, 118
- ochannels
 - BMeasureApi::BMeasureUnit, 79
- oconfigMeasurement
 - BMeasureApi::BMeasureUnit, 79
- oconnected
 - BMeasureApi::BMeasureUnit1, 83
 - Dfu, 142
- ocontext
 - BMeasureApi::CommsUsb, 118
 - Dfu, 142
- odataBlock
 - BMeasureApi::BMeasureUnit, 80
 - BMeasureApi::BMeasureUnitsDataBlock, 100
- odataBlocksFree

- BMeasureApi::BMeasureUnits, 96
- odataBlocksIn
 - BMeasureApi::BMeasureUnits, 96
- odataBlocksOut
 - BMeasureApi::BMeasureUnits, 97
- odataBlocksOutCount
 - BMeasureApi::BMeasureUnits, 97
- odataBlocksProcess
 - BMeasureApi::BMeasureUnits, 97
- odataBlocksProcessNum
 - BMeasureApi::BMeasureUnits, 97
- odataProcBlocks
 - BMeasureApi::BMeasureUnits, 97
- odataStreamNum
 - BMeasureApi::BMeasureUnits, 97
- odev
 - BMeasureApi::CommsUsb, 118
 - Dfu, 143
- odevice
 - BMeasureApi::BMeasureUnit, 80
 - BMeasureApi::CommsSerial, 115
 - BMeasureApi::CommsUsb, 118
- odisconnecting
 - BMeasureApi::BMeasureUnit, 80
- oenabled
 - BMeasureApi::BMeasureUnit1, 83
- offset
 - BMeasureApi::AwgConfig, 50
 - BMeasureApi::ChannelConfig, 108
- ofile
 - BMeasureApi::DataFile, 137
- ofileName
 - BMeasureApi::DataFile, 137
- ofill
 - BMeasureApi::BMeasureUnits, 97
 - BMeasureApi::BMeasureUnitsDataBlock, 101
- oformat
 - BMeasureApi::DataFile, 137
- oinfo
 - BMeasureApi::BMeasureUnit, 80
- oinUse
 - BMeasureApi::BMeasureUnitsDataBlock, 101
- oinWait
 - BMeasureApi::CommsNet, 112
- olocalTrigger
 - BMeasureApi::BMeasureUnits, 97
- oclockInput
 - BMeasureApi::BMeasureUnits, 98
- oclockOutput
 - BMeasureApi::BMeasureUnits, 98
- oclockProInput
 - BMeasureApi::BMeasureUnits, 98
- oclockUnits
 - BMeasureApi::BMeasureUnits, 98
- omeasureUnits
 - BMeasureApi::BMeasureUnit1, 83
- omode
 - BMeasureApi::DataFile, 137
- onodeInfo
 - BMeasureApi::BMeasureUnit, 80
- onum
 - BMeasureApi::CommsUsb, 118
- onumBlocks
 - BMeasureApi::BMeasureUnits, 98
- onumChannels
 - BMeasureApi::BMeasureUnits, 98
- onumConnected
 - BMeasureApi::BMeasureUnits, 98
- oorder
 - BMeasureApi::BMeasureUnit1, 83
- opacket
 - BMeasureApi::DataFile, 137
- opacketLen
 - BMeasureApi::DataFile, 137
- open
 - BMeasureApi::DataFile, 135
- oprocEnable
 - BMeasureApi::BMeasureUnits, 98
- oprocRunning
 - BMeasureApi::BMeasureUnits, 99
- osampleCount
 - BMeasureApi::BMeasureUnit, 80
- osequenceNext
 - BMeasureApi::BMeasureUnit, 80
- oserialNumber
 - BMeasureApi::BMeasureUnit1, 83
- oserialPort
 - BMeasureApi::CommsSerial, 115
- osocket
 - BMdns, 54
 - BMeasureApi::CommsNet, 113
- osource
 - BMeasureApi::BMeasureUnit1, 83
- ostartSample
 - BMeasureApi::BMeasureUnits, 99
- oterminated
 - BMeasureApi::CommsUsb, 119
- oterminating
 - BMeasureApi::CommsNet, 113
 - BMeasureApi::CommsUsb, 119
- otransactionId
 - BMdns, 54
- otriggered
 - BMeasureApi::BMeasureUnits, 99
- ounitMaster
 - BMeasureApi::BMeasureUnits, 99
- ounits
 - BMeasureApi::BMeasureUnits, 99
- output
 - BMeasureApi::AlarmConfig, 48
 - BMeasureApi::AwgConfig, 51
- outputBlock
 - BMeasureApi::BMeasureUnits, 92
- outputChannel
 - BMeasureApi::AlarmConfig, 48
- overbose

- Dfu, [143](#)
- overview.dox, [193](#)
- pageAddress
 - Dfu.cpp, [192](#)
- pageNumber
 - Dfu.cpp, [192](#)
- peakFilter
 - BMeasureApi::MeasurementConfig, [157](#)
- peakHigh
 - BMeasureApi::DataProc, [139](#)
- peakLow
 - BMeasureApi::DataProc, [139](#)
- pgaGain
 - BMeasureApi::ChannelConfig, [109](#)
- pollTimeout
 - DfuStatus, [143](#)
- process
 - BMeasureApi::ChannelConfig, [109](#)
- processdataBlock
 - BMeasureApi::BMeasureUnit, [78](#)
- processRequest
 - BMeasureApi::BMeasure, [68](#)
- program
 - BMeasureApi::Configuration, [125](#)
- read
 - BMeasureApi::CommsNet, [111](#)
 - BMeasureApi::CommsSerial, [114](#)
 - BMeasureApi::CommsUsb, [117](#)
- READ_UNPROTECT
 - Dfu.cpp, [192](#)
- readAvailable
 - BMeasureApi::CommsNet, [111](#)
 - BMeasureApi::CommsSerial, [114](#)
 - BMeasureApi::CommsUsb, [117](#)
- readChunk
 - BMeasureApi::CommsUsb, [117](#)
- readData
 - BMeasureApi::DataFile, [135](#)
- readInfo
 - BMeasureApi::DataFile, [135](#)
- reset
 - Dfu, [141](#)
- rms
 - BMeasureApi::DataProc, [139](#)
- round512
 - BMeasureApi, [34](#)
- roundDown512
 - BMeasureApi, [34](#)
- rs485BaudRate
 - BMeasureApi::Configuration, [125](#)
- rs485Bits
 - BMeasureApi::Configuration, [126](#)
- Rs485Mode
 - BMeasureApi, [26](#)
- rs485Mode
 - BMeasureApi::Configuration, [126](#)
- Rs485ModeBoap
 - BMeasureApi, [26](#)
- Rs485ModeOff
 - BMeasureApi, [26](#)
- rs485StopBits
 - BMeasureApi::Configuration, [126](#)
- run
 - BMeasureApi::BMeasureUnit, [78](#)
 - BMeasureApi::BMeasureUnits, [92](#)
- runBoardTest
 - BMeasureApi::BMeasure, [68](#)
- runBoardTestServe
 - BMeasureApi::BMeasure, [68](#)
- sampleFrequencyMode
 - BMeasureApi::Configuration, [126](#)
- sampleRate
 - BMeasureApi::MeasurementConfig, [157](#)
- SampleType
 - BMeasureApi, [26](#)
- sampleType
 - BMeasureApi::ChannelConfig, [109](#)
- SampleTypeBool
 - BMeasureApi, [26](#)
- SampleTypeFloat32
 - BMeasureApi, [26](#)
- SampleTypeFloat64
 - BMeasureApi, [26](#)
- SampleTypeInt16
 - BMeasureApi, [26](#)
- SampleTypeInt32
 - BMeasureApi, [26](#)
- SampleTypeInt8
 - BMeasureApi, [26](#)
- SampleTypeNone
 - BMeasureApi, [26](#)
- sampleTypeString
 - BMeasureApi, [34](#)
- scale
 - BMeasureApi::ChannelConfig, [109](#)
- SecurityMode
 - BMeasureApi, [26](#)
- securityMode
 - BMeasureApi::Configuration, [126](#)
 - BMeasureApi::NodeInfo, [159](#)
- SecurityModeBasic
 - BMeasureApi, [26](#)
- SecurityModeConfig
 - BMeasureApi, [26](#)
- SecurityModeFull
 - BMeasureApi, [26](#)
- sendChannelConfig
 - BMeasureApi::BMeasure, [68](#)
- sendChannelConfigServe
 - BMeasureApi::BMeasure, [68](#)
- sendData
 - BMeasureApi::BMeasure, [68](#)
- sendDataEnable
 - BMeasureApi::BMeasure, [69](#)
 - BMeasureApi::BMeasureUnits, [92](#)

- sendDataEnableServe
 - BMeasureApi::BMeasure, 69
- sendDataProcess
 - BMeasureApi::BMeasureUnits, 92
- sendDataProcessTrigger
 - BMeasureApi::BMeasureUnits, 92
- sendDataProcQueue
 - BMeasureApi::BMeasureUnits, 93
- sendDataQueue
 - BMeasureApi::BMeasureUnits, 93
- sendDataServe
 - BMeasureApi::BMeasure, 69
 - BMeasureApi::BMeasureUnit, 78
- sendDataServe1
 - BMeasureApi::BMeasureUnit, 78
 - BMeasureApi::BMeasureUnit1, 82
 - BMeasureApi::BMeasureUnits, 93
- sendInfo
 - BMeasureApi::BMeasure, 69
- sendInfoServe
 - BMeasureApi::BMeasure, 69
- sendMessage
 - BMeasureApi::BMeasure, 69
 - BMeasureApi::BMeasureUnits, 93
- sendMessageServe
 - BMeasureApi::BMeasure, 70
 - BMeasureApi::BMeasureUnit1, 82
 - BMeasureApi::BMeasureUnits, 93
- sendStatus
 - BMeasureApi::BMeasure, 70
- sendStatusServe
 - BMeasureApi::BMeasure, 70
- sendTime
 - BMeasureApi::BMeasure, 70
 - BMeasureApi::BMeasureUnits, 93
- sendTimeServe
 - BMeasureApi::BMeasure, 70
- sequence
 - BMeasureApi::DataBlock, 129
 - BMeasureApi::DataBlockProc, 132
- serialNumber
 - BMeasureApi::BMeasureUnit, 78
 - BMeasureApi::BMeasureUnit1, 82
 - BMeasureApi::BMeasureUnitDevice, 84
 - BMeasureApi::BoardConfig, 103
 - BMeasureApi::NodeInfo, 159
- SET_ADDRESS
 - Dfu.cpp, 192
- setAnalogueOut
 - BMeasureApi::BMeasure, 70
- setAnalogueOutServe
 - BMeasureApi::BMeasure, 71
- setAwgConfig
 - BMeasureApi::BMeasure, 71
 - BMeasureApi::BMeasureUnits, 93
- setAwgConfigServe
 - BMeasureApi::BMeasure, 71
- setAwgWaveform
 - BMeasureApi::BMeasure, 71
- setAwgWaveformServe
 - BMeasureApi::BMeasure, 71
- setBoardConfig
 - BMeasureApi::BMeasure, 72
- setBoardConfigServe
 - BMeasureApi::BMeasure, 72
- setChannelConfig
 - BMeasureApi::BMeasure, 72
 - BMeasureApi::BMeasureUnit, 79
 - BMeasureApi::BMeasureUnits, 94
- setChannelConfigFull
 - BMeasureApi::BMeasure, 72
- setChannelConfigFullServe
 - BMeasureApi::BMeasure, 72
- setChannelConfigServe
 - BMeasureApi::BMeasure, 72
- setConfig
 - BMeasureApi::BMeasure, 73
 - BMeasureApi::BMeasureUnits, 94
- setConfigServe
 - BMeasureApi::BMeasure, 73
- setDigital
 - BMeasureApi::BMeasure, 73
- setDigitalServe
 - BMeasureApi::BMeasure, 73
- setMeasurementConfig
 - BMeasureApi::BMeasure, 73
 - BMeasureApi::BMeasureUnit, 79
 - BMeasureApi::BMeasureUnits, 94
- setMeasurementConfigServe
 - BMeasureApi::BMeasure, 73
- setMode
 - BMeasureApi::BMeasure, 74
 - BMeasureApi::BMeasureUnits, 94
- setModeServe
 - BMeasureApi::BMeasure, 74
- setRelay
 - BMeasureApi::BMeasure, 74
- setRelayServe
 - BMeasureApi::BMeasure, 74
- setSerialNumber
 - BMeasureApi::BMeasureUnit1, 82
- siUnits
 - BMeasureApi::ChannelConfig, 109
- size
 - BMeasureApi::FilesysInfo, 147
- softwareVersion
 - BMeasureApi::NodeInfo, 160
- source
 - BMeasureApi::Configuration, 126
 - BMeasureApi::DataBlock, 129
 - BMeasureApi::DataBlockProc, 132
 - BMeasureApi::InfoBlock, 150
- spare
 - BMeasureApi::AlarmConfig, 48
 - BMeasureApi::AwgConfig, 51
 - BMeasureApi::BoardConfig, 104

- BMeasureApi::ConfigItem, 120
- BMeasureApi::DataBlock, 130
- BMeasureApi::DataBlockProc, 132
- BMeasureApi::DataProc, 139
- BMeasureApi::FileInfo, 146
- BMeasureApi::NodeInfo, 160
- BMeasureApi::NodeStatus, 161
- spare0
 - BMeasureApi::BoardConfig, 104
 - BMeasureApi::ChannelConfig, 109
 - BMeasureApi::Information, 153
- spare1
 - BMeasureApi::Configuration, 127
 - BMeasureApi::Information, 153
 - BMeasureApi::MeasurementConfig, 157
- spare2
 - BMeasureApi::Configuration, 127
 - BMeasureApi::Information, 153
 - BMeasureApi::MeasurementConfig, 157
- spare3
 - BMeasureApi::Configuration, 127
 - BMeasureApi::Information, 154
- spare4
 - BMeasureApi::Configuration, 127
- spare5
 - BMeasureApi::Configuration, 127
- spare6
 - BMeasureApi::Configuration, 127
- stage
 - BMeasureApi::CalibrateInfo, 105
- state
 - DfuStatus, 144
- STATE_APP_DETACH
 - Dfu.cpp, 190
- STATE_APP_IDLE
 - Dfu.cpp, 191
- STATE_DFU_DOWNLOAD_BUSY
 - Dfu.cpp, 191
- STATE_DFU_DOWNLOAD_IDLE
 - Dfu.cpp, 191
- STATE_DFU_DOWNLOAD_SYNC
 - Dfu.cpp, 191
- STATE_DFU_ERROR
 - Dfu.cpp, 191
- STATE_DFU_IDLE
 - Dfu.cpp, 191
- STATE_DFU_MANIFEST
 - Dfu.cpp, 191
- STATE_DFU_MANIFEST_SYNC
 - Dfu.cpp, 191
- STATE_DFU_MANIFEST_WAIT_RESET
 - Dfu.cpp, 192
- STATE_DFU_UPLOAD_IDLE
 - Dfu.cpp, 192
- Status
 - BMeasureApi, 26
- status
 - BMeasureApi::DataBlock, 130
 - BMeasureApi::DataBlockProc, 132
 - BMeasureApi::NodeStatus, 162
 - DfuStatus, 144
 - StatusAlarm
 - BMeasureApi, 27
 - StatusDataOverrun
 - BMeasureApi, 27
 - StatusEnd0
 - BMeasureApi, 27
 - StatusEnd1
 - BMeasureApi, 27
 - StatusError
 - BMeasureApi, 27
 - StatusFpgaOverrun
 - BMeasureApi, 27
 - StatusNone
 - BMeasureApi, 27
 - StatusRun
 - BMeasureApi, 27
 - StatusTriggerWait
 - BMeasureApi, 27
 - StatusWarning
 - BMeasureApi, 27
 - SyncMode
 - BMeasureApi, 27
 - SyncModeMaster
 - BMeasureApi, 27
 - SyncModeOff
 - BMeasureApi, 27
 - SyncModeSlave
 - BMeasureApi, 27
- TdsDataType
 - BMeasureApi, 27
- TdsTypeBoolean
 - BMeasureApi, 28
- TdsTypeComplexDoubleFloat
 - BMeasureApi, 28
- TdsTypeComplexSingleFloat
 - BMeasureApi, 28
- TdsTypeDAQmxRawData
 - BMeasureApi, 28
- TdsTypeDoubleFloat
 - BMeasureApi, 27
- TdsTypeDoubleFloatWithUnit
 - BMeasureApi, 28
- TdsTypeExtendedFloat
 - BMeasureApi, 28
- TdsTypeExtendedFloatWithUnit
 - BMeasureApi, 28
- TdsTypeFixedPoint
 - BMeasureApi, 28
- TdsType16
 - BMeasureApi, 27
- TdsType32
 - BMeasureApi, 27
- TdsType64
 - BMeasureApi, 27
- TdsType8
 - BMeasureApi, 27

- BMeasureApi, 27
- TdsTypeSingleFloat
 - BMeasureApi, 27
- TdsTypeSingleFloatWithUnit
 - BMeasureApi, 28
- TdsTypeString
 - BMeasureApi, 28
- TdsTypeTimeStamp
 - BMeasureApi, 28
- TdsTypeU16
 - BMeasureApi, 27
- TdsTypeU32
 - BMeasureApi, 27
- TdsTypeU64
 - BMeasureApi, 27
- TdsTypeU8
 - BMeasureApi, 27
- TdsTypeVoid
 - BMeasureApi, 27
- testMode
 - BMeasureApi::BoardConfig, 104
- time
 - BMeasureApi::DataBlock, 130
 - BMeasureApi::DataBlockProc, 132
 - BMeasureApi::FileInfo, 146
 - BMeasureApi::InfoBlock, 150
 - BMeasureApi::Information, 154
 - BMeasureApi::NodeStatus, 162
- toBString
 - BMeasureApi, 34–38
- toBStringJson
 - BMeasureApi, 39–44
 - BMeasureLib.cpp, 176
 - BMeasureLib.h, 177
- TocBigEndian
 - BMeasureApi, 44
- TocDaqRawData
 - BMeasureApi, 44
- TocInterleavedData
 - BMeasureApi, 44
- TocMetaData
 - BMeasureApi, 44
- TocNewObjList
 - BMeasureApi, 44
- TocRawData
 - BMeasureApi, 44
- toFloat
 - BMeasureApi, 45
- trackChannel
 - BMeasureApi::AwgConfig, 51
- triggerChannel
 - BMeasureApi::MeasurementConfig, 157
- TriggerConfig
 - BMeasureApi, 28
- triggerConfig
 - BMeasureApi::MeasurementConfig, 158
- TriggerConfigNone
 - BMeasureApi, 28
- triggerDelay
 - BMeasureApi::MeasurementConfig, 158
- triggerLevel
 - BMeasureApi::MeasurementConfig, 158
- TriggerMode
 - BMeasureApi, 28
- triggerMode
 - BMeasureApi::MeasurementConfig, 158
- TriggerModeNegative
 - BMeasureApi, 28
- TriggerModeOff
 - BMeasureApi, 28
- TriggerModePositive
 - BMeasureApi, 28
- type
 - BFirmwareInfo, 52
 - BMeasureApi::ChannelConfig, 109
 - BMeasureApi::ConfigItem, 120
 - BMeasureApi::DataBlock, 130
 - BMeasureApi::DataBlockProc, 133
 - BMeasureApi::Version, 163
- unit
 - BMeasureApi::BMeasureUnits, 94
- unitAdd
 - BMeasureApi::BMeasureUnits, 95
- unitDelete
 - BMeasureApi::BMeasureUnits, 95
- unitMaster
 - BMeasureApi::BMeasureUnits, 95
- unitsConnect
 - BMeasureApi::BMeasureUnits, 95
- unitsConnected
 - BMeasureApi::BMeasureUnits, 95
- unitsConnectedNum
 - BMeasureApi::BMeasureUnits, 95
- unitsDisconnect
 - BMeasureApi::BMeasureUnits, 95
- unitSetEnabled
 - BMeasureApi::BMeasureUnits, 96
- unitSetOrder
 - BMeasureApi::BMeasureUnits, 96
- unitsFind
 - BMeasureApi::BMeasureUnits, 96
- unitsNum
 - BMeasureApi::BMeasureUnits, 96
- unitSort
 - BMeasureApi, 45
- upload
 - Dfu, 142
- upload_cmd
 - Dfu, 142
- validateFile
 - Dfu, 142
- validateFormat
 - BMeasureApi::DataFile, 135
- value
 - BMeasureApi::CalibrateInfo, 106

- BMeasureApi::ConfigItem, [120](#)
- variant
 - BMeasureApi::NodeInfo, [160](#)
- ver0
 - BFirmwareInfo, [52](#)
 - BMeasureApi::Version, [163](#)
- ver1
 - BFirmwareInfo, [52](#)
 - BMeasureApi::Version, [163](#)
- ver2
 - BFirmwareInfo, [52](#)
 - BMeasureApi::Version, [163](#)
- version
 - BMeasureApi::Configuration, [127](#)
 - BMeasureApi::InfoBlock, [150](#)
- wait
 - BMeasureApi::CommsNet, [112](#)
 - BMeasureApi::CommsSerial, [115](#)
 - BMeasureApi::CommsUsb, [117](#)
- wifiAddress
 - BMeasureApi::Information, [154](#)
- wifiAp0
 - BMeasureApi::Configuration, [128](#)
- wifiGateway
 - BMeasureApi::Information, [154](#)
- wifiMacAddress
 - BMeasureApi::Information, [154](#)
- wifiMask
 - BMeasureApi::Information, [154](#)
- WifiMode
 - BMeasureApi, [28](#)
- wifiMode
 - BMeasureApi::Configuration, [128](#)
 - BMeasureApi::Information, [154](#)
- WifiModeAp
 - BMeasureApi, [28](#)
- WifiModeClient
 - BMeasureApi, [28](#)
- WifiModeOff
 - BMeasureApi, [28](#)
- wifiStatus
 - BMeasureApi::NodeStatus, [162](#)
- wifiVersion
 - BMeasureApi::BoardConfig, [104](#)
 - BMeasureApi::NodeInfo, [160](#)
- write
 - BMeasureApi::CommsNet, [112](#)
 - BMeasureApi::CommsSerial, [115](#)
 - BMeasureApi::CommsUsb, [118](#)
- writeAvailable
 - BMeasureApi::CommsNet, [112](#)
- writeChunks
 - BMeasureApi::CommsNet, [112](#)
- writeData
 - BMeasureApi::DataFile, [135](#), [136](#)
- writeEnd
 - BMeasureApi::DataFile, [136](#)
- writeInfo
 - BMeasureApi::DataFile, [136](#)
 - writeInfoBMeas
 - BMeasureApi::DataFile, [136](#)
 - writeInfoCsv
 - BMeasureApi::DataFile, [136](#)
 - writeInfoTdms
 - BMeasureApi::DataFile, [136](#)