

BMeasure-lib

0.7.9

Generated by Doxygen 1.8.14



# Contents

- 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 . . . . . 17
      - 6.1.1.1 ChannelConfigs . . . . . 17
    - 6.1.2 Enumeration Type Documentation . . . . . 17
      - 6.1.2.1 AwgOutput . . . . . 17
      - 6.1.2.2 BlockTypes . . . . . 17
      - 6.1.2.3 CalibrateStage . . . . . 18

---

6.1.2.4	ChannelType	18
6.1.2.5	DataBlockType	18
6.1.2.6	DataSend	19
6.1.2.7	DigitalMode	19
6.1.2.8	ErrorNum	19
6.1.2.9	FilesysDeleteType	19
6.1.2.10	FileType	20
6.1.2.11	LogDataMode	20
6.1.2.12	MeasureMode	20
6.1.2.13	MessageSource	20
6.1.2.14	Mode	21
6.1.2.15	NetworkMode	21
6.1.2.16	NodeType	21
6.1.2.17	SampleType	22
6.1.2.18	SecureMode	22
6.1.2.19	Status	22
6.1.2.20	SyncMode	23
6.1.2.21	TdsDataType	23
6.1.2.22	TriggerConfig	23
6.1.2.23	TriggerMode	24
6.1.2.24	Waveform	24
6.1.3	Function Documentation	24
6.1.3.1	channelTypeString()	24
6.1.3.2	round512()	25
6.1.3.3	sampleTypeString()	25
6.1.3.4	TocBigEndian()	25
6.1.3.5	TocDaqRawData()	25
6.1.3.6	TocInterleavedData()	25
6.1.3.7	TocMetaData()	25
6.1.3.8	TocNewObjList()	25
6.1.3.9	TocRawData()	26
6.1.3.10	toFloat()	26
6.1.3.11	unitSort()	26
6.1.4	Variable Documentation	26
6.1.4.1	apiVersion	26

---

<b>7 Class Documentation</b>	<b>27</b>
7.1 BMeasureApi::AwgConfig Class Reference	27
7.1.1 Member Function Documentation	27
7.1.1.1 getMembers()	27
7.1.2 Member Data Documentation	28
7.1.2.1 amplitude	28
7.1.2.2 duty	28
7.1.2.3 frequency	28
7.1.2.4 offset	28
7.1.2.5 output	28
7.1.2.6 spare	28
7.1.2.7 waveform	29
7.2 BFirmwareInfo Struct Reference	29
7.2.1 Member Data Documentation	29
7.2.1.1 checksum	29
7.2.1.2 length	29
7.2.1.3 magic	29
7.2.1.4 type	30
7.2.1.5 ver0	30
7.2.1.6 ver1	30
7.2.1.7 ver2	30
7.3 BMdns Class Reference	30
7.3.1 Constructor & Destructor Documentation	30
7.3.1.1 BMdns()	31
7.3.1.2 ~BMdns()	31
7.3.2 Member Function Documentation	31
7.3.2.1 findServices()	31
7.3.2.2 init()	31
7.3.3 Member Data Documentation	31
7.3.3.1 osocket	31

---

7.3.3.2	otransactionId	31
7.4	BMdnsService Class Reference	32
7.4.1	Member Data Documentation	32
7.4.1.1	address	32
7.4.1.2	extra	32
7.4.1.3	hostname	32
7.4.1.4	name	32
7.5	BMeasureApi::BMeasure Class Reference	33
7.5.1	Constructor & Destructor Documentation	35
7.5.1.1	BMeasure()	36
7.5.2	Member Function Documentation	36
7.5.2.1	calibrate()	36
7.5.2.2	calibrateServe()	36
7.5.2.3	factoryReset()	36
7.5.2.4	factoryResetServe()	36
7.5.2.5	fileClose()	36
7.5.2.6	fileCloseServe()	37
7.5.2.7	fileDelete()	37
7.5.2.8	fileDeleteServe()	37
7.5.2.9	fileList()	37
7.5.2.10	fileListServe()	37
7.5.2.11	fileOpen()	37
7.5.2.12	fileOpenServe()	38
7.5.2.13	fileRead()	38
7.5.2.14	fileReadServe()	38
7.5.2.15	fileSysDelete()	38
7.5.2.16	fileSysDeleteServe()	38
7.5.2.17	fileSysInfo()	38
7.5.2.18	fileSysInfoServe()	39
7.5.2.19	fileWrite()	39

---

---

7.5.2.20	fileWriteServe()	39
7.5.2.21	functionUnLock()	39
7.5.2.22	functionUnLockServe()	39
7.5.2.23	getAwgConfig()	39
7.5.2.24	getAwgConfigServe()	40
7.5.2.25	getBoardConfig()	40
7.5.2.26	getBoardConfigServe()	40
7.5.2.27	getChannelConfig()	40
7.5.2.28	getChannelConfigServe()	40
7.5.2.29	getConfig()	40
7.5.2.30	getConfigServe()	41
7.5.2.31	getDigital()	41
7.5.2.32	getDigitalServe()	41
7.5.2.33	getInfoBlock()	41
7.5.2.34	getInfoBlockServe()	41
7.5.2.35	getInformation()	41
7.5.2.36	getInformationServe()	42
7.5.2.37	getMeasurement()	42
7.5.2.38	getMeasurementConfig()	42
7.5.2.39	getMeasurementConfigServe()	42
7.5.2.40	getMeasurementServe()	42
7.5.2.41	getNodeInfo()	42
7.5.2.42	getNodeInfoServe()	43
7.5.2.43	getStatus()	43
7.5.2.44	getStatusServe()	43
7.5.2.45	getSwitch()	43
7.5.2.46	getSwitchServe()	43
7.5.2.47	login()	43
7.5.2.48	loginServe()	44
7.5.2.49	measure()	44

---

---

7.5.2.50	measureServe()	44
7.5.2.51	processRequest()	44
7.5.2.52	runBoardTest()	44
7.5.2.53	runBoardTestServe()	44
7.5.2.54	sendData()	45
7.5.2.55	sendDataEnable()	45
7.5.2.56	sendDataEnableServe()	45
7.5.2.57	sendDataServe()	45
7.5.2.58	sendInfo()	45
7.5.2.59	sendInfoServe()	45
7.5.2.60	sendMessage()	46
7.5.2.61	sendMessageServe()	46
7.5.2.62	sendStatus()	46
7.5.2.63	sendStatusServe()	46
7.5.2.64	sendTime()	46
7.5.2.65	sendTimeServe()	46
7.5.2.66	setAnalogueOut()	47
7.5.2.67	setAnalogueOutServe()	47
7.5.2.68	setAwgConfig()	47
7.5.2.69	setAwgConfigServe()	47
7.5.2.70	setAwgWaveform()	47
7.5.2.71	setAwgWaveformServe()	47
7.5.2.72	setBoardConfig()	48
7.5.2.73	setBoardConfigServe()	48
7.5.2.74	setChannelConfig()	48
7.5.2.75	setChannelConfigFull()	48
7.5.2.76	setChannelConfigFullServe()	48
7.5.2.77	setChannelConfigServe()	48
7.5.2.78	setConfig()	49
7.5.2.79	setConfigServe()	49



---

7.5.2.80	setDigital()	49
7.5.2.81	setDigitalServe()	49
7.5.2.82	setMeasurement()	49
7.5.2.83	setMeasurementConfig()	49
7.5.2.84	setMeasurementConfigServe()	50
7.5.2.85	setMeasurementServe()	50
7.5.2.86	setMode()	50
7.5.2.87	setModeServe()	50
7.5.2.88	setRelay()	50
7.5.2.89	setRelayServe()	50
7.5.2.90	setSecureMode()	51
7.5.2.91	setSecureModeServe()	51
7.6	BMeasureApi::BMeasureUnit Class Reference	51
7.6.1	Constructor & Destructor Documentation	52
7.6.1.1	BMeasureUnit()	52
7.6.1.2	~BMeasureUnit()	52
7.6.2	Member Function Documentation	53
7.6.2.1	connect()	53
7.6.2.2	device()	53
7.6.2.3	disconnect()	53
7.6.2.4	disconnected()	53
7.6.2.5	findDevices()	53
7.6.2.6	findDevicesNetwork()	53
7.6.2.7	findDevicesUsb()	54
7.6.2.8	info()	54
7.6.2.9	numChannels()	54
7.6.2.10	processdataBlock()	54
7.6.2.11	run()	54
7.6.2.12	sendDataServe()	54
7.6.2.13	sendDataServe1()	55

7.6.2.14	serialNumber()	55
7.6.2.15	setChannelConfig()	55
7.6.2.16	setMeasurement()	55
7.6.3	Member Data Documentation	55
7.6.3.1	blockNumChannels	55
7.6.3.2	blockNumSamples	55
7.6.3.3	oblockCount	56
7.6.3.4	ochannels	56
7.6.3.5	oconfigMeasurement	56
7.6.3.6	odataBlock	56
7.6.3.7	odevice	56
7.6.3.8	odisconnecting	56
7.6.3.9	oinfo	56
7.6.3.10	onodeInfo	57
7.6.3.11	osampleCount	57
7.6.3.12	osequenceNext	57
7.7	BMeasureApi::BMeasureUnit1 Class Reference	57
7.7.1	Constructor & Destructor Documentation	58
7.7.1.1	BMeasureUnit1()	58
7.7.2	Member Function Documentation	58
7.7.2.1	disconnected()	58
7.7.2.2	sendDataServe1()	58
7.7.2.3	sendMessageServe()	59
7.7.2.4	serialNumber()	59
7.7.2.5	setSerialNumber()	59
7.7.3	Member Data Documentation	59
7.7.3.1	oconnected	59
7.7.3.2	oenabled	59
7.7.3.3	omeasureUnits	59
7.7.3.4	oorder	60

---

7.7.3.5	oserialNumber	60
7.7.3.6	osource	60
7.8	BMeasureApi::BMeasureUnitDevice Class Reference	60
7.8.1	Constructor & Destructor Documentation	60
7.8.1.1	BMeasureUnitDevice()	60
7.8.2	Member Data Documentation	61
7.8.2.1	device	61
7.8.2.2	serialNumber	61
7.9	BMeasureApi::BMeasureUnits Class Reference	61
7.9.1	Constructor & Destructor Documentation	63
7.9.1.1	BMeasureUnits()	63
7.9.1.2	~BMeasureUnits()	63
7.9.2	Member Function Documentation	64
7.9.2.1	clear()	64
7.9.2.2	dataAvailable()	64
7.9.2.3	dataClear()	64
7.9.2.4	dataDone()	64
7.9.2.5	dataEvent()	64
7.9.2.6	dataProcessEnable()	64
7.9.2.7	dataRead()	65
7.9.2.8	dataSetNumStreams()	65
7.9.2.9	dataWait()	65
7.9.2.10	debugPrint()	65
7.9.2.11	disconnected()	65
7.9.2.12	getAwgConfig()	65
7.9.2.13	getChannelConfig()	66
7.9.2.14	getConfig()	66
7.9.2.15	getFreeBlock()	66
7.9.2.16	getInfoBlock()	66
7.9.2.17	getInformation()	66

---

7.9.2.18	<a href="#">getMeasurement()</a>	66
7.9.2.19	<a href="#">getMeasurementConfig()</a>	67
7.9.2.20	<a href="#">getStatus()</a>	67
7.9.2.21	<a href="#">numChannels()</a>	67
7.9.2.22	<a href="#">outputBlock()</a>	67
7.9.2.23	<a href="#">run()</a>	67
7.9.2.24	<a href="#">sendDataEnable()</a>	67
7.9.2.25	<a href="#">sendDataProcess()</a>	68
7.9.2.26	<a href="#">sendDataProcessTrigger()</a>	68
7.9.2.27	<a href="#">sendDataQueue()</a>	68
7.9.2.28	<a href="#">sendDataServe1()</a>	68
7.9.2.29	<a href="#">sendMessage()</a>	68
7.9.2.30	<a href="#">sendMessageServe()</a>	68
7.9.2.31	<a href="#">sendTime()</a>	68
7.9.2.32	<a href="#">setAwgConfig()</a>	69
7.9.2.33	<a href="#">setChannelConfig()</a>	69
7.9.2.34	<a href="#">setConfig()</a>	69
7.9.2.35	<a href="#">setMeasurement()</a>	69
7.9.2.36	<a href="#">setMeasurementConfig()</a>	69
7.9.2.37	<a href="#">setMode()</a>	69
7.9.2.38	<a href="#">unit()</a>	70
7.9.2.39	<a href="#">unitAdd()</a>	70
7.9.2.40	<a href="#">unitDelete()</a>	70
7.9.2.41	<a href="#">unitMaster()</a>	70
7.9.2.42	<a href="#">unitsConnect()</a>	70
7.9.2.43	<a href="#">unitsConnected()</a>	70
7.9.2.44	<a href="#">unitsConnectedNum()</a>	70
7.9.2.45	<a href="#">unitsDisconnect()</a>	71
7.9.2.46	<a href="#">unitSetEnabled()</a>	71
7.9.2.47	<a href="#">unitSetOrder()</a>	71

---

---

7.9.2.48	unitsFind()	71
7.9.2.49	unitsNum()	71
7.9.3	Member Data Documentation	71
7.9.3.1	odataBlocksFree	71
7.9.3.2	odataBlocksIn	72
7.9.3.3	odataBlocksOut	72
7.9.3.4	odataBlocksOutCount	72
7.9.3.5	odataBlocksProcess	72
7.9.3.6	odataBlocksProcessNum	72
7.9.3.7	odataStreamNum	72
7.9.3.8	ofill	72
7.9.3.9	olocalTrigger	72
7.9.3.10	olockInput	73
7.9.3.11	olockOutput	73
7.9.3.12	olockUnits	73
7.9.3.13	onumBlocks	73
7.9.3.14	onumChannels	73
7.9.3.15	onumConnected	73
7.9.3.16	oprocEnable	73
7.9.3.17	oprocRunning	74
7.9.3.18	ostartSample	74
7.9.3.19	otriggered	74
7.9.3.20	ounitMaster	74
7.9.3.21	ounits	74
7.10	BMeasureApi::BMeasureUnitsDataBlock Class Reference	74
7.10.1	Constructor & Destructor Documentation	75
7.10.1.1	BMeasureUnitsDataBlock()	75
7.10.1.2	~BMeasureUnitsDataBlock()	75
7.10.2	Member Function Documentation	75
7.10.2.1	init()	75

---

7.10.3	Member Data Documentation	75
7.10.3.1	odataBlock	75
7.10.3.2	ofill	76
7.10.3.3	oinUse	76
7.11	BMeasureApi::BoardConfig Class Reference	76
7.11.1	Member Function Documentation	76
7.11.1.1	getMembers()	76
7.11.2	Member Data Documentation	77
7.11.2.1	buildTime	77
7.11.2.2	calibAdcOffsets	77
7.11.2.3	calibAdcScales	77
7.11.2.4	calibAttenScales	77
7.11.2.5	calibDacOffsets	77
7.11.2.6	calibDacScales	77
7.11.2.7	calibTemp	77
7.11.2.8	calibTime	78
7.11.2.9	hardwareVersion	78
7.11.2.10	macAddress	78
7.11.2.11	magic	78
7.11.2.12	serialNumber	78
7.11.2.13	spare0	78
7.11.2.14	testMode	78
7.12	BMeasureApi::CalibrateInfo Class Reference	79
7.12.1	Member Function Documentation	79
7.12.1.1	getMembers()	79
7.12.2	Member Data Documentation	79
7.12.2.1	calibrateFrequency	79
7.12.2.2	calibrateTime	79
7.12.2.3	stage	80
7.12.2.4	value	80

---

7.13 BMeasureApi::ChannelConfig Class Reference . . . . .	80
7.13.1 Member Function Documentation . . . . .	81
7.13.1.1 getMembers() . . . . .	81
7.13.2 Member Data Documentation . . . . .	81
7.13.2.1 attenuator . . . . .	81
7.13.2.2 calibOffset . . . . .	81
7.13.2.3 calibScale . . . . .	81
7.13.2.4 calibScaleAtten1 . . . . .	82
7.13.2.5 dataChannel . . . . .	82
7.13.2.6 enabled . . . . .	82
7.13.2.7 id . . . . .	82
7.13.2.8 name . . . . .	82
7.13.2.9 number . . . . .	82
7.13.2.10 offset . . . . .	83
7.13.2.11 pgaGain . . . . .	83
7.13.2.12 process . . . . .	83
7.13.2.13 sampleType . . . . .	83
7.13.2.14 scale . . . . .	83
7.13.2.15 siUnits . . . . .	83
7.13.2.16 spare0 . . . . .	83
7.13.2.17 type . . . . .	84
7.14 BMeasureApi::CommsNet Class Reference . . . . .	84
7.14.1 Constructor & Destructor Documentation . . . . .	84
7.14.1.1 CommsNet() . . . . .	85
7.14.1.2 ~CommsNet() . . . . .	85
7.14.2 Member Function Documentation . . . . .	85
7.14.2.1 connect() . . . . .	85
7.14.2.2 disconnect() . . . . .	85
7.14.2.3 init() . . . . .	85
7.14.2.4 read() . . . . .	85

---

7.14.2.5	<a href="#">readAvailable()</a>	86
7.14.2.6	<a href="#">wait()</a>	86
7.14.2.7	<a href="#">write()</a>	86
7.14.2.8	<a href="#">writeAvailable()</a>	86
7.14.2.9	<a href="#">writeChunks()</a>	86
7.14.3	<a href="#">Member Data Documentation</a>	86
7.14.3.1	<a href="#">osocket</a>	87
7.15	<a href="#">BMeasureApi::CommsSerial Class Reference</a>	87
7.15.1	<a href="#">Constructor &amp; Destructor Documentation</a>	87
7.15.1.1	<a href="#">CommsSerial()</a>	87
7.15.1.2	<a href="#">~CommsSerial()</a>	88
7.15.2	<a href="#">Member Function Documentation</a>	88
7.15.2.1	<a href="#">connect()</a>	88
7.15.2.2	<a href="#">disconnect()</a>	88
7.15.2.3	<a href="#">read()</a>	88
7.15.2.4	<a href="#">readAvailable()</a>	88
7.15.2.5	<a href="#">wait()</a>	88
7.15.2.6	<a href="#">write()</a>	89
7.15.3	<a href="#">Member Data Documentation</a>	89
7.15.3.1	<a href="#">odevice</a>	89
7.15.3.2	<a href="#">oserialPort</a>	89
7.16	<a href="#">BMeasureApi::CommsUsb Class Reference</a>	89
7.16.1	<a href="#">Constructor &amp; Destructor Documentation</a>	90
7.16.1.1	<a href="#">CommsUsb()</a>	90
7.16.1.2	<a href="#">~CommsUsb()</a>	90
7.16.2	<a href="#">Member Function Documentation</a>	90
7.16.2.1	<a href="#">connect()</a>	90
7.16.2.2	<a href="#">disconnect()</a>	90
7.16.2.3	<a href="#">read()</a>	91
7.16.2.4	<a href="#">readAvailable()</a>	91



---

7.16.2.5	readChunk()	91
7.16.2.6	wait()	91
7.16.2.7	write()	91
7.16.3	Member Data Documentation	91
7.16.3.1	obuffer	92
7.16.3.2	ocontext	92
7.16.3.3	odev	92
7.16.3.4	odevice	92
7.16.3.5	onum	92
7.16.3.6	oterminated	92
7.17	BMeasureApi::ConfigItem Class Reference	92
7.17.1	Member Function Documentation	93
7.17.1.1	getMembers()	93
7.17.2	Member Data Documentation	93
7.17.2.1	name	93
7.17.2.2	spare	93
7.17.2.3	type	93
7.17.2.4	value	94
7.18	BMeasureApi::Configuration Class Reference	94
7.18.1	Member Function Documentation	95
7.18.1.1	getMembers()	95
7.18.2	Member Data Documentation	95
7.18.2.1	digitalMode	95
7.18.2.2	ethernetEnable	95
7.18.2.3	location	96
7.18.2.4	logData	96
7.18.2.5	logDataDevice	96
7.18.2.6	logDataMode	96
7.18.2.7	mode	96
7.18.2.8	name	96

7.18.2.9	networkAddress	97
7.18.2.10	networkGateway	97
7.18.2.11	networkMask	97
7.18.2.12	networkMode	97
7.18.2.13	networkTimeServer	97
7.18.2.14	program	97
7.18.2.15	rs485BaudRate	98
7.18.2.16	rs485Bits	98
7.18.2.17	rs485StopBits	98
7.18.2.18	sampleFrequencyMode	98
7.18.2.19	source	98
7.18.2.20	spare1	98
7.18.2.21	spare3	99
7.18.2.22	spare4	99
7.18.2.23	usbaEnable	99
7.18.2.24	usbbEnable	99
7.18.2.25	version	99
7.18.2.26	wifiEnable	99
7.19	BMeasureApi::DataBlock Class Reference	100
7.19.1	Member Function Documentation	100
7.19.1.1	getMembers()	100
7.19.2	Member Data Documentation	100
7.19.2.1	data	100
7.19.2.2	numChannels	101
7.19.2.3	numSamples	101
7.19.2.4	sequence	101
7.19.2.5	source	101
7.19.2.6	spare	101
7.19.2.7	status	101
7.19.2.8	time	102

---

7.19.2.9	type	102
7.20	BMeasureApi::DataFile Class Reference	102
7.20.1	Constructor & Destructor Documentation	103
7.20.1.1	DataFile()	103
7.20.1.2	~DataFile()	103
7.20.2	Member Function Documentation	103
7.20.2.1	close()	103
7.20.2.2	getFileName()	103
7.20.2.3	init()	104
7.20.2.4	open()	104
7.20.2.5	readData()	104
7.20.2.6	readInfo()	104
7.20.2.7	validateFormat()	104
7.20.2.8	writeData()	104
7.20.2.9	writeEnd()	105
7.20.2.10	writeInfo()	105
7.20.2.11	writeInfoBMeas()	105
7.20.2.12	writeInfoTdms()	105
7.20.3	Member Data Documentation	105
7.20.3.1	ofile	105
7.20.3.2	ofilename	105
7.20.3.3	offormat	106
7.20.3.4	omode	106
7.20.3.5	opacket	106
7.20.3.6	opacketLen	106
7.21	Dfu Class Reference	106
7.21.1	Detailed Description	107
7.21.2	Constructor & Destructor Documentation	107
7.21.2.1	Dfu()	107
7.21.2.2	~Dfu()	107

7.21.3	Member Function Documentation	107
7.21.3.1	clearStatus()	107
7.21.3.2	connect()	107
7.21.3.3	detectDevice()	108
7.21.3.4	disconnect()	108
7.21.3.5	getStatus()	108
7.21.3.6	init()	108
7.21.3.7	reset()	108
7.21.3.8	upload()	108
7.21.3.9	upload_cmd()	109
7.21.3.10	validateFile()	109
7.21.4	Member Data Documentation	109
7.21.4.1	oconnected	109
7.21.4.2	ocontext	109
7.21.4.3	odev	109
7.21.4.4	overbose	109
7.22	DfuStatus Struct Reference	110
7.22.1	Member Data Documentation	110
7.22.1.1	iString	110
7.22.1.2	pollTimeout	110
7.22.1.3	state	110
7.22.1.4	status	110
7.23	BMeasureApi::FileData Class Reference	110
7.23.1	Member Function Documentation	111
7.23.1.1	getMembers()	111
7.23.2	Member Data Documentation	111
7.23.2.1	data	111
7.23.2.2	length	111
7.24	BMeasureApi::FileInfo Class Reference	111
7.24.1	Member Function Documentation	112

---

7.24.1.1	getMembers()	112
7.24.2	Member Data Documentation	112
7.24.2.1	fileLength	112
7.24.2.2	fileType	112
7.24.2.3	name	113
7.24.2.4	spare	113
7.24.2.5	time	113
7.25	BMeasureApi::FilesysInfo Class Reference	113
7.25.1	Member Function Documentation	113
7.25.1.1	getMembers()	114
7.25.2	Member Data Documentation	114
7.25.2.1	free	114
7.25.2.2	name	114
7.25.2.3	size	114
7.26	BMeasureApi::InfoBlock Class Reference	114
7.26.1	Member Function Documentation	115
7.26.1.1	getMembers()	115
7.26.2	Member Data Documentation	115
7.26.2.1	location	115
7.26.2.2	measureConfig	115
7.26.2.3	name	116
7.26.2.4	nodeInfo	116
7.26.2.5	numChannels	116
7.26.2.6	source	116
7.26.2.7	spare0	116
7.26.2.8	time	116
7.26.2.9	version	117
7.27	BMeasureApi::Information Class Reference	117
7.27.1	Member Function Documentation	117
7.27.1.1	getMembers()	118

---

7.27.2	Member Data Documentation	118
7.27.2.1	networkAddress	118
7.27.2.2	networkGateway	118
7.27.2.3	networkMask	118
7.27.2.4	networkMode	118
7.27.2.5	networkTimeServer	118
7.27.2.6	nodeInfo	119
7.27.2.7	numChannels	119
7.27.2.8	numConfigItems	119
7.27.2.9	spare0	119
7.27.2.10	spare1	119
7.27.2.11	time	119
7.28	BMeasureApi::MeasurementConfig Class Reference	120
7.28.1	Member Function Documentation	120
7.28.1.1	getMembers()	120
7.28.2	Member Data Documentation	120
7.28.2.1	description	120
7.28.2.2	measureMode	121
7.28.2.3	measurePeriod	121
7.28.2.4	numSamples0	121
7.28.2.5	numSamples1	121
7.28.2.6	numSamplesBlock	121
7.28.2.7	sampleRate	121
7.28.2.8	triggerChannel	121
7.28.2.9	triggerConfig	122
7.28.2.10	triggerDelay	122
7.28.2.11	triggerLevel	122
7.28.2.12	triggerMode	122
7.29	BMeasureApi::NodeInfo Class Reference	122
7.29.1	Member Function Documentation	123

---

---

7.29.1.1	getMembers()	123
7.29.2	Member Data Documentation	123
7.29.2.1	apiVersion	123
7.29.2.2	fpgaVersion	123
7.29.2.3	hardwareVersion	123
7.29.2.4	serialNumber	123
7.29.2.5	softwareVersion	123
7.30	BMeasureApi::NodeStatus Class Reference	124
7.30.1	Member Function Documentation	124
7.30.1.1	getMembers()	124
7.30.2	Member Data Documentation	124
7.30.2.1	error	124
7.30.2.2	errorStr	124
7.30.2.3	mode	125
7.30.2.4	spare	125
7.30.2.5	status	125
7.30.2.6	time	125
7.31	BMeasureApi::Version Class Reference	125
7.31.1	Member Function Documentation	125
7.31.1.1	getMembers()	126
7.31.2	Member Data Documentation	126
7.31.2.1	type	126
7.31.2.2	ver0	126
7.31.2.3	ver1	126
7.31.2.4	ver2	126

<b>8 File Documentation</b>	<b>127</b>
8.1 BMdns.cpp File Reference	127
8.1.1 Macro Definition Documentation	127
8.1.1.1 BDEBUGL1	128
8.1.2 Enumeration Type Documentation	128
8.1.2.1 MdnsClass	128
8.1.2.2 MdnsEntryType	128
8.1.2.3 MdnsRecordType	128
8.1.3 Function Documentation	129
8.1.3.1 mdns_read_string()	129
8.1.3.2 mdns_read_strings()	129
8.1.3.3 mdns_write_string()	129
8.2 BMdns.h File Reference	129
8.3 BMeasureB.cpp File Reference	129
8.4 BMeasureB.h File Reference	130
8.5 BMeasureD.cpp File Reference	130
8.5.1 Macro Definition Documentation	130
8.5.1.1 boffsetof	131
8.6 BMeasureD.h File Reference	131
8.7 BMeasureLib.cpp File Reference	133
8.7.1 Macro Definition Documentation	133
8.7.1.1 BDEBUGL1	133
8.7.1.2 BDEBUGL2	133
8.8 BMeasureLib.h File Reference	133
8.9 BMeasureS.cpp File Reference	134
8.10 BMeasureUnit.cpp File Reference	134
8.10.1 Macro Definition Documentation	135
8.10.1.1 BDEBUGL1	135
8.10.1.2 BDEBUGL2	135
8.10.1.3 CONVERT_FLOAT	135



8.11 BMeasureUnit.h File Reference . . . . .	135
8.12 BMeasureUnits.cpp File Reference . . . . .	136
8.12.1 Macro Definition Documentation . . . . .	136
8.12.1.1 BDEBUGL1 . . . . .	136
8.12.1.2 BDEBUGL2 . . . . .	136
8.12.1.3 BDEBUGL3 . . . . .	136
8.13 BMeasureUnits.h File Reference . . . . .	137
8.14 CommsNet.cpp File Reference . . . . .	137
8.14.1 Macro Definition Documentation . . . . .	137
8.14.1.1 BDEBUGL1 . . . . .	137
8.14.1.2 BDEBUGL2 . . . . .	138
8.14.1.3 BDEBUGL3 . . . . .	138
8.15 CommsNet.h File Reference . . . . .	138
8.16 CommsSerial.cpp File Reference . . . . .	138
8.17 CommsSerial.h File Reference . . . . .	138
8.18 CommsUsb.cpp File Reference . . . . .	139
8.18.1 Macro Definition Documentation . . . . .	139
8.18.1.1 BDEBUGL1 . . . . .	139
8.18.1.2 BDEBUGL2 . . . . .	139
8.19 CommsUsb.h File Reference . . . . .	139
8.20 DataFile.cpp File Reference . . . . .	140
8.20.1 Macro Definition Documentation . . . . .	140
8.20.1.1 BDEBUGL1 . . . . .	141
8.20.1.2 BDEBUGL2 . . . . .	141
8.21 DataFile.h File Reference . . . . .	141
8.22 Dfu.cpp File Reference . . . . .	141
8.22.1 Macro Definition Documentation . . . . .	143
8.22.1.1 BDEBUGL1 . . . . .	143
8.22.1.2 BDEBUGL2 . . . . .	143
8.22.1.3 DFU_ABORT . . . . .	143

---

8.22.1.4	DFU_CLRSTATUS	143
8.22.1.5	DFU_DETACH	143
8.22.1.6	DFU_DNLOAD	144
8.22.1.7	DFU_GETSTATE	144
8.22.1.8	DFU_GETSTATUS	144
8.22.1.9	DFU_IFF_ALT	144
8.22.1.10	DFU_IFF_CONFIG	144
8.22.1.11	DFU_IFF_DEVNUM	144
8.22.1.12	DFU_IFF_DFU	144
8.22.1.13	DFU_IFF_IFACE	144
8.22.1.14	DFU_IFF_PATH	145
8.22.1.15	DFU_IFF_PRODUCT	145
8.22.1.16	DFU_IFF_VENDOR	145
8.22.1.17	DFU_STATUS_ERROR_ADDRESS	145
8.22.1.18	DFU_STATUS_ERROR_CHECK_ERASED	145
8.22.1.19	DFU_STATUS_ERROR_ERASE	145
8.22.1.20	DFU_STATUS_ERROR_FILE	145
8.22.1.21	DFU_STATUS_ERROR_FIRMWARE	145
8.22.1.22	DFU_STATUS_ERROR_NOTDONE	146
8.22.1.23	DFU_STATUS_ERROR_POR	146
8.22.1.24	DFU_STATUS_ERROR_PROG	146
8.22.1.25	DFU_STATUS_ERROR_STALLEDPKT	146
8.22.1.26	DFU_STATUS_ERROR_TARGET	146
8.22.1.27	DFU_STATUS_ERROR_UNKNOWN	146
8.22.1.28	DFU_STATUS_ERROR_USBR	146
8.22.1.29	DFU_STATUS_ERROR_VENDOR	146
8.22.1.30	DFU_STATUS_ERROR_VERIFY	147
8.22.1.31	DFU_STATUS_ERROR_WRITE	147
8.22.1.32	DFU_STATUS_OK	147
8.22.1.33	DFU_UPLOAD	147

---

8.22.1.34 STATE_APP_DETACH . . . . .	147
8.22.1.35 STATE_APP_IDLE . . . . .	147
8.22.1.36 STATE_DFU_DOWNLOAD_BUSY . . . . .	147
8.22.1.37 STATE_DFU_DOWNLOAD_IDLE . . . . .	147
8.22.1.38 STATE_DFU_DOWNLOAD_SYNC . . . . .	148
8.22.1.39 STATE_DFU_ERROR . . . . .	148
8.22.1.40 STATE_DFU_IDLE . . . . .	148
8.22.1.41 STATE_DFU_MANIFEST . . . . .	148
8.22.1.42 STATE_DFU_MANIFEST_SYNC . . . . .	148
8.22.1.43 STATE_DFU_MANIFEST_WAIT_RESET . . . . .	148
8.22.1.44 STATE_DFU_UPLOAD_IDLE . . . . .	148
8.22.2 Enumeration Type Documentation . . . . .	148
8.22.2.1 dfuse_command . . . . .	148
8.22.3 Function Documentation . . . . .	149
8.22.3.1 pageAddress() . . . . .	149
8.22.3.2 pageNumber() . . . . .	149
8.22.4 Variable Documentation . . . . .	149
8.22.4.1 BFirmwareInfoEncrypt1 . . . . .	149
8.22.4.2 BFirmwareInfoMagic . . . . .	149
8.23 Dfu.h File Reference . . . . .	149
8.24 overview.dox File Reference . . . . .	149
<b>Index</b>	<b>151</b>



# Chapter 1

## Main Page

### Author

Dr Terry Barnaby

### Version

0.2.8

### Date

2019-10-14

## 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-api.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;
    BInformation    info;
    BConfiguration  config;
    BMeasurementConfig mc;
    BDataBlock      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.setMeasurement(mc))
        return err;

    printf("Run single measurement\n");
    if(err = bmeasure.measure(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.setMeasurement(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 = 0;
        else:
            state = 1;

        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::AwgConfig . . . . .	27
BComms [external]	
BMeasureApi::CommsNet . . . . .	84
BMeasureApi::CommsSerial . . . . .	87
BMeasureApi::CommsUsb . . . . .	89
BFirmwareInfo . . . . .	29
BMdns . . . . .	30
BMdnsService . . . . .	32
BMeasureApi::BMeasureUnitDevice . . . . .	60
BMeasureApi::BMeasureUnitsDataBlock . . . . .	74
BoapMc1Comms [external]	
BMeasureApi::BMeasure . . . . .	33
BMeasureApi::BMeasureUnit . . . . .	51
BMeasureApi::BMeasureUnit1 . . . . .	57
BMeasureApi::BoardConfig . . . . .	76
BTask [external]	
BMeasureApi::BMeasureUnit . . . . .	51
BMeasureApi::BMeasureUnits . . . . .	61
BMeasureApi::CalibrateInfo . . . . .	79
BMeasureApi::ChannelConfig . . . . .	80
BMeasureApi::ConfigItem . . . . .	92
BMeasureApi::Configuration . . . . .	94
BMeasureApi::DataBlock . . . . .	100
BMeasureApi::DataFile . . . . .	102
Dfu . . . . .	106
DfuStatus . . . . .	110
BMeasureApi::FileData . . . . .	110
BMeasureApi::FileInfo . . . . .	111
BMeasureApi::FilesysInfo . . . . .	113
BMeasureApi::InfoBlock . . . . .	114
BMeasureApi::Information . . . . .	117
BMeasureApi::MeasurementConfig . . . . .	120
BMeasureApi::NodeInfo . . . . .	122
BMeasureApi::NodeStatus . . . . .	124
BMeasureApi::Version . . . . .	125



# Chapter 4

## Class Index

### 4.1 Class List

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

BMeasureApi::AwgConfig	27
BFirmwareInfo	29
BMdns	30
BMdnsService	32
BMeasureApi::BMeasure	33
BMeasureApi::BMeasureUnit	51
BMeasureApi::BMeasureUnit1	57
BMeasureApi::BMeasureUnitDevice	60
BMeasureApi::BMeasureUnits	61
BMeasureApi::BMeasureUnitsDataBlock	74
BMeasureApi::BoardConfig	76
BMeasureApi::CalibrateInfo	79
BMeasureApi::ChannelConfig	80
BMeasureApi::CommsNet	84
BMeasureApi::CommsSerial	87
BMeasureApi::CommsUsb	89
BMeasureApi::ConfigItem	92
BMeasureApi::Configuration	94
BMeasureApi::DataBlock	100
BMeasureApi::DataFile	102
Dfu	
The Dfu access class	106
DfuStatus	110
BMeasureApi::FileData	110
BMeasureApi::FileInfo	111
BMeasureApi::FilesysInfo	113
BMeasureApi::InfoBlock	114
BMeasureApi::Information	117
BMeasureApi::MeasurementConfig	120
BMeasureApi::NodeInfo	122
BMeasureApi::NodeStatus	124
BMeasureApi::Version	125





# Chapter 5

## File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

BMdns.cpp	127
BMdns.h	129
BMeasureB.cpp	129
BMeasureB.h	130
BMeasureD.cpp	130
BMeasureD.h	131
BMeasureLib.cpp	133
BMeasureLib.h	133
BMeasureS.cpp	134
BMeasureUnit.cpp	134
BMeasureUnit.h	135
BMeasureUnits.cpp	136
BMeasureUnits.h	137
CommsNet.cpp	137
CommsNet.h	138
CommsSerial.cpp	138
CommsSerial.h	138
CommsUsb.cpp	139
CommsUsb.h	139
DataFile.cpp	140
DataFile.h	141
Dfu.cpp	141
Dfu.h	149



# Chapter 6

## Namespace Documentation

### 6.1 BMeasureApi Namespace Reference

#### Classes

- 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 [DataFile](#)
- 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` { `ErrorSystem` = 64, `ErrorDataOverrun` = 65 }
- enum `NodeType` { `NodeTypeNone` = 0, `NodeTypeBMeasure1` = 1 }
- enum `SecureMode` { `SecureModeOpen`, `SecureMoteRemote`, `SecureModeFull` }
- enum `Status` {  
`StatusNone` = 0x00, `StatusError` = 0x01, `StatusWarning` = 0x02, `StatusRun` = 0x04,  
`StatusTriggerWait` = 0x08, `StatusEnd0` = 0x10, `StatusEnd1` = 0x20, `StatusDataOverrun` = 0x40,  
`StatusFpgaOverrun` = 0x80 }
- enum `Mode` {  
`Modeldle` = 0, `ModeRun` = 1, `ModeRunProgram` = 2, `ModelInternal` = 3,  
`ModeSleep` = 4, `ModeDemo1` = 5 }
- enum `BlockTypes` { `BlockTypeInfo` = 0x424E4531, `BlockTypeData` = 0x424E4532 }
- enum `ChannelType` {  
`ChannelTypeNone` = 0, `ChannelTypeAnalogueIn` = 1, `ChannelTypeAnalogueOut` = 0x81, `ChannelTypeDigitalIn`  
= 2,  
`ChannelTypeDigitalOut` = 0x82 }
- 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 `TriggerMode` { `TriggerModeOff` = 0, `TriggerModePositive` = 1, `TriggerModeNegative` = 2 }
- enum `TriggerConfig` { `TriggerConfigNone` = 0 }
- enum `DigitalMode` {  
`DigitalModeInput` = 0, `DigitalModeOutput` = 1, `DigitalInOut` = 2, `DigitalModeSyncMaster` = 3,  
`DigitalModeSyncSlave` = 4 }
- enum `Waveform` {  
`WaveformNone`, `WaveformDc`, `WaveformSine`, `WaveformSquare`,  
`WaveformTriangle`, `WaveformNoise`, `WaveformArbitrary` }
- enum `AwgOutput` { `AwgOutputNone`, `AwgOutputAO0`, `AwgOutputAO1`, `AwgOutputAO01` }
- enum `FileType` { `FileTypeNone`, `FileTypeFile`, `FileTypeDir` }
- enum `FilesysDeleteType` { `FilesysDeleteTypeNone`, `FilesysDeleteTypeData`, `FilesysDeleteTypeFormat` }
- enum `LogDataMode` { `LogDataModeNormal`, `LogDataModeDeleteOld` }
- enum `DataBlockType` { `DataBlockTypeFloat32`, `DataBlockType125i` }
- enum `DataSend` { `DataSendOff`, `DataSendOn` }
- enum `CalibrateStage` {  
`CalibrateStageNone` = 0, `CalibrateStageClear` = 1, `CalibrateStageSettle` = 2, `CalibrateStageAdcOffsets` = 3,  
`CalibrateStageDacOffsets` = 4, `CalibrateStageDacScaling0` = 5, `CalibrateStageDacScaling1` = 6,  
`CalibrateStageAdcScaling` = 7 }
- enum `MessageSource` {  
`MessageSourceGeneral` = 0, `MessageSourceDebug` = 1, `MessageSourceTest` = 2, `MessageSourceWifi` = 3,  
`MessageSourceWifiTest` = 4 }
- enum `NetworkMode` { `NetworkModeOff` = 0, `NetworkModeDhcp` = 1, `NetworkModeManual` = 2 }
- enum `TdsDataType` {  
`TdsTypeVoid`, `TdsTypeI8`, `TdsTypeI16`, `TdsTypeI32`,  
`TdsTypeI64`, `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

- const char \* [channelTypeString](#) ([ChannelType](#) type)
- const char \* [sampleTypeString](#) ([SampleType](#) type)
- [BFloat32 toFloat](#) ( [BUInt32](#) v)
- static int [unitSort](#) ([BMeasureUnit1](#) \*&u1, [BMeasureUnit1](#) \*&u2)
- 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 AwgOutput

```
enum BMeasureApi::AwgOutput
```

##### Enumerator

<a href="#">AwgOutputNone</a>	
<a href="#">AwgOutputAO0</a>	
<a href="#">AwgOutputAO1</a>	
<a href="#">AwgOutputAO01</a>	

#### 6.1.2.2 BlockTypes

```
enum BMeasureApi::BlockTypes
```

**Enumerator**

BlockTypeInfo	
BlockTypeData	

**6.1.2.3 CalibrateStage**

enum `BMeasureApi::CalibrateStage`

**Enumerator**

CalibrateStageNone	
CalibrateStageClear	
CalibrateStageSettle	
CalibrateStageAdcOffsets	
CalibrateStageDacOffsets	
CalibrateStageDacScaling0	
CalibrateStageDacScaling1	
CalibrateStageAdcScaling	

**6.1.2.4 ChannelType**

enum `BMeasureApi::ChannelType`

**Enumerator**

ChannelTypeNone	
ChannelTypeAnalogueIn	
ChannelTypeAnalogueOut	
ChannelTypeDigitalIn	
ChannelTypeDigitalOut	

**6.1.2.5 DataBlockType**

enum `BMeasureApi::DataBlockType`

**Enumerator**

DataBlockTypeFloat32	
DataBlockType125i	

### 6.1.2.6 DataSend

enum `BMeasureApi::DataSend`

#### Enumerator

DataSendOff	
DataSendOn	

### 6.1.2.7 DigitalMode

enum `BMeasureApi::DigitalMode`

#### Enumerator

DigitalModeInput	
DigitalModeOutput	
DigitalInOut	
DigitalModeSyncMaster	
DigitalModeSyncSlave	

### 6.1.2.8 ErrorNum

enum `BMeasureApi::ErrorNum`

#### Enumerator

ErrorSystem	
ErrorDataOverrun	

### 6.1.2.9 FilesysDeleteType

enum `BMeasureApi::FilesysDeleteType`

#### Enumerator

FilesysDeleteTypeNone	
FilesysDeleteTypeData	
FilesysDeleteTypeFormat	

### 6.1.2.10 FileType

enum `BMeasureApi::FileType`

#### Enumerator

FileTypeNone	
FileTypeFile	
FileTypeDir	

### 6.1.2.11 LogDataMode

enum `BMeasureApi::LogDataMode`

#### Enumerator

LogDataModeNormal	
LogDataModeDeleteOld	

### 6.1.2.12 MeasureMode

enum `BMeasureApi::MeasureMode`

#### Enumerator

MeasureModeOff	
MeasureModeOneShot	
MeasureModeRepeat	
MeasureModeContinuous	

### 6.1.2.13 MessageSource

enum `BMeasureApi::MessageSource`

#### Enumerator

MessageSourceGeneral	
MessageSourceDebug	



## Enumerator

MessageSourceTest	
MessageSourceWifi	
MessageSourceWifiTest	

## 6.1.2.14 Mode

```
enum BMeasureApi::Mode
```

## Enumerator

ModeIdle	
ModeRun	
ModeRunProgram	
ModeInternal	
ModeSleep	
ModeDemo1	

## 6.1.2.15 NetworkMode

```
enum BMeasureApi::NetworkMode
```

## Enumerator

NetworkModeOff	
NetworkModeDhcp	
NetworkModeManual	

## 6.1.2.16 NodeType

```
enum BMeasureApi::NodeType
```

## Enumerator

NodeTypeNone	
NodeTypeBMeasure1	

### 6.1.2.17 SampleType

enum `BMeasureApi::SampleType`

#### Enumerator

SampleTypeNone	
SampleTypeBool	
SampleTypeInt8	
SampleTypeInt16	
SampleTypeInt32	
SampleTypeFloat32	
SampleTypeFloat64	

### 6.1.2.18 SecureMode

enum `BMeasureApi::SecureMode`

#### Enumerator

SecureModeOpen	
SecureMoteRemote	
SecureModeFull	

### 6.1.2.19 Status

enum `BMeasureApi::Status`

#### Enumerator

StatusNone	
StatusError	
StatusWarning	
StatusRun	
StatusTriggerWait	
StatusEnd0	
StatusEnd1	
StatusDataOverrun	
StatusFpgaOverrun	

## 6.1.2.20 SyncMode

```
enum BMeasureApi::SyncMode
```

## Enumerator

SyncModeOff	
SyncModeMaster	
SyncModeSlave	

## 6.1.2.21 TdsDataType

```
enum BMeasureApi::TdsDataType
```

## Enumerator

TdsTypeVoid	
TdsTypeI8	
TdsTypeI16	
TdsTypeI32	
TdsTypeI64	
TdsTypeU8	
TdsTypeU16	
TdsTypeU32	
TdsTypeU64	
TdsTypeSingleFloat	
TdsTypeDoubleFloat	
TdsTypeExtendedFloat	
TdsTypeSingleFloatWithUnit	
TdsTypeDoubleFloatWithUnit	
TdsTypeExtendedFloatWithUnit	
TdsTypeString	
TdsTypeBoolean	
TdsTypeTimeStamp	
TdsTypeFixedPoint	
TdsTypeComplexSingleFloat	
TdsTypeComplexDoubleFloat	
TdsTypeDAQmxRawData	

## 6.1.2.22 TriggerConfig

```
enum BMeasureApi::TriggerConfig
```

**Enumerator**

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

**6.1.2.23 TriggerMode**

```
enum BMeasureApi::TriggerMode
```

**Enumerator**

TriggerModeOff	
TriggerModePositive	
TriggerModeNegative	

**6.1.2.24 Waveform**

```
enum BMeasureApi::Waveform
```

**Enumerator**

WaveformNone	
WaveformDc	
WaveformSine	
WaveformSquare	
WaveformTriangle	
WaveformNoise	
WaveformArbitrary	

**6.1.3 Function Documentation****6.1.3.1 channelTypeString()**

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

### 6.1.3.2 round512()

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

### 6.1.3.3 sampleTypeString()

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

### 6.1.3.4 TocBigEndian()

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

### 6.1.3.5 TocDaqRawData()

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

### 6.1.3.6 TocInterleavedData()

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

### 6.1.3.7 TocMetaData()

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

### 6.1.3.8 TocNewObjList()

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

#### 6.1.3.9 TocRawData()

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

#### 6.1.3.10 toFloat()

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

#### 6.1.3.11 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::AwgConfig Class Reference

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

#### Static Public Member Functions

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

#### Public Attributes

- [Waveform waveform](#)  
*The waveform.*
- [AwgOutput output](#)  
*The output channels.*
- **BUint8 spare** [2]
- **BFloat32 frequency**  
*The frequency.*
- **BFloat32 amplitude**  
*The peak amplitude in Volts.*
- **BFloat32 offset**  
*The DC offset in volts.*
- **BFloat32 duty**  
*The Duty cycle in %.*

#### 7.1.1 Member Function Documentation

##### 7.1.1.1 getMembers()

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

## 7.1.2 Member Data Documentation

### 7.1.2.1 amplitude

**BFloat32** BMeasureApi::AwgConfig::amplitude

The peak amplitude in Volts.

### 7.1.2.2 duty

**BFloat32** BMeasureApi::AwgConfig::duty

The Duty cycle in %.

### 7.1.2.3 frequency

**BFloat32** BMeasureApi::AwgConfig::frequency

The frequency.

### 7.1.2.4 offset

**BFloat32** BMeasureApi::AwgConfig::offset

The DC offset in volts.

### 7.1.2.5 output

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

The output channels.

### 7.1.2.6 spare

**BUInt8** BMeasureApi::AwgConfig::spare[2]



### 7.1.2.7 waveform

`Waveform` BMeasureApi::AwgConfig::waveform

The waveform.

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

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

## 7.2 BFirmwareInfo Struct Reference

### Public Attributes

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

### 7.2.1 Member Data Documentation

#### 7.2.1.1 checksum

**BUInt32** BFirmwareInfo::checksum

#### 7.2.1.2 length

**BUInt32** BFirmwareInfo::length

#### 7.2.1.3 magic

**BUInt32** BFirmwareInfo::magic

#### 7.2.1.4 type

**BUInt8** BFirmwareInfo::type

#### 7.2.1.5 ver0

**BUInt8** BFirmwareInfo::ver0

#### 7.2.1.6 ver1

**BUInt8** BFirmwareInfo::ver1

#### 7.2.1.7 ver2

**BUInt8** BFirmwareInfo::ver2

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

- [Dfu.cpp](#)

## 7.3 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](#)

#### 7.3.1 Constructor & Destructor Documentation

### 7.3.1.1 BMdns()

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

### 7.3.1.2 ~BMdns()

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

## 7.3.2 Member Function Documentation

### 7.3.2.1 findServices()

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

Unicast response, class IN

### 7.3.2.2 init()

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

## 7.3.3 Member Data Documentation

### 7.3.3.1 osocket

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

### 7.3.3.2 otransactionId

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

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

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

## 7.4 BMdnsService Class Reference

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

### Public Attributes

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

### 7.4.1 Member Data Documentation

#### 7.4.1.1 address

```
BSocketAddressINET BMdnsService::address
```

#### 7.4.1.2 extra

```
BStringList BMdnsService::extra
```

#### 7.4.1.3 hostname

```
BString BMdnsService::hostname
```

#### 7.4.1.4 name

```
BString BMdnsService::name
```

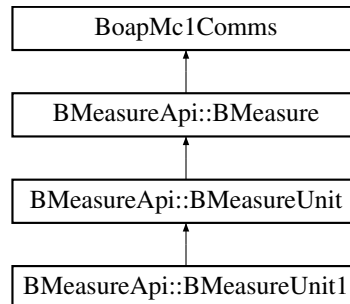
The documentation for this class was generated from the following file:

- [BMdns.h](#)

## 7.5 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.*
- void [factoryReset](#) (const **BInt32** &bootLoader, const **BInt32** &resetConfig)  
*Factory reset.*
- **BError** [getStatus](#) ([NodeStatus](#) &nodeStatus)  
*Get the node status.*
- void [sendStatus](#) (const [NodeStatus](#) &nodeStatus)  
*Sends the current status.*
- void [sendTime](#) (const **BTimeUs** &time)  
*Sends the current time.*
- **BError** [setSecureMode](#) (const **BUInt64** &key, const [SecureMode](#) &secureMode)  
*Set the security mode.*
- **BError** [login](#) (const **BUInt64** &key, const **BString** &user, const **BString** &password)  
*Provides user/password information for secure connection.*
- **BError** [setMode](#) (const [Mode](#) &mode)  
*Set the current operational mode.*
- **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)  
*Should we have this, not generic for different instruments ?*
- **BError** [setConfig](#) (const [Configuration](#) &config)  
*Should we have this, not generic for different instruments ?*
- **BError** [getMeasurementConfig](#) ([MeasurementConfig](#) &measurementConfig)  
*Get measurement config.*
- **BError** [setMeasurementConfig](#) (const [MeasurementConfig](#) &measurementConfig)  
*Set measurement config.*
- **BError** [getMeasurement](#) ([MeasurementConfig](#) &measurementConfig)

- Get measurement config.*

  - **BError** [setMeasurement](#) (const [MeasurementConfig](#) &measurementConfig)
- Set measurement config.*

  - **BError** [sendDataEnable](#) (const [DataSend](#) &sendType)
- Enable the sending of data.*

  - void [sendInfo](#) (const [InfoBlock](#) &infoBlock)
- Sends an info block.*

  - void [sendData](#) (const [DataBlock](#) &dataBlock)
- Sends a data block.*

  - **BError** [measure](#) ([DataBlock](#) &dataBlock)
- Performs a single measurement.*

  - **BError** [getAwgConfig](#) ([AwgConfig](#) &awgConfig)
- Get AWG Configuration.*

  - **BError** [setAwgConfig](#) (const [AwgConfig](#) &awgConfig)
- Configure AWG.*

  - **BError** [setAwgWaveform](#) (const [DataBlock](#) &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 [BInt32](#) &state)
- Set relay.*

  - **BError** [getSwitch](#) (const [BUInt32](#) &switchNum, [BInt32](#) &state)
- Get digital bits.*

  - **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](#) ( [BTimeout](#) timeoutUs= [BTimeoutForever](#))
- virtual **BError** [getNodeInfoServe](#) ([NodeInfo](#) &nodeInfo)
- virtual void [factoryResetServe](#) (const [BInt32](#) &bootLoader, const [BInt32](#) &resetConfig)
- virtual **BError** [getStatusServe](#) ([NodeStatus](#) &nodeStatus)
- virtual void [sendStatusServe](#) (const [NodeStatus](#) &nodeStatus)

- virtual void [sendTimeServe](#) (const **BTimeUs** &time)
- virtual **BError** [setSecureModeServe](#) (const **BUInt64** &key, const [SecureMode](#) &secureMode)
- virtual **BError** [loginServe](#) (const **BUInt64** &key, const **BString** &user, const **BString** &password)
- virtual **BError** [setModeServe](#) (const [Mode](#) &mode)
- 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](#) &channel←  
Config)
- 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](#) ([MeasurementConfig](#) &measurementConfig)
- virtual **BError** [setMeasurementConfigServe](#) (const [MeasurementConfig](#) &measurementConfig)
- virtual **BError** [getMeasurementServe](#) ([MeasurementConfig](#) &measurementConfig)
- virtual **BError** [setMeasurementServe](#) (const [MeasurementConfig](#) &measurementConfig)
- virtual **BError** [sendDataEnableServe](#) (const [DataSend](#) &sendType)
- virtual void [sendInfoServe](#) (const [InfoBlock](#) &infoBlock)
- virtual void [sendDataServe](#) (const [DataBlock](#) &dataBlock)
- virtual **BError** [measureServe](#) ([DataBlock](#) &dataBlock)
- virtual **BError** [getAwgConfigServe](#) ([AwgConfig](#) &awgConfig)
- virtual **BError** [setAwgConfigServe](#) (const [AwgConfig](#) &awgConfig)
- virtual **BError** [setAwgWaveformServe](#) (const [DataBlock](#) &dataBlock)
- 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 **BInt32** &state)
- virtual **BError** [getSwitchServe](#) (const **BUInt32** &switchNum, **BInt32** &state)
- 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

### 7.5.1 Constructor & Destructor Documentation

### 7.5.1.1 BMeasure()

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

## 7.5.2 Member Function Documentation

### 7.5.2.1 calibrate()

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

Calibrate system.

### 7.5.2.2 calibrateServe()

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

### 7.5.2.3 factoryReset()

```
void BMeasureApi::BMeasure::factoryReset (
    const BInt32 & bootLoader,
    const BInt32 & resetConfig )
```

Factory reset.

### 7.5.2.4 factoryResetServe()

```
void BMeasureApi::BMeasure::factoryResetServe (
    const BInt32 & bootLoader,
    const BInt32 & resetConfig ) [virtual]
```

### 7.5.2.5 fileClose()

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



### 7.5.2.6 fileCloseServe()

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

### 7.5.2.7 fileDelete()

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

### 7.5.2.8 fileDeleteServe()

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

### 7.5.2.9 fileList()

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

### 7.5.2.10 fileListServe()

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

### 7.5.2.11 fileOpen()

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

### 7.5.2.12 fileOpenServe()

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

### 7.5.2.13 fileRead()

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

### 7.5.2.14 fileReadServe()

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

### 7.5.2.15 fileSysDelete()

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

### 7.5.2.16 fileSysDeleteServe()

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

### 7.5.2.17 fileSysInfo()

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

#### 7.5.2.18 fileSysInfoServe()

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

#### 7.5.2.19 fileWrite()

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

#### 7.5.2.20 fileWriteServe()

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

#### 7.5.2.21 functionUnlock()

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

Unlock/Lock special functions.

#### 7.5.2.22 functionUnlockServe()

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

#### 7.5.2.23 getAwgConfig()

```
BError BMeasureApi::BMeasure::getAwgConfig (
    AwgConfig & awgConfig )
```

Get AWG [Configuration](#).

#### 7.5.2.24 getAwgConfigServe()

```
BError BMeasureApi::BMeasure::getAwgConfigServe (
    AwgConfig & awgConfig ) [virtual]
```

#### 7.5.2.25 getBoardConfig()

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

Get the boards configuration.

#### 7.5.2.26 getBoardConfigServe()

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

#### 7.5.2.27 getChannelConfig()

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

#### 7.5.2.28 getChannelConfigServe()

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

#### 7.5.2.29 getConfig()

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

Should we have this, not generic for different instruments ?

### 7.5.2.30 getConfigServe()

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

### 7.5.2.31 getDigital()

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

Get digital bits.

### 7.5.2.32 getDigitalServe()

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

### 7.5.2.33 getInfoBlock()

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

### 7.5.2.34 getInfoBlockServe()

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

### 7.5.2.35 getInformation()

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

#### 7.5.2.36 `getInformationServe()`

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

#### 7.5.2.37 `getMeasurement()`

```
BError BMeasureApi::BMeasure::getMeasurement (
    MeasurementConfig & measurementConfig )
```

Get measurement config.

#### 7.5.2.38 `getMeasurementConfig()`

```
BError BMeasureApi::BMeasure::getMeasurementConfig (
    MeasurementConfig & measurementConfig )
```

Get measurement config.

#### 7.5.2.39 `getMeasurementConfigServe()`

```
BError BMeasureApi::BMeasure::getMeasurementConfigServe (
    MeasurementConfig & measurementConfig ) [virtual]
```

#### 7.5.2.40 `getMeasurementServe()`

```
BError BMeasureApi::BMeasure::getMeasurementServe (
    MeasurementConfig & measurementConfig ) [virtual]
```

#### 7.5.2.41 `getNodeInfo()`

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

Get node information.

#### 7.5.2.42 getNodeInfoServe()

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

#### 7.5.2.43 getStatus()

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

Get the node status.

#### 7.5.2.44 getStatusServe()

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

#### 7.5.2.45 getSwitch()

```
BError BMeasureApi::BMeasure::getSwitch (
    const BUInt32 & switchNum,
    BInt32 & state )
```

Get digital bits.

#### 7.5.2.46 getSwitchServe()

```
BError BMeasureApi::BMeasure::getSwitchServe (
    const BUInt32 & switchNum,
    BInt32 & state ) [virtual]
```

#### 7.5.2.47 login()

```
BError BMeasureApi::BMeasure::login (
    const BUInt64 & key,
    const BString & user,
    const BString & password )
```

Provides user/password information for secure connection.

#### 7.5.2.48 loginServe()

```
BError BMeasureApi::BMeasure::loginServe (
    const BUInt64 & key,
    const BString & user,
    const BString & password ) [virtual]
```

#### 7.5.2.49 measure()

```
BError BMeasureApi::BMeasure::measure (
    DataBlock & dataBlock )
```

Performs a single measurement.

#### 7.5.2.50 measureServe()

```
BError BMeasureApi::BMeasure::measureServe (
    DataBlock & dataBlock ) [virtual]
```

#### 7.5.2.51 processRequest()

```
BError BMeasureApi::BMeasure::processRequest (
    BTimeout timeoutUs = BTimeoutForever ) [virtual]
```

Reimplemented from **BoapMc1Comms**.

#### 7.5.2.52 runBoardTest()

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

Runs the given board test.

#### 7.5.2.53 runBoardTestServe()

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



#### 7.5.2.54 sendData()

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

Sends a data block.

#### 7.5.2.55 sendDataEnable()

```
BError BMeasureApi::BMeasure::sendDataEnable (
    const DataSend & sendType )
```

Enable the sending of data.

#### 7.5.2.56 sendDataEnableServe()

```
BError BMeasureApi::BMeasure::sendDataEnableServe (
    const DataSend & sendType ) [virtual]
```

#### 7.5.2.57 sendDataServe()

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

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

#### 7.5.2.58 sendInfo()

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

Sends an info block.

#### 7.5.2.59 sendInfoServe()

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

### 7.5.2.60 sendMessage()

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

Sends text messages.

### 7.5.2.61 sendMessageServe()

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

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

### 7.5.2.62 sendStatus()

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

Sends the current status.

### 7.5.2.63 sendStatusServe()

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

### 7.5.2.64 sendTime()

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

Sends the current time.

### 7.5.2.65 sendTimeServe()

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

#### 7.5.2.66 setAnalogueOut()

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

Set analogue output value.

#### 7.5.2.67 setAnalogueOutServe()

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

#### 7.5.2.68 setAwgConfig()

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

Configure AWG.

#### 7.5.2.69 setAwgConfigServe()

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

#### 7.5.2.70 setAwgWaveform()

```
BError BMeasureApi::BMeasure::setAwgWaveform (
    const DataBlock & dataBlock )
```

Configure AWG Arbitrary waveform.

#### 7.5.2.71 setAwgWaveformServe()

```
BError BMeasureApi::BMeasure::setAwgWaveformServe (
    const DataBlock & dataBlock ) [virtual]
```

#### 7.5.2.72 setBoardConfig()

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

Sets the boards configuration, requires key.

#### 7.5.2.73 setBoardConfigServe()

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

#### 7.5.2.74 setChannelConfig()

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

#### 7.5.2.75 setChannelConfigFull()

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

#### 7.5.2.76 setChannelConfigFullServe()

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

#### 7.5.2.77 setChannelConfigServe()

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

#### 7.5.2.78 setConfig()

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

Should we have this, not generic for different instruments ?

#### 7.5.2.79 setConfigServe()

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

#### 7.5.2.80 setDigital()

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

Set digital bits.

#### 7.5.2.81 setDigitalServe()

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

#### 7.5.2.82 setMeasurement()

```
BError BMeasureApi::BMeasure::setMeasurement (
    const MeasurementConfig & measurementConfig )
```

Set measurement config.

#### 7.5.2.83 setMeasurementConfig()

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

Set measurement config.

#### 7.5.2.84 setMeasurementConfigServe()

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

#### 7.5.2.85 setMeasurementServe()

```
BError BMeasureApi::BMeasure::setMeasurementServe (
    const MeasurementConfig & measurementConfig ) [virtual]
```

#### 7.5.2.86 setMode()

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

Set the current operational mode.

#### 7.5.2.87 setModeServe()

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

#### 7.5.2.88 setRelay()

```
BError BMeasureApi::BMeasure::setRelay (
    const BUInt32 & relayNum,
    const BInt32 & state )
```

Set relay.

#### 7.5.2.89 setRelayServe()

```
BError BMeasureApi::BMeasure::setRelayServe (
    const BUInt32 & relayNum,
    const BInt32 & state ) [virtual]
```

## 7.5.2.90 setSecureMode()

```
BError BMeasureApi::BMeasure::setSecureMode (
    const BUInt64 & key,
    const SecureMode & secureMode )
```

Set the security mode.

## 7.5.2.91 setSecureModeServe()

```
BError BMeasureApi::BMeasure::setSecureModeServe (
    const BUInt64 & key,
    const SecureMode & secureMode ) [virtual]
```

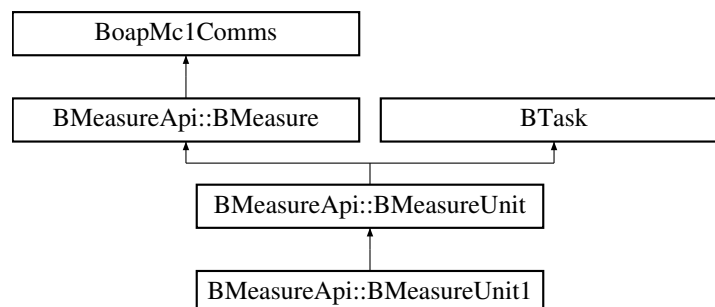
The documentation for this class was generated from the following files:

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

## 7.6 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** **setMeasurement** (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

### 7.6.1 Constructor & Destructor Documentation

#### 7.6.1.1 BMeasureUnit()

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

#### 7.6.1.2 ~BMeasureUnit()

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



## 7.6.2 Member Function Documentation

### 7.6.2.1 connect()

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

Connect to a device.

### 7.6.2.2 device()

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

### 7.6.2.3 disconnect()

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

### 7.6.2.4 disconnected()

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

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

### 7.6.2.5 findDevices()

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

Find available devices.

### 7.6.2.6 findDevicesNetwork()

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

Find available devices on Network.

### 7.6.2.7 findDevicesUsb()

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

Find available devices on USB bus.

### 7.6.2.8 info()

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

### 7.6.2.9 numChannels()

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

The number of channels of data.

### 7.6.2.10 processdataBlock()

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

### 7.6.2.11 run()

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

Threaded run mode.

Reimplemented from **BTask**.

### 7.6.2.12 sendDataServe()

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

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

### 7.6.2.13 sendDataServe1()

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

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

### 7.6.2.14 serialNumber()

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

### 7.6.2.15 setChannelConfig()

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

### 7.6.2.16 setMeasurement()

```
BError BMeasureApi::BMeasureUnit::setMeasurement (
    const MeasurementConfig & configMeasurement ) [virtual]
```

## 7.6.3 Member Data Documentation

### 7.6.3.1 blockNumChannels

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

### 7.6.3.2 blockNumSamples

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

### 7.6.3.3 oblockCount

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

### 7.6.3.4 ochannels

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

### 7.6.3.5 oconfigMeasurement

MeasurementConfig BMeasureApi::BMeasureUnit::oconfigMeasurement [protected]

### 7.6.3.6 odataBlock

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

### 7.6.3.7 odevice

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

### 7.6.3.8 odisconnecting

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

### 7.6.3.9 oinfo

Information BMeasureApi::BMeasureUnit::oinfo [protected]

Instrument info.

## 7.6.3.10 onodeInfo

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

## 7.6.3.11 osampleCount

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

## 7.6.3.12 osequenceNext

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

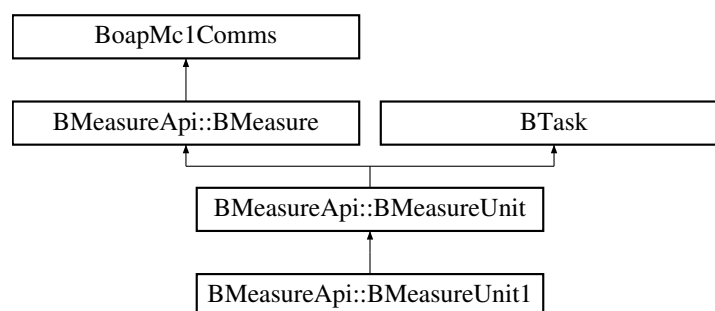
The documentation for this class was generated from the following files:

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

## 7.7 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

### 7.7.1 Constructor & Destructor Documentation

#### 7.7.1.1 BMeasureUnit1()

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

### 7.7.2 Member Function Documentation

#### 7.7.2.1 disconnected()

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

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

#### 7.7.2.2 sendDataServe1()

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

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

### 7.7.2.3 sendMessageServe()

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

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

### 7.7.2.4 serialNumber()

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

### 7.7.2.5 setSerialNumber()

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

## 7.7.3 Member Data Documentation

### 7.7.3.1 oconnected

```
Bool BMeasureApi::BMeasureUnit1::oconnected
```

### 7.7.3.2 oenabled

```
Bool BMeasureApi::BMeasureUnit1::oenabled
```

### 7.7.3.3 omeasureUnits

```
BMeasureUnits& BMeasureApi::BMeasureUnit1::omeasureUnits
```

#### 7.7.3.4 oorder

```
BUInt BMeasureApi::BMeasureUnit1::oorder
```

#### 7.7.3.5 oserialNumber

```
BString BMeasureApi::BMeasureUnit1::oserialNumber
```

#### 7.7.3.6 osource

```
BUInt BMeasureApi::BMeasureUnit1::osource
```

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

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

## 7.8 BMeasureApi::BMeasureUnitDevice Class Reference

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

### Public Member Functions

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

### Public Attributes

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

### 7.8.1 Constructor & Destructor Documentation

#### 7.8.1.1 BMeasureUnitDevice()

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



## 7.8.2 Member Data Documentation

### 7.8.2.1 device

**BString** BMeasureApi::BMeasureUnitDevice::device

### 7.8.2.2 serialNumber

**BString** BMeasureApi::BMeasureUnitDevice::serialNumber

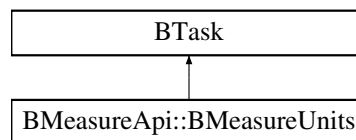
The documentation for this class was generated from the following file:

- [BMeasureUnit.h](#)

## 7.9 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 `dataProcessEnable` ( **Bool** on)
  - Enable the processing 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)
- void `run` ()
  - Threaded run mode.*
- void `sendDataQueue` (const `DataBlock` &dataBlock)
- void `sendDataProcess` ()
- void `sendDataProcessTrigger` ()
- void `outputBlock` (`BMeasureUnitsDataBlock` \*block)
- 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` (`MeasurementConfig` &measurement)
  - Get measurement config.*
- virtual **BError** `setMeasurementConfig` (const `MeasurementConfig` &measurement)
  - Set measurement config.*
- virtual **BError** `getMeasurement` (`MeasurementConfig` &measurement)
  - Get measurement settings.*
- virtual **BError** `setMeasurement` (const `MeasurementConfig` &measurement)
  - Set measurement settings.*
- virtual **BError** `sendDataEnable` (const `DataSend` &dataSend)
  - Enables the sending of data.*
- virtual **BError** `getAwgConfig` (`AwgConfig` &awgConfig)
  - Get AWG Configuration.*
- virtual **BError** `setAwgConfig` (const `AwgConfig` &awgConfig)
  - Configure AWG.*
- 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]
- [MeasurementConfig](#) [olocalTrigger](#)
- [Bool](#) [otriggered](#)
- [BUInt](#) [ostartSample](#)

## Additional Inherited Members

### 7.9.1 Constructor & Destructor Documentation

#### 7.9.1.1 BMeasureUnits()

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

#### 7.9.1.2 ~BMeasureUnits()

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

## 7.9.2 Member Function Documentation

### 7.9.2.1 clear()

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

### 7.9.2.2 dataAvailable()

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

### 7.9.2.3 dataClear()

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

### 7.9.2.4 dataDone()

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

### 7.9.2.5 dataEvent()

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

### 7.9.2.6 dataProcessEnable()

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

Enable the processing of data.

### 7.9.2.7 dataRead()

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

### 7.9.2.8 dataSetNumStreams()

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

Set the number of data output channels.

### 7.9.2.9 dataWait()

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

### 7.9.2.10 debugPrint()

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

### 7.9.2.11 disconnected()

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

### 7.9.2.12 getAwgConfig()

```
BError BMeasureApi::BMeasureUnits::getAwgConfig (
    AwgConfig & awgConfig ) [virtual]
```

Get AWG [Configuration](#).

### 7.9.2.13 getChannelConfig()

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

### 7.9.2.14 getConfig()

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

Should we have this, not generic for different instruments ?

### 7.9.2.15 getFreeBlock()

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

### 7.9.2.16 getInfoBlock()

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

### 7.9.2.17 getInformation()

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

### 7.9.2.18 getMeasurement()

```
BError BMeasureApi::BMeasureUnits::getMeasurement (
    MeasurementConfig & measurement ) [virtual]
```

Get measurement settings.

### 7.9.2.19 getMeasurementConfig()

```
BError BMeasureApi::BMeasureUnits::getMeasurementConfig (
    MeasurementConfig & measurement ) [virtual]
```

Get measurement config.

### 7.9.2.20 getStatus()

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

### 7.9.2.21 numChannels()

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

The number of channels of data.

### 7.9.2.22 outputBlock()

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

### 7.9.2.23 run()

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

Threaded run mode.

Reimplemented from **BTask**.

### 7.9.2.24 sendDataEnable()

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

Enables the sending of data.

#### 7.9.2.25 sendDataProcess()

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

#### 7.9.2.26 sendDataProcessTrigger()

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

#### 7.9.2.27 sendDataQueue()

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

#### 7.9.2.28 sendDataServe1()

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

#### 7.9.2.29 sendMessage()

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

#### 7.9.2.30 sendMessageServe()

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

#### 7.9.2.31 sendTime()

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

Sends the current time.



### 7.9.2.32 setAwgConfig()

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

Configure AWG.

### 7.9.2.33 setChannelConfig()

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

### 7.9.2.34 setConfig()

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

Should we have this, not generic for different instruments ?

### 7.9.2.35 setMeasurement()

```
BError BMeasureApi::BMeasureUnits::setMeasurement (
    const MeasurementConfig & measurement ) [virtual]
```

Set measurement settings.

### 7.9.2.36 setMeasurementConfig()

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

Set measurement config.

### 7.9.2.37 setMode()

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

Set the current operational mode.

### 7.9.2.38 unit()

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

### 7.9.2.39 unitAdd()

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

### 7.9.2.40 unitDelete()

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

### 7.9.2.41 unitMaster()

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

### 7.9.2.42 unitsConnect()

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

### 7.9.2.43 unitsConnected()

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

### 7.9.2.44 unitsConnectedNum()

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

#### 7.9.2.45 unitsDisconnect()

**BError** BMeasureApi::BMeasureUnits::unitsDisconnect ( )

#### 7.9.2.46 unitSetEnabled()

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

#### 7.9.2.47 unitSetOrder()

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

#### 7.9.2.48 unitsFind()

**BError** BMeasureApi::BMeasureUnits::unitsFind ( )

#### 7.9.2.49 unitsNum()

**BUInt** BMeasureApi::BMeasureUnits::unitsNum ( )

### 7.9.3 Member Data Documentation

#### 7.9.3.1 odataBlocksFree

**BList**<[BMeasureUnitsDataBlock\\*](#)> BMeasureApi::BMeasureUnits::odataBlocksFree [private]

### 7.9.3.2 odataBlocksIn

**BList**<[BMeasureUnitsDataBlock\\*](#)> BMeasureApi::BMeasureUnits::odataBlocksIn [private]

### 7.9.3.3 odataBlocksOut

**BList**<[BMeasureUnitsDataBlock\\*](#)> BMeasureApi::BMeasureUnits::odataBlocksOut[2] [private]

### 7.9.3.4 odataBlocksOutCount

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

### 7.9.3.5 odataBlocksProcess

**BList**<[BMeasureUnitsDataBlock\\*](#)> BMeasureApi::BMeasureUnits::odataBlocksProcess [private]

### 7.9.3.6 odataBlocksProcessNum

**BCondInt** BMeasureApi::BMeasureUnits::odataBlocksProcessNum [private]

### 7.9.3.7 odataStreamNum

**BUInt** BMeasureApi::BMeasureUnits::odataStreamNum [private]

### 7.9.3.8 ofill

**BUInt32** BMeasureApi::BMeasureUnits::ofill [private]

### 7.9.3.9 olocalTrigger

[MeasurementConfig](#) BMeasureApi::BMeasureUnits::olocalTrigger [private]

#### 7.9.3.10 olockInput

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

#### 7.9.3.11 olockOutput

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

#### 7.9.3.12 olockUnits

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

#### 7.9.3.13 onumBlocks

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

#### 7.9.3.14 onumChannels

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

#### 7.9.3.15 onumConnected

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

#### 7.9.3.16 oprocEnable

**BSemaphoreBool** BMeasureApi::BMeasureUnits::oprocEnable [private]

Enable processing.

### 7.9.3.17 oprocRunning

**BSemaphoreBool** BMeasureApi::BMeasureUnits::oprocRunning [private]

Processing is running.

### 7.9.3.18 ostartSample

**BUInt** BMeasureApi::BMeasureUnits::ostartSample [private]

### 7.9.3.19 otriggered

**Bool** BMeasureApi::BMeasureUnits::otriggered [private]

### 7.9.3.20 ounitMaster

**BInt** BMeasureApi::BMeasureUnits::ounitMaster [private]

### 7.9.3.21 ounits

**BList**<BMeasureUnit1\*> BMeasureApi::BMeasureUnits::ounits [private]

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

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

## 7.10 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`

## 7.10.1 Constructor & Destructor Documentation

### 7.10.1.1 BMeasureUnitsDataBlock()

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

### 7.10.1.2 ~BMeasureUnitsDataBlock()

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

## 7.10.2 Member Function Documentation

### 7.10.2.1 init()

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

## 7.10.3 Member Data Documentation

### 7.10.3.1 odataBlock

```
DataBlock* BMeasureApi::BMeasureUnitsDataBlock::odataBlock
```

### 7.10.3.2 ofill

```
BUInt32 BMeasureApi::BMeasureUnitsDataBlock::ofill
```

### 7.10.3.3 oinUse

```
BUInt BMeasureApi::BMeasureUnitsDataBlock::oinUse
```

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

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

## 7.11 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]

### 7.11.1 Member Function Documentation

#### 7.11.1.1 getMembers()

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



## 7.11.2 Member Data Documentation

### 7.11.2.1 buildTime

**BTime** BMeasureApi::BoardConfig::buildTime

### 7.11.2.2 calibAdcOffsets

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

### 7.11.2.3 calibAdcScales

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

### 7.11.2.4 calibAttenScales

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

### 7.11.2.5 calibDacOffsets

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

### 7.11.2.6 calibDacScales

**BFloat64** BMeasureApi::BoardConfig::calibDacScales[2]

### 7.11.2.7 calibTemp

**BFloat32** BMeasureApi::BoardConfig::calibTemp

#### 7.11.2.8 calibTime

**BTime** BMeasureApi::BoardConfig::calibTime

#### 7.11.2.9 hardwareVersion

**Version** BMeasureApi::BoardConfig::hardwareVersion

#### 7.11.2.10 macAddress

**BUInt8** BMeasureApi::BoardConfig::macAddress[6]

#### 7.11.2.11 magic

**BUInt32** BMeasureApi::BoardConfig::magic

#### 7.11.2.12 serialNumber

**BChar** BMeasureApi::BoardConfig::serialNumber[12]

#### 7.11.2.13 spare0

**BUInt8** BMeasureApi::BoardConfig::spare0

#### 7.11.2.14 testMode

**BUInt8** BMeasureApi::BoardConfig::testMode

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

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

## 7.12 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 frequency of calibrations, Awg set to this.*
- **BFloat64** [calibrateTime](#)  
*Number of seconds to calibrate over (synced to multiple AWG cycles)*
- **BFloat64** [value](#)  
*Target/Set Value.*

### 7.12.1 Member Function Documentation

#### 7.12.1.1 [getMembers\(\)](#)

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

### 7.12.2 Member Data Documentation

#### 7.12.2.1 [calibrateFrequency](#)

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

The frequency of calibrations, Awg set to this.

#### 7.12.2.2 [calibrateTime](#)

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

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

### 7.12.2.3 stage

**BUInt32** BMeasureApi::CalibrateInfo::stage

Stage to run.

### 7.12.2.4 value

**BFloat64** BMeasureApi::CalibrateInfo::value

Target/Set Value.

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

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

## 7.13 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]

## 7.13.1 Member Function Documentation

### 7.13.1.1 getMembers()

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

## 7.13.2 Member Data Documentation

### 7.13.2.1 attenuator

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

Attenuator number in use.

### 7.13.2.2 calibOffset

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

The calibration data offset.

### 7.13.2.3 calibScale

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

The calibration data scale factor to volts.

#### 7.13.2.4 calibScaleAtten1

**BFloat64** BMeasureApi::ChannelConfig::calibScaleAtten1

Attenuator 1 scaling.

#### 7.13.2.5 dataChannel

**BUInt32** BMeasureApi::ChannelConfig::dataChannel

Data channel.

#### 7.13.2.6 enabled

**BUInt8** BMeasureApi::ChannelConfig::enabled

Channel is enabled.

#### 7.13.2.7 id

**BChar** BMeasureApi::ChannelConfig::id[16]

#### 7.13.2.8 name

**BChar** BMeasureApi::ChannelConfig::name[16]

#### 7.13.2.9 number

**BUInt8** BMeasureApi::ChannelConfig::number

The channel number.

#### 7.13.2.10 offset

**BFloat64** BMeasureApi::ChannelConfig::offset

The user data offset.

#### 7.13.2.11 pgaGain

**BFloat64** BMeasureApi::ChannelConfig::pgaGain

The PGA gain.

#### 7.13.2.12 process

**BChar** BMeasureApi::ChannelConfig::process[32]

#### 7.13.2.13 sampleType

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

The sample type.

#### 7.13.2.14 scale

**BFloat64** BMeasureApi::ChannelConfig::scale

The user data scale factor.

#### 7.13.2.15 siUnits

**BChar** BMeasureApi::ChannelConfig::siUnits[8]

#### 7.13.2.16 spare0

**BUInt8** BMeasureApi::ChannelConfig::spare0[3]

### 7.13.2.17 type

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

The channel type.

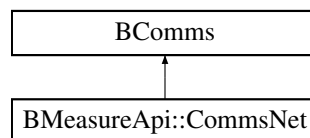
The documentation for this class was generated from the following files:

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

## 7.14 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` (`BEventWaitSet` events, `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`

### Additional Inherited Members

#### 7.14.1 Constructor & Destructor Documentation



### 7.14.1.1 CommsNet()

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

### 7.14.1.2 ~CommsNet()

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

## 7.14.2 Member Function Documentation

### 7.14.2.1 connect()

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

### 7.14.2.2 disconnect()

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

Reimplemented from **BComms**.

### 7.14.2.3 init()

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

Reimplemented from **BComms**.

### 7.14.2.4 read()

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

Implements **BComms**.

#### 7.14.2.5 readAvailable()

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

Reimplemented from **BComms**.

#### 7.14.2.6 wait()

```
BError BMeasureApi::CommsNet::wait (
    BEventWaitSet events,
    BTimeout timeout = -1,
    BUInt32 num = 1 ) [virtual]
```

Reimplemented from **BComms**.

#### 7.14.2.7 write()

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

Implements **BComms**.

#### 7.14.2.8 writeAvailable()

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

Reimplemented from **BComms**.

#### 7.14.2.9 writeChunks()

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

Reimplemented from **BComms**.

### 7.14.3 Member Data Documentation

## 7.14.3.1 osocket

**BSocket** BMeasureApi::CommsNet::osocket [protected]

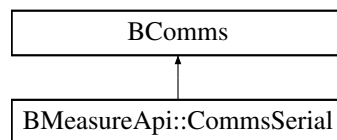
The documentation for this class was generated from the following files:

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

## 7.15 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](#) ( **BEventWaitSet** events, **BTimeout** timeout=-1, **BUInt32** num=1)

## Private Attributes

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

## Additional Inherited Members

## 7.15.1 Constructor &amp; Destructor Documentation

## 7.15.1.1 CommsSerial()

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

### 7.15.1.2 ~CommsSerial()

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

## 7.15.2 Member Function Documentation

### 7.15.2.1 connect()

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

### 7.15.2.2 disconnect()

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

Reimplemented from **BComms**.

### 7.15.2.3 read()

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

Implements **BComms**.

### 7.15.2.4 readAvailable()

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

Reimplemented from **BComms**.

### 7.15.2.5 wait()

```
BError BMeasureApi::CommsSerial::wait (
    BEventWaitSet events,
    BTimeout timeout = -1,
    BUInt32 num = 1 ) [virtual]
```

Reimplemented from **BComms**.

## 7.15.2.6 write()

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

Implements **BComms**.

## 7.15.3 Member Data Documentation

## 7.15.3.1 odevice

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

## 7.15.3.2 oserialPort

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

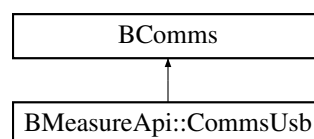
The documentation for this class was generated from the following file:

- [CommsSerial.h](#)

## 7.16 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](#) ( **BEventWaitSet** events, **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`

## Additional Inherited Members

### 7.16.1 Constructor & Destructor Documentation

#### 7.16.1.1 CommsUsb()

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

#### 7.16.1.2 ~CommsUsb()

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

### 7.16.2 Member Function Documentation

#### 7.16.2.1 connect()

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

#### 7.16.2.2 disconnect()

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

Reimplemented from **BComms**.

### 7.16.2.3 read()

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

Implements **BComms**.

### 7.16.2.4 readAvailable()

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

Reimplemented from **BComms**.

### 7.16.2.5 readChunk()

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

### 7.16.2.6 wait()

```
BError BMeasureApi::CommsUsb::wait (
    BEventWaitSet events,
    BTimeout timeout = -1,
    BUInt32 num = 1 ) [virtual]
```

Reimplemented from **BComms**.

### 7.16.2.7 write()

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

Implements **BComms**.

## 7.16.3 Member Data Documentation

### 7.16.3.1 obuffer

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

### 7.16.3.2 ocontext

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

### 7.16.3.3 odev

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

### 7.16.3.4 odevice

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

### 7.16.3.5 onum

```
BUInt BMeasureApi::CommsUsb::onum [private]
```

### 7.16.3.6 oterminated

```
Bool BMeasureApi::CommsUsb::oterminated [private]
```

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

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

## 7.17 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]

## 7.17.1 Member Function Documentation

### 7.17.1.1 getMembers()

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

## 7.17.2 Member Data Documentation

### 7.17.2.1 name

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

### 7.17.2.2 spare

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

### 7.17.2.3 type

```
BUInt8 BMeasureApi::ConfigItem::type
```

The type of data.

## 7.17.2.4 value

**BChar** BMeasureApi::ConfigItem::value[16]

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

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

## 7.18 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.*
- **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.*
- **BUInt8** [spare1](#)
- **DigitalMode** [digitalMode](#)  
*The digital mode.*
- **BUInt8** [ethernetEnable](#)  
*Enable ethernet interface.*
- **BUInt8** [wifiEnable](#)  
*Enable wifi interface.*
- **BUInt8** [usbaEnable](#)  
*Enable USB-A interface.*
- **BUInt8** [usbbEnable](#)  
*Enable USB-B interface.*
- **NetworkMode** [networkMode](#)  
*The network mode (0 - dhcp, 1 - static)*

- **BUInt8** `spare3` [3]
- **BUInt32** `networkAddress`  
*The network IP address.*
- **BUInt32** `networkMask`  
*The network netmask.*
- **BUInt32** `networkGateway`  
*The network gateway.*
- **BUInt32** `networkTimeServer`  
*The network time server.*
- **BUInt32** `rs485BaudRate`  
*The RS485 baud rate.*
- **BUInt8** `rs485Bits`  
*The RS485 number of bits.*
- **BUInt8** `rs485StopBits`  
*The RS485 stop bits.*
- **BUInt8** `spare4` [2]
- **BChar** `program` [32]

## 7.18.1 Member Function Documentation

### 7.18.1.1 `getMembers()`

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

## 7.18.2 Member Data Documentation

### 7.18.2.1 `digitalMode`

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

The digital mode.

### 7.18.2.2 `ethernetEnable`

```
BUInt8 BMeasureApi::Configuration::ethernetEnable
```

Enable ethernet interface.

### 7.18.2.3 location

**BChar** BMeasureApi::Configuration::location[16]

### 7.18.2.4 logData

**BUInt8** BMeasureApi::Configuration::logData

Log the data.

### 7.18.2.5 logDataDevice

**BUInt8** BMeasureApi::Configuration::logDataDevice

The device to store data.

### 7.18.2.6 logDataMode

**BUInt8** BMeasureApi::Configuration::logDataMode

Log data mode.

### 7.18.2.7 mode

**Mode** BMeasureApi::Configuration::mode

The boot run mode.

### 7.18.2.8 name

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

### 7.18.2.9 `networkAddress`

**BUInt32** BMeasureApi::Configuration::networkAddress

The network IP address.

### 7.18.2.10 `networkGateway`

**BUInt32** BMeasureApi::Configuration::networkGateway

The network gateway.

### 7.18.2.11 `networkMask`

**BUInt32** BMeasureApi::Configuration::networkMask

The network netmask.

### 7.18.2.12 `networkMode`

`NetworkMode` BMeasureApi::Configuration::networkMode

The network mode (0 - dhcp, 1 - static)

### 7.18.2.13 `networkTimeServer`

**BUInt32** BMeasureApi::Configuration::networkTimeServer

The network time server.

### 7.18.2.14 `program`

**BChar** BMeasureApi::Configuration::program[32]

#### 7.18.2.15 rs485BaudRate

**BUInt32** BMeasureApi::Configuration::rs485BaudRate

The RS485 baud rate.

#### 7.18.2.16 rs485Bits

**BUInt8** BMeasureApi::Configuration::rs485Bits

The RS485 number of bits.

#### 7.18.2.17 rs485StopBits

**BUInt8** BMeasureApi::Configuration::rs485StopBits

The RS485 stop bits.

#### 7.18.2.18 sampleFrequencyMode

**BUInt8** BMeasureApi::Configuration::sampleFrequencyMode

The base sample frequency mode.

#### 7.18.2.19 source

**BUInt8** BMeasureApi::Configuration::source

The source number if multiple units are in use.

#### 7.18.2.20 spare1

**BUInt8** BMeasureApi::Configuration::spare1

#### 7.18.2.21 spare3

**BUInt8** BMeasureApi::Configuration::spare3[3]

#### 7.18.2.22 spare4

**BUInt8** BMeasureApi::Configuration::spare4[2]

#### 7.18.2.23 usbaEnable

**BUInt8** BMeasureApi::Configuration::usbaEnable

Enable USB-A interface.

#### 7.18.2.24 usbbEnable

**BUInt8** BMeasureApi::Configuration::usbbEnable

Enable USB-B interface.

#### 7.18.2.25 version

**BUInt32** BMeasureApi::Configuration::version

The configuration version.

#### 7.18.2.26 wifiEnable

**BUInt8** BMeasureApi::Configuration::wifiEnable

Enable wifi interface.

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

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

## 7.19 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.*
- **DataBlockType** [type](#)  
*The type of data block.*
- **BUInt8** [spare](#) [7]
- **BFloat32** [data](#) [117]

### 7.19.1 Member Function Documentation

#### 7.19.1.1 [getMembers\(\)](#)

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

### 7.19.2 Member Data Documentation

#### 7.19.2.1 [data](#)

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



### 7.19.2.2 numChannels

**BUInt16** BMeasureApi::DataBlock::numChannels

The number of data channels.

### 7.19.2.3 numSamples

**BUInt16** BMeasureApi::DataBlock::numSamples

The number of samples.

### 7.19.2.4 sequence

**BUInt32** BMeasureApi::DataBlock::sequence

The sequence number.

### 7.19.2.5 source

**BUInt16** BMeasureApi::DataBlock::source

The source unit.

### 7.19.2.6 spare

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

### 7.19.2.7 status

**BUInt16** BMeasureApi::DataBlock::status

### 7.19.2.8 time

**BUInt64** BMeasureApi::DataBlock::time

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

### 7.19.2.9 type

**DataBlockType** BMeasureApi::DataBlock::type

The type of data block.

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

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

## 7.20 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** [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** [writeInfoTdms](#) (const [InfoBlock](#) &infoBlock, const [ChannelConfigs](#) &channels)
- **BError** [writeInfoBMeas](#) (const [InfoBlock](#) &infoBlock, const [ChannelConfigs](#) &channels)

## Private Attributes

- **BString** `ofilename`
- **BString** `omode`
- **BString** `offormat`
- **BFile** `ofile`
- **BUInt32** `opacketLen`
- **BoapMc1Packet** \* `opacket`

## 7.20.1 Constructor & Destructor Documentation

### 7.20.1.1 DataFile()

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

### 7.20.1.2 ~DataFile()

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

## 7.20.2 Member Function Documentation

### 7.20.2.1 close()

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

Close the file.

### 7.20.2.2 getFileName()

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

Return the file name.

### 7.20.2.3 init()

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

Initialise.

### 7.20.2.4 open()

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

Open the file for read or write.

### 7.20.2.5 readData()

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

Read a block of data.

### 7.20.2.6 readInfo()

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

### 7.20.2.7 validateFormat()

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

### 7.20.2.8 writeData()

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

Write a block of data.

### 7.20.2.9 writeEnd()

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

### 7.20.2.10 writeInfo()

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

### 7.20.2.11 writeInfoBMeas()

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

### 7.20.2.12 writeInfoTdms()

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

## 7.20.3 Member Data Documentation

### 7.20.3.1 ofile

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

### 7.20.3.2 ofileName

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

### 7.20.3.3 oformat

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

### 7.20.3.4 omode

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

### 7.20.3.5 opacket

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

### 7.20.3.6 opacketLen

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

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

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

## 7.21 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](#)

### 7.21.1 Detailed Description

The [Dfu](#) access class.

### 7.21.2 Constructor & Destructor Documentation

#### 7.21.2.1 Dfu()

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

#### 7.21.2.2 ~Dfu()

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

### 7.21.3 Member Function Documentation

#### 7.21.3.1 clearStatus()

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

#### 7.21.3.2 connect()

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

Connect to USB DFU device.

### 7.21.3.3 detectDevice()

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

Check if DFU devuce exists.

### 7.21.3.4 disconnect()

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

Disconnect from USB DFU device.

### 7.21.3.5 getStatus()

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

### 7.21.3.6 init()

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

Initialise.

### 7.21.3.7 reset()

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

Reset.

### 7.21.3.8 upload()

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

Upload a file.



### 7.21.3.9 upload\_cmd()

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

### 7.21.3.10 validateFile()

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

Check if file is valid firmware.

## 7.21.4 Member Data Documentation

### 7.21.4.1 oconnected

```
Bool Dfu::oconnected [private]
```

### 7.21.4.2 ocontext

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

### 7.21.4.3 odev

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

### 7.21.4.4 overbose

```
Bool Dfu::overbose [private]
```

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

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

## 7.22 DfuStatus Struct Reference

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

### Public Attributes

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

### 7.22.1 Member Data Documentation

#### 7.22.1.1 iString

```
BUInt8 DfuStatus::iString
```

#### 7.22.1.2 pollTimeout

```
BUInt DfuStatus::pollTimeout
```

#### 7.22.1.3 state

```
BUInt8 DfuStatus::state
```

#### 7.22.1.4 status

```
BUInt8 DfuStatus::status
```

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

- [Dfu.h](#)

## 7.23 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]

### 7.23.1 Member Function Documentation

#### 7.23.1.1 getMembers()

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

### 7.23.2 Member Data Documentation

#### 7.23.2.1 data

```
BUInt8 BMeasureApi::FileData::data[512]
```

#### 7.23.2.2 length

```
BUInt32 BMeasureApi::FileData::length
```

The data length.

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

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

## 7.24 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.*

## 7.24.1 Member Function Documentation

### 7.24.1.1 [getMembers\(\)](#)

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

## 7.24.2 Member Data Documentation

### 7.24.2.1 [fileLength](#)

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

The file length.

### 7.24.2.2 [fileType](#)

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

The file type.

### 7.24.2.3 name

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

### 7.24.2.4 spare

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

### 7.24.2.5 time

**BTime** BMeasureApi::FileInfo::time

The file date/time.

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

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

## 7.25 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.*

### 7.25.1 Member Function Documentation

### 7.25.1.1 getMembers()

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

## 7.25.2 Member Data Documentation

### 7.25.2.1 free

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

The store free space.

### 7.25.2.2 name

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

### 7.25.2.3 size

```
BUInt64 BMeasureApi::FilesysInfo::size
```

The store size.

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

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

## 7.26 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.*
- **BUInt16** [spare0](#)
- **BChar** [name](#) [16]
- **BChar** [location](#) [16]
- [NodeInfo](#) [nodeInfo](#)  
*Information on the unit.*
- [MeasurementConfig](#) [measureConfig](#)  
*The measurement configuration.*

## 7.26.1 Member Function Documentation

### 7.26.1.1 [getMembers\(\)](#)

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

## 7.26.2 Member Data Documentation

### 7.26.2.1 [location](#)

```
BChar BMeasureApi::InfoBlock::location[16]
```

### 7.26.2.2 [measureConfig](#)

```
MeasurementConfig BMeasureApi::InfoBlock::measureConfig
```

The measurement configuration.

### 7.26.2.3 name

**BChar** BMeasureApi::InfoBlock::name[16]

### 7.26.2.4 nodeInfo

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

[Information](#) on the unit.

### 7.26.2.5 numChannels

**BUInt16** BMeasureApi::InfoBlock::numChannels

The number of data channels.

### 7.26.2.6 source

**BUInt16** BMeasureApi::InfoBlock::source

The source unit.

### 7.26.2.7 spare0

**BUInt16** BMeasureApi::InfoBlock::spare0

### 7.26.2.8 time

**BUInt64** BMeasureApi::InfoBlock::time

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



## 7.26.2.9 version

```
BUInt16 BMeasureApi::InfoBlock::version
```

The info/data version.

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

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

## 7.27 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.*
- **BUInt32** `networkAddress`  
*The network IP address.*
- **BUInt32** `networkMask`  
*The network netmask.*
- **BUInt32** `networkGateway`  
*The network gateway.*
- **BUInt32** `networkTimeServer`  
*The network time server.*
- **BUInt8** `spare1` [32]

### 7.27.1 Member Function Documentation

### 7.27.1.1 getMembers()

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

## 7.27.2 Member Data Documentation

### 7.27.2.1 networkAddress

```
BUInt32 BMeasureApi::Information::networkAddress
```

The network IP address.

### 7.27.2.2 networkGateway

```
BUInt32 BMeasureApi::Information::networkGateway
```

The network gateway.

### 7.27.2.3 networkMask

```
BUInt32 BMeasureApi::Information::networkMask
```

The network netmask.

### 7.27.2.4 networkMode

```
BUInt32 BMeasureApi::Information::networkMode
```

The network Mode.

### 7.27.2.5 networkTimeServer

```
BUInt32 BMeasureApi::Information::networkTimeServer
```

The network time server.

### 7.27.2.6 nodeInfo

`NodeInfo` BMeasureApi::Information::nodeInfo

### 7.27.2.7 numChannels

`BUInt8` BMeasureApi::Information::numChannels

The number of channels.

### 7.27.2.8 numConfigItems

`BUInt8` BMeasureApi::Information::numConfigItems

The number of config items.

### 7.27.2.9 spare0

`BUInt8` BMeasureApi::Information::spare0[2]

### 7.27.2.10 spare1

`BUInt8` BMeasureApi::Information::spare1[32]

### 7.27.2.11 time

`BTimeUs` BMeasureApi::Information::time

The system time.

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

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

## 7.28 BMeasureApi::MeasurementConfig Class Reference

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

### Static Public Member Functions

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

### Public Attributes

- [MeasureMode](#) [measureMode](#)
- [TriggerMode](#) [triggerMode](#)
- [TriggerConfig](#) [triggerConfig](#)  
*Trigger config including direction, filters etc.*
- **BUInt8** [triggerChannel](#)
- **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 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]

### 7.28.1 Member Function Documentation

#### 7.28.1.1 [getMembers\(\)](#)

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

### 7.28.2 Member Data Documentation

#### 7.28.2.1 [description](#)

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

### 7.28.2.2 measureMode

`MeasureMode` BMeasureApi::MeasurementConfig::measureMode

### 7.28.2.3 measurePeriod

`BUInt32` BMeasureApi::MeasurementConfig::measurePeriod

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

### 7.28.2.4 numSamples0

`BUInt32` BMeasureApi::MeasurementConfig::numSamples0

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

### 7.28.2.5 numSamples1

`BUInt32` BMeasureApi::MeasurementConfig::numSamples1

The number of samples to capture. 0 is continuous.

### 7.28.2.6 numSamplesBlock

`BUInt32` BMeasureApi::MeasurementConfig::numSamplesBlock

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

### 7.28.2.7 sampleRate

`BFloat64` BMeasureApi::MeasurementConfig::sampleRate

### 7.28.2.8 triggerChannel

`BUInt8` BMeasureApi::MeasurementConfig::triggerChannel

### 7.28.2.9 triggerConfig

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

Trigger config including direction, filters etc.

### 7.28.2.10 triggerDelay

**BUint32** `BMeasureApi::MeasurementConfig::triggerDelay`

Trigger delay in samples.

### 7.28.2.11 triggerLevel

**BFloat64** `BMeasureApi::MeasurementConfig::triggerLevel`

### 7.28.2.12 triggerMode

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

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

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

## 7.29 BMeasureApi::NodeInfo Class Reference

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

### Static Public Member Functions

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

### Public Attributes

- **BUint32** [apiVersion](#)
- [Version](#) [hardwareVersion](#)
- [Version](#) [fpgaVersion](#)
- [Version](#) [softwareVersion](#)
- **BChar** [serialNumber](#) [12]

## 7.29.1 Member Function Documentation

### 7.29.1.1 getMembers()

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

## 7.29.2 Member Data Documentation

### 7.29.2.1 apiVersion

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

### 7.29.2.2 fpgaVersion

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

### 7.29.2.3 hardwareVersion

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

### 7.29.2.4 serialNumber

```
BChar BMeasureApi::NodeInfo::serialNumber[12]
```

### 7.29.2.5 softwareVersion

```
Version BMeasureApi::NodeInfo::softwareVersion
```

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

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

## 7.30 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** [spare](#) [3]

### 7.30.1 Member Function Documentation

#### 7.30.1.1 [getMembers\(\)](#)

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

### 7.30.2 Member Data Documentation

#### 7.30.2.1 [error](#)

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

#### 7.30.2.2 [errorStr](#)

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



### 7.30.2.3 mode

`Mode` BMeasureApi::NodeStatus::mode

### 7.30.2.4 spare

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

### 7.30.2.5 status

`BUInt32` BMeasureApi::NodeStatus::status

### 7.30.2.6 time

`BTimeUs` BMeasureApi::NodeStatus::time

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

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

## 7.31 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](#)

### 7.31.1 Member Function Documentation

### 7.31.1.1 getMembers()

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

## 7.31.2 Member Data Documentation

### 7.31.2.1 type

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

### 7.31.2.2 ver0

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

### 7.31.2.3 ver1

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

### 7.31.2.4 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 <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.cpp File Reference

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

## Namespaces

- [BMeasureApi](#)

## 8.4 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

## 8.5 BMeasureD.cpp File Reference

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

## Namespaces

- [BMeasureApi](#)

## Macros

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

### 8.5.1 Macro Definition Documentation

### 8.5.1.1 boffsetof

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

## 8.6 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::Configuration](#)
- class [BMeasureApi::ConfigItem](#)
- class [BMeasureApi::MeasurementConfig](#)
- class [BMeasureApi::DataBlock](#)
- 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::ErrorSystem` = 64, `BMeasureApi::ErrorDataOverrun` = 65 }
- enum `BMeasureApi::NodeType` { `BMeasureApi::NodeTypeNone` = 0, `BMeasureApi::NodeTypeBMeasure1` = 1 }
- enum `BMeasureApi::SecureMode` { `BMeasureApi::SecureModeOpen`, `BMeasureApi::SecureMoteRemote`, `BMeasureApi::SecureModeFull` }
- 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::StatusDataOverrun` = 0x40, `BMeasureApi::StatusFpgaOverrun` = 0x80 }
- enum `BMeasureApi::Mode` { `BMeasureApi::ModelIdle` = 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` = 0x81, `BMeasureApi::ChannelTypeDigitalIn` = 2, `BMeasureApi::ChannelTypeDigitalOut` = 0x82 }
- 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::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::DigitalInOut` = 2, `BMeasureApi::DigitalModeSyncMaster` = 3, `BMeasureApi::DigitalModeSyncSlave` = 4 }
- enum `BMeasureApi::Waveform` { `BMeasureApi::WaveformNone`, `BMeasureApi::WaveformDc`, `BMeasureApi::WaveformSine`, `BMeasureApi::WaveformSquare`, `BMeasureApi::WaveformTriangle`, `BMeasureApi::WaveformNoise`, `BMeasureApi::WaveformArbitrary` }
- 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::LogDataMode` { `BMeasureApi::LogDataModeNormal`, `BMeasureApi::LogDataModeDeleteOld` }
- enum `BMeasureApi::DataBlockType` { `BMeasureApi::DataBlockTypeFloat32`, `BMeasureApi::DataBlockType125i` }
- enum `BMeasureApi::DataSend` { `BMeasureApi::DataSendOff`, `BMeasureApi::DataSendOn` }
- 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 }



- 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 }

## 8.7 BMeasureLib.cpp File Reference

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

### Namespaces

- [BMeasureApi](#)

### Macros

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

### 8.7.1 Macro Definition Documentation

#### 8.7.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

#### 8.7.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

## 8.8 BMeasureLib.h File Reference

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

### Namespaces

- [BMeasureApi](#)

## Typedefs

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

## 8.9 BMeasureS.cpp File Reference

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

## Namespaces

- [BMeasureApi](#)

## 8.10 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>
```

## Namespaces

- [BMeasureApi](#)

## Macros

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

## 8.10.1 Macro Definition Documentation

### 8.10.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

### 8.10.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

### 8.10.1.3 CONVERT\_FLOAT

```
#define CONVERT_FLOAT 0
```

Convert to floating point.

## 8.11 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)

## 8.12 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)

### 8.12.1 Macro Definition Documentation

#### 8.12.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

#### 8.12.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

#### 8.12.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

## 8.13 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](#)

## 8.14 CommsNet.cpp File Reference

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

### Namespaces

- [BMeasureApi](#)

### Macros

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

### 8.14.1 Macro Definition Documentation

#### 8.14.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

### 8.14.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

### 8.14.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

## 8.15 CommsNet.h File Reference

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

### Classes

- class [BMeasureApi::CommsNet](#)

### Namespaces

- [BMeasureApi](#)

## 8.16 CommsSerial.cpp File Reference

## 8.17 CommsSerial.h File Reference

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

### Classes

- class [BMeasureApi::CommsSerial](#)

### Namespaces

- [BMeasureApi](#)

## 8.18 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`

#### 8.18.1 Macro Definition Documentation

##### 8.18.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

##### 8.18.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

## 8.19 CommsUsb.h File Reference

```
#include <BComms.h>
#include <BMutex.h>
#include <libusb-1.0/libusb.h>
```

### Classes

- class [BMeasureApi::CommsUsb](#)

## Namespaces

- [BMeasureApi](#)

## 8.20 DataFile.cpp File Reference

```
#include <DataFile.h>
#include <BoapMcl.h>
#include <BBuffer.h>
#include <BDebug.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](#),  
[BMeasureApi::TdsTypeString](#) =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)

### 8.20.1 Macro Definition Documentation



### 8.20.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

### 8.20.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

## 8.21 DataFile.h File Reference

```
#include <BString.h>  
#include <BFile.h>  
#include <BMeasureLib.h>  
#include <BoapMcl.h>
```

### Classes

- class [BMeasureApi::DataFile](#)

### Namespaces

- [BMeasureApi](#)

## 8.22 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

### 8.22.1 Macro Definition Documentation

#### 8.22.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

#### 8.22.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

#### 8.22.1.3 DFU\_ABORT

```
#define DFU_ABORT 6
```

#### 8.22.1.4 DFU\_CLRSTATUS

```
#define DFU_CLRSTATUS 4
```

#### 8.22.1.5 DFU\_DETACH

```
#define DFU_DETACH 0
```

### 8.22.1.6 DFU\_DNLOAD

```
#define DFU_DNLOAD 1
```

### 8.22.1.7 DFU\_GETSTATE

```
#define DFU_GETSTATE 5
```

### 8.22.1.8 DFU\_GETSTATUS

```
#define DFU_GETSTATUS 3
```

### 8.22.1.9 DFU\_IFF\_ALT

```
#define DFU_IFF_ALT 0x1000
```

### 8.22.1.10 DFU\_IFF\_CONFIG

```
#define DFU_IFF_CONFIG 0x0400
```

### 8.22.1.11 DFU\_IFF\_DEVNUM

```
#define DFU_IFF_DEVNUM 0x2000
```

### 8.22.1.12 DFU\_IFF\_DFU

```
#define DFU_IFF_DFU 0x0001 /* DFU Mode, (not Runtime) */
```

### 8.22.1.13 DFU\_IFF\_IFACE

```
#define DFU_IFF_IFACE 0x0800
```

**8.22.1.14 DFU\_IFF\_PATH**

```
#define DFU_IFF_PATH 0x4000
```

**8.22.1.15 DFU\_IFF\_PRODUCT**

```
#define DFU_IFF_PRODUCT 0x0200
```

**8.22.1.16 DFU\_IFF\_VENDOR**

```
#define DFU_IFF_VENDOR 0x0100
```

**8.22.1.17 DFU\_STATUS\_ERROR\_ADDRESS**

```
#define DFU_STATUS_ERROR_ADDRESS 0x08
```

**8.22.1.18 DFU\_STATUS\_ERROR\_CHECK\_ERASED**

```
#define DFU_STATUS_ERROR_CHECK_ERASED 0x05
```

**8.22.1.19 DFU\_STATUS\_ERROR\_ERASE**

```
#define DFU_STATUS_ERROR_ERASE 0x04
```

**8.22.1.20 DFU\_STATUS\_ERROR\_FILE**

```
#define DFU_STATUS_ERROR_FILE 0x02
```

**8.22.1.21 DFU\_STATUS\_ERROR\_FIRMWARE**

```
#define DFU_STATUS_ERROR_FIRMWARE 0x0a
```

**8.22.1.22 DFU\_STATUS\_ERROR\_NOTDONE**

```
#define DFU_STATUS_ERROR_NOTDONE 0x09
```

**8.22.1.23 DFU\_STATUS\_ERROR\_POR**

```
#define DFU_STATUS_ERROR_POR 0x0d
```

**8.22.1.24 DFU\_STATUS\_ERROR\_PROG**

```
#define DFU_STATUS_ERROR_PROG 0x06
```

**8.22.1.25 DFU\_STATUS\_ERROR\_STALLEDPKT**

```
#define DFU_STATUS_ERROR_STALLEDPKT 0x0f
```

**8.22.1.26 DFU\_STATUS\_ERROR\_TARGET**

```
#define DFU_STATUS_ERROR_TARGET 0x01
```

**8.22.1.27 DFU\_STATUS\_ERROR\_UNKNOWN**

```
#define DFU_STATUS_ERROR_UNKNOWN 0x0e
```

**8.22.1.28 DFU\_STATUS\_ERROR\_USBR**

```
#define DFU_STATUS_ERROR_USBR 0x0c
```

**8.22.1.29 DFU\_STATUS\_ERROR\_VENDOR**

```
#define DFU_STATUS_ERROR_VENDOR 0x0b
```

**8.22.1.30 DFU\_STATUS\_ERROR\_VERIFY**

```
#define DFU_STATUS_ERROR_VERIFY 0x07
```

**8.22.1.31 DFU\_STATUS\_ERROR\_WRITE**

```
#define DFU_STATUS_ERROR_WRITE 0x03
```

**8.22.1.32 DFU\_STATUS\_OK**

```
#define DFU_STATUS_OK 0x00
```

**8.22.1.33 DFU\_UPLOAD**

```
#define DFU_UPLOAD 2
```

**8.22.1.34 STATE\_APP\_DETACH**

```
#define STATE_APP_DETACH 0x01
```

**8.22.1.35 STATE\_APP\_IDLE**

```
#define STATE_APP_IDLE 0x00
```

**8.22.1.36 STATE\_DFU\_DOWNLOAD\_BUSY**

```
#define STATE_DFU_DOWNLOAD_BUSY 0x04
```

**8.22.1.37 STATE\_DFU\_DOWNLOAD\_IDLE**

```
#define STATE_DFU_DOWNLOAD_IDLE 0x05
```

### 8.22.1.38 STATE\_DFU\_DOWNLOAD\_SYNC

```
#define STATE_DFU_DOWNLOAD_SYNC 0x03
```

### 8.22.1.39 STATE\_DFU\_ERROR

```
#define STATE_DFU_ERROR 0x0a
```

### 8.22.1.40 STATE\_DFU\_IDLE

```
#define STATE_DFU_IDLE 0x02
```

### 8.22.1.41 STATE\_DFU\_MANIFEST

```
#define STATE_DFU_MANIFEST 0x07
```

### 8.22.1.42 STATE\_DFU\_MANIFEST\_SYNC

```
#define STATE_DFU_MANIFEST_SYNC 0x06
```

### 8.22.1.43 STATE\_DFU\_MANIFEST\_WAIT\_RESET

```
#define STATE_DFU_MANIFEST_WAIT_RESET 0x08
```

### 8.22.1.44 STATE\_DFU\_UPLOAD\_IDLE

```
#define STATE_DFU_UPLOAD_IDLE 0x09
```

## 8.22.2 Enumeration Type Documentation

### 8.22.2.1 dfuse\_command

```
enum dfuse_command
```



## Enumerator

SET_ADDRESS	
ERASE_PAGE	
MASS_ERASE	
READ_UNPROTECT	

## 8.22.3 Function Documentation

## 8.22.3.1 pageAddress()

```
static BUInt32 pageAddress (
    BUInt32 page ) [static]
```

## 8.22.3.2 pageNumber()

```
static BInt32 pageNumber (
    BUInt32 address ) [static]
```

## 8.22.4 Variable Documentation

## 8.22.4.1 BFirmwareInfoEncrypt1

```
const BUInt8 BFirmwareInfoEncrypt1 = 0x40
```

## 8.22.4.2 BFirmwareInfoMagic

```
const BUInt32 BFirmwareInfoMagic = 0xBBEEAA00
```

## 8.23 Dfu.h File Reference

```
#include <BError.h>
#include <libusb-1.0/libusb.h>
```

## Classes

- struct [DfuStatus](#)
- class [Dfu](#)  
The *Dfu* access class.

## 8.24 overview.dox File Reference



# Index

- ~BMdns
  - BMdns, 31
- ~BMeasureUnit
  - BMeasureApi::BMeasureUnit, 52
- ~BMeasureUnits
  - BMeasureApi::BMeasureUnits, 63
- ~BMeasureUnitsDataBlock
  - BMeasureApi::BMeasureUnitsDataBlock, 75
- ~CommsNet
  - BMeasureApi::CommsNet, 85
- ~CommsSerial
  - BMeasureApi::CommsSerial, 87
- ~CommsUsb
  - BMeasureApi::CommsUsb, 90
- ~DataFile
  - BMeasureApi::DataFile, 103
- ~Dfu
  - Dfu, 107
- address
  - BMdnsService, 32
- amplitude
  - BMeasureApi::AwgConfig, 28
- apiVersion
  - BMeasureApi, 26
  - BMeasureApi::NodeInfo, 123
- attenuator
  - BMeasureApi::ChannelConfig, 81
- AwgOutput
  - BMeasureApi, 17
- BDEBUGL1
  - BMdns.cpp, 127
  - BMeasureLib.cpp, 133
  - BMeasureUnit.cpp, 135
  - BMeasureUnits.cpp, 136
  - CommsNet.cpp, 137
  - CommsUsb.cpp, 139
  - DataFile.cpp, 140
  - Dfu.cpp, 143
- BDEBUGL2
  - BMeasureLib.cpp, 133
  - BMeasureUnit.cpp, 135
  - BMeasureUnits.cpp, 136
  - CommsNet.cpp, 137
  - CommsUsb.cpp, 139
  - DataFile.cpp, 141
  - Dfu.cpp, 143
- BDEBUGL3
  - BMeasureUnits.cpp, 136
- CommsNet.cpp, 138
- BFirmwareInfo, 29
  - checksum, 29
  - length, 29
  - magic, 29
  - type, 29
  - ver0, 30
  - ver1, 30
  - ver2, 30
- BFirmwareInfoEncrypt1
  - Dfu.cpp, 149
- BFirmwareInfoMagic
  - Dfu.cpp, 149
- BMdns, 30
  - ~BMdns, 31
  - BMdns, 30
  - findServices, 31
  - init, 31
  - osocket, 31
  - otransactionId, 31
- BMdns.cpp, 127
  - BDEBUGL1, 127
  - mdns\_read\_string, 129
  - mdns\_read\_strings, 129
  - mdns\_write\_string, 129
  - MdnsClass, 128
  - MdnsEntryType, 128
  - MdnsRecordType, 128
- BMdns.h, 129
- BMdnsService, 32
  - address, 32
  - extra, 32
  - hostname, 32
  - name, 32
- BMeasure
  - BMeasureApi::BMeasure, 35
- BMeasureApi, 15
  - apiVersion, 26
  - AwgOutput, 17
  - BlockTypes, 17
  - CalibrateStage, 18
  - ChannelConfigs, 17
  - ChannelType, 18
  - channelTypeString, 24
  - DataBlockType, 18
  - DataSend, 19
  - DigitalMode, 19
  - ErrorNum, 19
  - FileType, 20

- FilesysDeleteType, 19
- LogDataMode, 20
- MeasureMode, 20
- MessageSource, 20
- Mode, 21
- NetworkMode, 21
- NodeType, 21
- round512, 24
- SampleType, 21
- sampleTypeString, 25
- SecureMode, 22
- Status, 22
- SyncMode, 22
- TdsDataType, 23
- toFloat, 26
- TocBigEndian, 25
- TocDaqRawData, 25
- TocInterleavedData, 25
- TocMetaData, 25
- TocNewObjList, 25
- TocRawData, 25
- TriggerConfig, 23
- TriggerMode, 24
- unitSort, 26
- Waveform, 24
- BMeasureApi::AwgConfig, 27
  - amplitude, 28
  - duty, 28
  - frequency, 28
  - getMembers, 27
  - offset, 28
  - output, 28
  - spare, 28
  - waveform, 28
- BMeasureApi::BMeasure, 33
  - BMeasure, 35
  - calibrate, 36
  - calibrateServe, 36
  - factoryReset, 36
  - factoryResetServe, 36
  - fileClose, 36
  - fileCloseServe, 36
  - fileDelete, 37
  - fileDeleteServe, 37
  - fileList, 37
  - fileListServe, 37
  - fileOpen, 37
  - fileOpenServe, 37
  - fileRead, 38
  - fileReadServe, 38
  - fileWrite, 39
  - fileWriteServe, 39
  - filesysDelete, 38
  - filesysDeleteServe, 38
  - filesysInfo, 38
  - filesysInfoServe, 38
  - functionUnLock, 39
  - functionUnLockServe, 39
  - getAwgConfig, 39
  - getAwgConfigServe, 39
  - getBoardConfig, 40
  - getBoardConfigServe, 40
  - getChannelConfig, 40
  - getChannelConfigServe, 40
  - getConfig, 40
  - getConfigServe, 40
  - getDigital, 41
  - getDigitalServe, 41
  - getInfoBlock, 41
  - getInfoBlockServe, 41
  - getInformation, 41
  - getInformationServe, 41
  - getMeasurement, 42
  - getMeasurementConfig, 42
  - getMeasurementConfigServe, 42
  - getMeasurementServe, 42
  - getNodeInfo, 42
  - getNodeInfoServe, 42
  - getStatus, 43
  - getStatusServe, 43
  - getSwitch, 43
  - getSwitchServe, 43
  - login, 43
  - loginServe, 43
  - measure, 44
  - measureServe, 44
  - processRequest, 44
  - runBoardTest, 44
  - runBoardTestServe, 44
  - sendData, 44
  - sendDataEnable, 45
  - sendDataEnableServe, 45
  - sendDataServe, 45
  - sendInfo, 45
  - sendInfoServe, 45
  - sendMessage, 45
  - sendMessageServe, 46
  - sendStatus, 46
  - sendStatusServe, 46
  - sendTime, 46
  - sendTimeServe, 46
  - setAnalogueOut, 46
  - setAnalogueOutServe, 47
  - setAwgConfig, 47
  - setAwgConfigServe, 47
  - setAwgWaveform, 47
  - setAwgWaveformServe, 47
  - setBoardConfig, 47
  - setBoardConfigServe, 48
  - setChannelConfig, 48
  - setChannelConfigFull, 48
  - setChannelConfigFullServe, 48
  - setChannelConfigServe, 48
  - setConfig, 48
  - setConfigServe, 49
  - setDigital, 49

- setDigitalServe, 49
- setMeasurement, 49
- setMeasurementConfig, 49
- setMeasurementConfigServe, 49
- setMeasurementServe, 50
- setMode, 50
- setModeServe, 50
- setRelay, 50
- setRelayServe, 50
- setSecureMode, 50
- setSecureModeServe, 51
- BMeasureApi::BMeasureUnit, 51
  - ~BMeasureUnit, 52
  - BMeasureUnit, 52
  - blockNumChannels, 55
  - blockNumSamples, 55
  - connect, 53
  - device, 53
  - disconnect, 53
  - disconnected, 53
  - findDevices, 53
  - findDevicesNetwork, 53
  - findDevicesUsb, 53
  - info, 54
  - numChannels, 54
  - oblockCount, 55
  - ochannels, 56
  - oconfigMeasurement, 56
  - odataBlock, 56
  - odevice, 56
  - odisconnecting, 56
  - oinfo, 56
  - onodeInfo, 56
  - osampleCount, 57
  - osequenceNext, 57
  - processdataBlock, 54
  - run, 54
  - sendDataServe, 54
  - sendDataServe1, 54
  - serialNumber, 55
  - setChannelConfig, 55
  - setMeasurement, 55
- BMeasureApi::BMeasureUnit1, 57
  - BMeasureUnit1, 58
  - disconnected, 58
  - oconnected, 59
  - oenabled, 59
  - omeasureUnits, 59
  - oorder, 59
  - oserialNumber, 60
  - osource, 60
  - sendDataServe1, 58
  - sendMessageServe, 58
  - serialNumber, 59
  - setSerialNumber, 59
- BMeasureApi::BMeasureUnitDevice, 60
  - BMeasureUnitDevice, 60
  - device, 61
  - serialNumber, 61
- BMeasureApi::BMeasureUnits, 61
  - ~BMeasureUnits, 63
  - BMeasureUnits, 63
  - clear, 64
  - dataAvailable, 64
  - dataClear, 64
  - dataDone, 64
  - dataEvent, 64
  - dataProcessEnable, 64
  - dataRead, 64
  - dataSetNumStreams, 65
  - dataWait, 65
  - debugPrint, 65
  - disconnected, 65
  - getAwgConfig, 65
  - getChannelConfig, 65
  - getConfig, 66
  - getFreeBlock, 66
  - getInfoBlock, 66
  - getInformation, 66
  - getMeasurement, 66
  - getMeasurementConfig, 66
  - getStatus, 67
  - numChannels, 67
  - odataBlocksFree, 71
  - odataBlocksIn, 71
  - odataBlocksOut, 72
  - odataBlocksOutCount, 72
  - odataBlocksProcess, 72
  - odataBlocksProcessNum, 72
  - odataStreamNum, 72
  - ofill, 72
  - olocalTrigger, 72
  - olockInput, 72
  - olockOutput, 73
  - olockUnits, 73
  - onumBlocks, 73
  - onumChannels, 73
  - onumConnected, 73
  - oprocEnable, 73
  - oprocRunning, 73
  - ostartSample, 74
  - otriggered, 74
  - ounitMaster, 74
  - ounits, 74
  - outputBlock, 67
  - run, 67
  - sendDataEnable, 67
  - sendDataProcess, 67
  - sendDataProcessTrigger, 68
  - sendDataQueue, 68
  - sendDataServe1, 68
  - sendMessage, 68
  - sendMessageServe, 68
  - sendTime, 68
  - setAwgConfig, 68
  - setChannelConfig, 69

- setConfig, 69
- setMeasurement, 69
- setMeasurementConfig, 69
- setMode, 69
- unit, 69
- unitAdd, 70
- unitDelete, 70
- unitMaster, 70
- unitSetEnabled, 71
- unitSetOrder, 71
- unitsConnect, 70
- unitsConnected, 70
- unitsConnectedNum, 70
- unitsDisconnect, 70
- unitsFind, 71
- unitsNum, 71
- BMeasureApi::BMeasureUnitsDataBlock, 74
  - ~BMeasureUnitsDataBlock, 75
  - BMeasureUnitsDataBlock, 75
  - init, 75
  - odataBlock, 75
  - ofill, 75
  - oinUse, 76
- BMeasureApi::BoardConfig, 76
  - buildTime, 77
  - calibAdcOffsets, 77
  - calibAdcScales, 77
  - calibAttenScales, 77
  - calibDacOffsets, 77
  - calibDacScales, 77
  - calibTemp, 77
  - calibTime, 77
  - getMembers, 76
  - hardwareVersion, 78
  - macAddress, 78
  - magic, 78
  - serialNumber, 78
  - spare0, 78
  - testMode, 78
- BMeasureApi::CalibrateInfo, 79
  - calibrateFrequency, 79
  - calibrateTime, 79
  - getMembers, 79
  - stage, 79
  - value, 80
- BMeasureApi::ChannelConfig, 80
  - attenuator, 81
  - calibOffset, 81
  - calibScale, 81
  - calibScaleAtten1, 81
  - dataChannel, 82
  - enabled, 82
  - getMembers, 81
  - id, 82
  - name, 82
  - number, 82
  - offset, 82
  - pgaGain, 83
  - process, 83
  - sampleType, 83
  - scale, 83
  - siUnits, 83
  - spare0, 83
  - type, 83
- BMeasureApi::CommsNet, 84
  - ~CommsNet, 85
  - CommsNet, 84
  - connect, 85
  - disconnect, 85
  - init, 85
  - osocket, 86
  - read, 85
  - readAvailable, 85
  - wait, 86
  - write, 86
  - writeAvailable, 86
  - writeChunks, 86
- BMeasureApi::CommsSerial, 87
  - ~CommsSerial, 87
  - CommsSerial, 87
  - connect, 88
  - disconnect, 88
  - odevice, 89
  - oserialPort, 89
  - read, 88
  - readAvailable, 88
  - wait, 88
  - write, 88
- BMeasureApi::CommsUsb, 89
  - ~CommsUsb, 90
  - CommsUsb, 90
  - connect, 90
  - disconnect, 90
  - obuffer, 91
  - ocontext, 92
  - odev, 92
  - odevice, 92
  - onum, 92
  - oterminated, 92
  - read, 90
  - readAvailable, 91
  - readChunk, 91
  - wait, 91
  - write, 91
- BMeasureApi::ConfigItem, 92
  - getMembers, 93
  - name, 93
  - spare, 93
  - type, 93
  - value, 93
- BMeasureApi::Configuration, 94
  - digitalMode, 95
  - ethernetEnable, 95
  - getMembers, 95
  - location, 95
  - logData, 96

- logDataDevice, 96
- logDataMode, 96
- mode, 96
- name, 96
- networkAddress, 96
- networkGateway, 97
- networkMask, 97
- networkMode, 97
- networkTimeServer, 97
- program, 97
- rs485BaudRate, 97
- rs485Bits, 98
- rs485StopBits, 98
- sampleFrequencyMode, 98
- source, 98
- spare1, 98
- spare3, 98
- spare4, 99
- usbaEnable, 99
- usbbEnable, 99
- version, 99
- wifiEnable, 99
- BMeasureApi::DataBlock, 100
  - data, 100
  - getMembers, 100
  - numChannels, 100
  - numSamples, 101
  - sequence, 101
  - source, 101
  - spare, 101
  - status, 101
  - time, 101
  - type, 102
- BMeasureApi::DataFile, 102
  - ~DataFile, 103
  - close, 103
  - DataFile, 103
  - getFileName, 103
  - init, 103
  - ofile, 105
  - ofFileName, 105
  - ofFormat, 105
  - omode, 106
  - opacket, 106
  - opacketLen, 106
  - open, 104
  - readData, 104
  - readInfo, 104
  - validateFormat, 104
  - writeData, 104
  - writeEnd, 104
  - writeInfo, 105
  - writeInfoBMeas, 105
  - writeInfoTdms, 105
- BMeasureApi::FileData, 110
  - data, 111
  - getMembers, 111
  - length, 111
- BMeasureApi::FileInfo, 111
  - fileLength, 112
  - fileType, 112
  - getMembers, 112
  - name, 112
  - spare, 113
  - time, 113
- BMeasureApi::FilesysInfo, 113
  - free, 114
  - getMembers, 113
  - name, 114
  - size, 114
- BMeasureApi::InfoBlock, 114
  - getMembers, 115
  - location, 115
  - measureConfig, 115
  - name, 115
  - nodeInfo, 116
  - numChannels, 116
  - source, 116
  - spare0, 116
  - time, 116
  - version, 116
- BMeasureApi::Information, 117
  - getMembers, 117
  - networkAddress, 118
  - networkGateway, 118
  - networkMask, 118
  - networkMode, 118
  - networkTimeServer, 118
  - nodeInfo, 118
  - numChannels, 119
  - numConfigItems, 119
  - spare0, 119
  - spare1, 119
  - time, 119
- BMeasureApi::MeasurementConfig, 120
  - description, 120
  - getMembers, 120
  - measureMode, 120
  - measurePeriod, 121
  - numSamples0, 121
  - numSamples1, 121
  - numSamplesBlock, 121
  - sampleRate, 121
  - triggerChannel, 121
  - triggerConfig, 121
  - triggerDelay, 122
  - triggerLevel, 122
  - triggerMode, 122
- BMeasureApi::NodeInfo, 122
  - apiVersion, 123
  - fpgaVersion, 123
  - getMembers, 123
  - hardwareVersion, 123
  - serialNumber, 123
  - softwareVersion, 123
- BMeasureApi::NodeStatus, 124

- error, 124
- errorStr, 124
- getMembers, 124
- mode, 124
- spare, 125
- status, 125
- time, 125
- BMeasureApi::Version, 125
  - getMembers, 125
  - type, 126
  - ver0, 126
  - ver1, 126
  - ver2, 126
- BMeasureB.cpp, 129
- BMeasureB.h, 130
- BMeasureD.cpp, 130
  - boffsetof, 130
- BMeasureD.h, 131
- BMeasureLib.cpp, 133
  - BDEBUGL1, 133
  - BDEBUGL2, 133
- BMeasureLib.h, 133
- BMeasureS.cpp, 134
- BMeasureUnit
  - BMeasureApi::BMeasureUnit, 52
- BMeasureUnit.cpp, 134
  - BDEBUGL1, 135
  - BDEBUGL2, 135
  - CONVERT\_FLOAT, 135
- BMeasureUnit.h, 135
- BMeasureUnit1
  - BMeasureApi::BMeasureUnit1, 58
- BMeasureUnitDevice
  - BMeasureApi::BMeasureUnitDevice, 60
- BMeasureUnits
  - BMeasureApi::BMeasureUnits, 63
- BMeasureUnits.cpp, 136
  - BDEBUGL1, 136
  - BDEBUGL2, 136
  - BDEBUGL3, 136
- BMeasureUnits.h, 137
- BMeasureUnitsDataBlock
  - BMeasureApi::BMeasureUnitsDataBlock, 75
- blockNumChannels
  - BMeasureApi::BMeasureUnit, 55
- blockNumSamples
  - BMeasureApi::BMeasureUnit, 55
- BlockTypes
  - BMeasureApi, 17
- boffsetof
  - BMeasureD.cpp, 130
- buildTime
  - BMeasureApi::BoardConfig, 77
- CONVERT\_FLOAT
  - BMeasureUnit.cpp, 135
- calibAdcOffsets
  - BMeasureApi::BoardConfig, 77
- calibAdcScales
  - BMeasureApi::BoardConfig, 77
- calibAttenScales
  - BMeasureApi::BoardConfig, 77
- calibDacOffsets
  - BMeasureApi::BoardConfig, 77
- calibDacScales
  - BMeasureApi::BoardConfig, 77
- calibOffset
  - BMeasureApi::ChannelConfig, 81
- calibScale
  - BMeasureApi::ChannelConfig, 81
- calibScaleAtten1
  - BMeasureApi::ChannelConfig, 81
- calibTemp
  - BMeasureApi::BoardConfig, 77
- calibTime
  - BMeasureApi::BoardConfig, 77
- calibrate
  - BMeasureApi::BMeasure, 36
- calibrateFrequency
  - BMeasureApi::CalibrateInfo, 79
- calibrateServe
  - BMeasureApi::BMeasure, 36
- CalibrateStage
  - BMeasureApi, 18
- calibrateTime
  - BMeasureApi::CalibrateInfo, 79
- ChannelConfigs
  - BMeasureApi, 17
- ChannelType
  - BMeasureApi, 18
- channelTypeString
  - BMeasureApi, 24
- checksum
  - BFirmwareInfo, 29
- clear
  - BMeasureApi::BMeasureUnits, 64
- clearStatus
  - Dfu, 107
- close
  - BMeasureApi::DataFile, 103
- CommsNet
  - BMeasureApi::CommsNet, 84
- CommsNet.cpp, 137
  - BDEBUGL1, 137
  - BDEBUGL2, 137
  - BDEBUGL3, 138
- CommsNet.h, 138
- CommsSerial
  - BMeasureApi::CommsSerial, 87
- CommsSerial.cpp, 138
- CommsSerial.h, 138
- CommsUsb
  - BMeasureApi::CommsUsb, 90
- CommsUsb.cpp, 139
  - BDEBUGL1, 139
  - BDEBUGL2, 139
- CommsUsb.h, 139



- connect
  - BMeasureApi::BMeasureUnit, [53](#)
  - BMeasureApi::CommsNet, [85](#)
  - BMeasureApi::CommsSerial, [88](#)
  - BMeasureApi::CommsUsb, [90](#)
  - Dfu, [107](#)
- DFU\_ABORT
  - Dfu.cpp, [143](#)
- DFU\_CLRSTATUS
  - Dfu.cpp, [143](#)
- DFU\_DETACH
  - Dfu.cpp, [143](#)
- DFU\_DNLOAD
  - Dfu.cpp, [143](#)
- DFU\_GETSTATUS
  - Dfu.cpp, [144](#)
- DFU\_GETSTATE
  - Dfu.cpp, [144](#)
- DFU\_IFF\_ALT
  - Dfu.cpp, [144](#)
- DFU\_IFF\_CONFIG
  - Dfu.cpp, [144](#)
- DFU\_IFF\_DEVNUM
  - Dfu.cpp, [144](#)
- DFU\_IFF\_DFU
  - Dfu.cpp, [144](#)
- DFU\_IFF\_IFACE
  - Dfu.cpp, [144](#)
- DFU\_IFF\_PATH
  - Dfu.cpp, [144](#)
- DFU\_IFF\_PRODUCT
  - Dfu.cpp, [145](#)
- DFU\_IFF\_VENDOR
  - Dfu.cpp, [145](#)
- DFU\_STATUS\_ERROR\_ADDRESS
  - Dfu.cpp, [145](#)
- DFU\_STATUS\_ERROR\_CHECK\_ERASED
  - Dfu.cpp, [145](#)
- DFU\_STATUS\_ERROR\_ERASE
  - Dfu.cpp, [145](#)
- DFU\_STATUS\_ERROR\_FILE
  - Dfu.cpp, [145](#)
- DFU\_STATUS\_ERROR\_FIRMWARE
  - Dfu.cpp, [145](#)
- DFU\_STATUS\_ERROR\_NOTDONE
  - Dfu.cpp, [145](#)
- DFU\_STATUS\_ERROR\_POR
  - Dfu.cpp, [146](#)
- DFU\_STATUS\_ERROR\_PROG
  - Dfu.cpp, [146](#)
- DFU\_STATUS\_ERROR\_STALLEDPKT
  - Dfu.cpp, [146](#)
- DFU\_STATUS\_ERROR\_TARGET
  - Dfu.cpp, [146](#)
- DFU\_STATUS\_ERROR\_UNKNOWN
  - Dfu.cpp, [146](#)
- DFU\_STATUS\_ERROR\_USBR
  - Dfu.cpp, [146](#)
- DFU\_STATUS\_ERROR\_VENDOR
  - Dfu.cpp, [146](#)
- DFU\_STATUS\_ERROR\_VERIFY
  - Dfu.cpp, [146](#)
- DFU\_STATUS\_ERROR\_WRITE
  - Dfu.cpp, [147](#)
- DFU\_STATUS\_OK
  - Dfu.cpp, [147](#)
- DFU\_UPLOAD
  - Dfu.cpp, [147](#)
- data
  - BMeasureApi::DataBlock, [100](#)
  - BMeasureApi::FileData, [111](#)
- dataAvailable
  - BMeasureApi::BMeasureUnits, [64](#)
- DataBlockType
  - BMeasureApi, [18](#)
- dataChannel
  - BMeasureApi::ChannelConfig, [82](#)
- dataClear
  - BMeasureApi::BMeasureUnits, [64](#)
- dataDone
  - BMeasureApi::BMeasureUnits, [64](#)
- dataEvent
  - BMeasureApi::BMeasureUnits, [64](#)
- DataFile
  - BMeasureApi::DataFile, [103](#)
- DataFile.cpp, [140](#)
- BDEBUGL1, [140](#)
- BDEBUGL2, [141](#)
- DataFile.h, [141](#)
- dataProcessEnable
  - BMeasureApi::BMeasureUnits, [64](#)
- dataRead
  - BMeasureApi::BMeasureUnits, [64](#)
- DataSend
  - BMeasureApi, [19](#)
- dataSetNumStreams
  - BMeasureApi::BMeasureUnits, [65](#)
- dataWait
  - BMeasureApi::BMeasureUnits, [65](#)
- debugPrint
  - BMeasureApi::BMeasureUnits, [65](#)
- description
  - BMeasureApi::MeasurementConfig, [120](#)
- detectDevice
  - Dfu, [107](#)
- device
  - BMeasureApi::BMeasureUnit, [53](#)
  - BMeasureApi::BMeasureUnitDevice, [61](#)
- Dfu, [106](#)
- ~Dfu, [107](#)
- clearStatus, [107](#)
- connect, [107](#)
- detectDevice, [107](#)
- Dfu, [107](#)
- disconnect, [108](#)
- getStatus, [108](#)

- init, [108](#)
- oconnected, [109](#)
- ocontext, [109](#)
- odev, [109](#)
- overbose, [109](#)
- reset, [108](#)
- upload, [108](#)
- upload\_cmd, [108](#)
- validateFile, [109](#)
- Dfu.cpp, [141](#)
  - BDEBUGL1, [143](#)
  - BDEBUGL2, [143](#)
  - BFirmwareInfoEncrypt1, [149](#)
  - BFirmwareInfoMagic, [149](#)
  - DFU\_ABORT, [143](#)
  - DFU\_CLRSTATUS, [143](#)
  - DFU\_DETACH, [143](#)
  - DFU\_DNLOAD, [143](#)
  - DFU\_GETSTATUS, [144](#)
  - DFU\_GETSTATE, [144](#)
  - DFU\_IFF\_ALT, [144](#)
  - DFU\_IFF\_CONFIG, [144](#)
  - DFU\_IFF\_DEVNUM, [144](#)
  - DFU\_IFF\_DFU, [144](#)
  - DFU\_IFF\_IFACE, [144](#)
  - DFU\_IFF\_PATH, [144](#)
  - DFU\_IFF\_PRODUCT, [145](#)
  - DFU\_IFF\_VENDOR, [145](#)
  - DFU\_STATUS\_ERROR\_ADDRESS, [145](#)
  - DFU\_STATUS\_ERROR\_CHECK\_ERASED, [145](#)
  - DFU\_STATUS\_ERROR\_ERASE, [145](#)
  - DFU\_STATUS\_ERROR\_FILE, [145](#)
  - DFU\_STATUS\_ERROR\_FIRMWARE, [145](#)
  - DFU\_STATUS\_ERROR\_NOTDONE, [145](#)
  - DFU\_STATUS\_ERROR\_POR, [146](#)
  - DFU\_STATUS\_ERROR\_PROG, [146](#)
  - DFU\_STATUS\_ERROR\_STALLEDPKT, [146](#)
  - DFU\_STATUS\_ERROR\_TARGET, [146](#)
  - DFU\_STATUS\_ERROR\_UNKNOWN, [146](#)
  - DFU\_STATUS\_ERROR\_USBR, [146](#)
  - DFU\_STATUS\_ERROR\_VENDOR, [146](#)
  - DFU\_STATUS\_ERROR\_VERIFY, [146](#)
  - DFU\_STATUS\_ERROR\_WRITE, [147](#)
  - DFU\_STATUS\_OK, [147](#)
  - DFU\_UPLOAD, [147](#)
  - dfuse\_command, [148](#)
  - pageAddress, [149](#)
  - pageNumber, [149](#)
  - STATE\_APP\_DETACH, [147](#)
  - STATE\_APP\_IDLE, [147](#)
  - STATE\_DFU\_DOWNLOAD\_BUSY, [147](#)
  - STATE\_DFU\_DOWNLOAD\_IDLE, [147](#)
  - STATE\_DFU\_DOWNLOAD\_SYNC, [147](#)
  - STATE\_DFU\_ERROR, [148](#)
  - STATE\_DFU\_IDLE, [148](#)
  - STATE\_DFU\_MANIFEST\_SYNC, [148](#)
  - STATE\_DFU\_MANIFEST\_WAIT\_RESET, [148](#)
  - STATE\_DFU\_MANIFEST, [148](#)
  - STATE\_DFU\_UPLOAD\_IDLE, [148](#)
- Dfu.h, [149](#)
- DfuStatus, [110](#)
  - iString, [110](#)
  - pollTimeout, [110](#)
  - state, [110](#)
  - status, [110](#)
- dfuse\_command
  - Dfu.cpp, [148](#)
- DigitalMode
  - BMeasureApi, [19](#)
- digitalMode
  - BMeasureApi::Configuration, [95](#)
- disconnect
  - BMeasureApi::BMeasureUnit, [53](#)
  - BMeasureApi::CommsNet, [85](#)
  - BMeasureApi::CommsSerial, [88](#)
  - BMeasureApi::CommsUsb, [90](#)
  - Dfu, [108](#)
- disconnected
  - BMeasureApi::BMeasureUnit, [53](#)
  - BMeasureApi::BMeasureUnit1, [58](#)
  - BMeasureApi::BMeasureUnits, [65](#)
- duty
  - BMeasureApi::AwgConfig, [28](#)
- enabled
  - BMeasureApi::ChannelConfig, [82](#)
- error
  - BMeasureApi::NodeStatus, [124](#)
- ErrorNum
  - BMeasureApi, [19](#)
- errorStr
  - BMeasureApi::NodeStatus, [124](#)
- ethernetEnable
  - BMeasureApi::Configuration, [95](#)
- extra
  - BMdnsService, [32](#)
- factoryReset
  - BMeasureApi::BMeasure, [36](#)
- factoryResetServe
  - BMeasureApi::BMeasure, [36](#)
- fileClose
  - BMeasureApi::BMeasure, [36](#)
- fileCloseServe
  - BMeasureApi::BMeasure, [36](#)
- fileDelete
  - BMeasureApi::BMeasure, [37](#)
- fileDeleteServe
  - BMeasureApi::BMeasure, [37](#)
- fileLength
  - BMeasureApi::FileInfo, [112](#)
- fileList
  - BMeasureApi::BMeasure, [37](#)
- fileListServe
  - BMeasureApi::BMeasure, [37](#)
- fileOpen
  - BMeasureApi::BMeasure, [37](#)

- fileOpenServe
  - BMeasureApi::BMeasure, 37
- fileRead
  - BMeasureApi::BMeasure, 38
- fileReadServe
  - BMeasureApi::BMeasure, 38
- FileType
  - BMeasureApi, 20
- fileType
  - BMeasureApi::FileInfo, 112
- fileWrite
  - BMeasureApi::BMeasure, 39
- fileWriteServe
  - BMeasureApi::BMeasure, 39
- fileysDelete
  - BMeasureApi::BMeasure, 38
- fileysDeleteServe
  - BMeasureApi::BMeasure, 38
- FileysDeleteType
  - BMeasureApi, 19
- fileysInfo
  - BMeasureApi::BMeasure, 38
- fileysInfoServe
  - BMeasureApi::BMeasure, 38
- findDevices
  - BMeasureApi::BMeasureUnit, 53
- findDevicesNetwork
  - BMeasureApi::BMeasureUnit, 53
- findDevicesUsb
  - BMeasureApi::BMeasureUnit, 53
- findServices
  - BMdns, 31
- fpgaVersion
  - BMeasureApi::NodeInfo, 123
- free
  - BMeasureApi::FilesysInfo, 114
- frequency
  - BMeasureApi::AwgConfig, 28
- functionUnLock
  - BMeasureApi::BMeasure, 39
- functionUnLockServe
  - BMeasureApi::BMeasure, 39
- getAwgConfig
  - BMeasureApi::BMeasure, 39
  - BMeasureApi::BMeasureUnits, 65
- getAwgConfigServe
  - BMeasureApi::BMeasure, 39
- getBoardConfig
  - BMeasureApi::BMeasure, 40
- getBoardConfigServe
  - BMeasureApi::BMeasure, 40
- getChannelConfig
  - BMeasureApi::BMeasure, 40
  - BMeasureApi::BMeasureUnits, 65
- getChannelConfigServe
  - BMeasureApi::BMeasure, 40
- getConfig
  - BMeasureApi::BMeasure, 40
- BMeasureApi::BMeasureUnits, 66
- getConfigServe
  - BMeasureApi::BMeasure, 40
- getDigital
  - BMeasureApi::BMeasure, 41
- getDigitalServe
  - BMeasureApi::BMeasure, 41
- getFileName
  - BMeasureApi::DataFile, 103
- getFreeBlock
  - BMeasureApi::BMeasureUnits, 66
- getInfoBlock
  - BMeasureApi::BMeasure, 41
  - BMeasureApi::BMeasureUnits, 66
- getInfoBlockServe
  - BMeasureApi::BMeasure, 41
- getInformation
  - BMeasureApi::BMeasure, 41
  - BMeasureApi::BMeasureUnits, 66
- getInformationServe
  - BMeasureApi::BMeasure, 41
- getMeasurement
  - BMeasureApi::BMeasure, 42
  - BMeasureApi::BMeasureUnits, 66
- getMeasurementConfig
  - BMeasureApi::BMeasure, 42
  - BMeasureApi::BMeasureUnits, 66
- getMeasurementConfigServe
  - BMeasureApi::BMeasure, 42
- getMeasurementServe
  - BMeasureApi::BMeasure, 42
- getMembers
  - BMeasureApi::AwgConfig, 27
  - BMeasureApi::BoardConfig, 76
  - BMeasureApi::CalibrateInfo, 79
  - BMeasureApi::ChannelConfig, 81
  - BMeasureApi::ConfigItem, 93
  - BMeasureApi::Configuration, 95
  - BMeasureApi::DataBlock, 100
  - BMeasureApi::FileData, 111
  - BMeasureApi::FileInfo, 112
  - BMeasureApi::FilesysInfo, 113
  - BMeasureApi::InfoBlock, 115
  - BMeasureApi::Information, 117
  - BMeasureApi::MeasurementConfig, 120
  - BMeasureApi::NodeInfo, 123
  - BMeasureApi::NodeStatus, 124
  - BMeasureApi::Version, 125
- getNodeInfo
  - BMeasureApi::BMeasure, 42
- getNodeInfoServe
  - BMeasureApi::BMeasure, 42
- getStatus
  - BMeasureApi::BMeasure, 43
  - BMeasureApi::BMeasureUnits, 67
  - Dfu, 108
- getStatusServe
  - BMeasureApi::BMeasure, 43

- getSwitch
  - BMeasureApi::BMeasure, 43
- getSwitchServe
  - BMeasureApi::BMeasure, 43
- hardwareVersion
  - BMeasureApi::BoardConfig, 78
  - BMeasureApi::NodeInfo, 123
- hostname
  - BMdnsService, 32
- iString
  - DfuStatus, 110
- id
  - BMeasureApi::ChannelConfig, 82
- info
  - BMeasureApi::BMeasureUnit, 54
- init
  - BMdns, 31
  - BMeasureApi::BMeasureUnitsDataBlock, 75
  - BMeasureApi::CommsNet, 85
  - BMeasureApi::DataFile, 103
  - Dfu, 108
- length
  - BFirmwareInfo, 29
  - BMeasureApi::FileData, 111
- location
  - BMeasureApi::Configuration, 95
  - BMeasureApi::InfoBlock, 115
- logData
  - BMeasureApi::Configuration, 96
- logDataDevice
  - BMeasureApi::Configuration, 96
- LogDataMode
  - BMeasureApi, 20
- logDataMode
  - BMeasureApi::Configuration, 96
- login
  - BMeasureApi::BMeasure, 43
- loginServe
  - BMeasureApi::BMeasure, 43
- macAddress
  - BMeasureApi::BoardConfig, 78
- magic
  - BFirmwareInfo, 29
  - BMeasureApi::BoardConfig, 78
- mdns\_read\_string
  - BMdns.cpp, 129
- mdns\_read\_strings
  - BMdns.cpp, 129
- mdns\_write\_string
  - BMdns.cpp, 129
- MdnsClass
  - BMdns.cpp, 128
- MdnsEntryType
  - BMdns.cpp, 128
- MdnsRecordType
  - BMdns.cpp, 128
- measure
  - BMeasureApi::BMeasure, 44
- measureConfig
  - BMeasureApi::InfoBlock, 115
- MeasureMode
  - BMeasureApi, 20
- measureMode
  - BMeasureApi::MeasurementConfig, 120
- measurePeriod
  - BMeasureApi::MeasurementConfig, 121
- measureServe
  - BMeasureApi::BMeasure, 44
- MessageSource
  - BMeasureApi, 20
- Mode
  - BMeasureApi, 21
- mode
  - BMeasureApi::Configuration, 96
  - BMeasureApi::NodeStatus, 124
- name
  - BMdnsService, 32
  - BMeasureApi::ChannelConfig, 82
  - BMeasureApi::ConfigItem, 93
  - BMeasureApi::Configuration, 96
  - BMeasureApi::FileInfo, 112
  - BMeasureApi::FilesysInfo, 114
  - BMeasureApi::InfoBlock, 115
- networkAddress
  - BMeasureApi::Configuration, 96
  - BMeasureApi::Information, 118
- networkGateway
  - BMeasureApi::Configuration, 97
  - BMeasureApi::Information, 118
- networkMask
  - BMeasureApi::Configuration, 97
  - BMeasureApi::Information, 118
- NetworkMode
  - BMeasureApi, 21
- networkMode
  - BMeasureApi::Configuration, 97
  - BMeasureApi::Information, 118
- networkTimeServer
  - BMeasureApi::Configuration, 97
  - BMeasureApi::Information, 118
- nodeInfo
  - BMeasureApi::InfoBlock, 116
  - BMeasureApi::Information, 118
- NodeType
  - BMeasureApi, 21
- numChannels
  - BMeasureApi::BMeasureUnit, 54
  - BMeasureApi::BMeasureUnits, 67
  - BMeasureApi::DataBlock, 100
  - BMeasureApi::InfoBlock, 116
  - BMeasureApi::Information, 119
- numConfigItems
  - BMeasureApi::Information, 119

- numSamples
  - BMeasureApi::DataBlock, [101](#)
- numSamples0
  - BMeasureApi::MeasurementConfig, [121](#)
- numSamples1
  - BMeasureApi::MeasurementConfig, [121](#)
- numSamplesBlock
  - BMeasureApi::MeasurementConfig, [121](#)
- number
  - BMeasureApi::ChannelConfig, [82](#)
- oblockCount
  - BMeasureApi::BMeasureUnit, [55](#)
- obuffer
  - BMeasureApi::CommsUsb, [91](#)
- ochannels
  - BMeasureApi::BMeasureUnit, [56](#)
- oconfigMeasurement
  - BMeasureApi::BMeasureUnit, [56](#)
- oconnected
  - BMeasureApi::BMeasureUnit1, [59](#)
  - Dfu, [109](#)
- ocontext
  - BMeasureApi::CommsUsb, [92](#)
  - Dfu, [109](#)
- odataBlock
  - BMeasureApi::BMeasureUnit, [56](#)
  - BMeasureApi::BMeasureUnitsDataBlock, [75](#)
- odataBlocksFree
  - BMeasureApi::BMeasureUnits, [71](#)
- odataBlocksIn
  - BMeasureApi::BMeasureUnits, [71](#)
- odataBlocksOut
  - BMeasureApi::BMeasureUnits, [72](#)
- odataBlocksOutCount
  - BMeasureApi::BMeasureUnits, [72](#)
- odataBlocksProcess
  - BMeasureApi::BMeasureUnits, [72](#)
- odataBlocksProcessNum
  - BMeasureApi::BMeasureUnits, [72](#)
- odataStreamNum
  - BMeasureApi::BMeasureUnits, [72](#)
- odev
  - BMeasureApi::CommsUsb, [92](#)
  - Dfu, [109](#)
- odevice
  - BMeasureApi::BMeasureUnit, [56](#)
  - BMeasureApi::CommsSerial, [89](#)
  - BMeasureApi::CommsUsb, [92](#)
- odisconnecting
  - BMeasureApi::BMeasureUnit, [56](#)
- oenabled
  - BMeasureApi::BMeasureUnit1, [59](#)
- offset
  - BMeasureApi::AwgConfig, [28](#)
  - BMeasureApi::ChannelConfig, [82](#)
- ofile
  - BMeasureApi::DataFile, [105](#)
- ofilename
  - BMeasureApi::DataFile, [105](#)
- ofill
  - BMeasureApi::BMeasureUnits, [72](#)
  - BMeasureApi::BMeasureUnitsDataBlock, [75](#)
- oformat
  - BMeasureApi::DataFile, [105](#)
- oinUse
  - BMeasureApi::BMeasureUnitsDataBlock, [76](#)
- oinfo
  - BMeasureApi::BMeasureUnit, [56](#)
- olocalTrigger
  - BMeasureApi::BMeasureUnits, [72](#)
- olockInput
  - BMeasureApi::BMeasureUnits, [72](#)
- olockOutput
  - BMeasureApi::BMeasureUnits, [73](#)
- olockUnits
  - BMeasureApi::BMeasureUnits, [73](#)
- omeasureUnits
  - BMeasureApi::BMeasureUnit1, [59](#)
- omode
  - BMeasureApi::DataFile, [106](#)
- onodeInfo
  - BMeasureApi::BMeasureUnit, [56](#)
- onum
  - BMeasureApi::CommsUsb, [92](#)
- onumBlocks
  - BMeasureApi::BMeasureUnits, [73](#)
- onumChannels
  - BMeasureApi::BMeasureUnits, [73](#)
- onumConnected
  - BMeasureApi::BMeasureUnits, [73](#)
- oorder
  - BMeasureApi::BMeasureUnit1, [59](#)
- opacket
  - BMeasureApi::DataFile, [106](#)
- opacketLen
  - BMeasureApi::DataFile, [106](#)
- open
  - BMeasureApi::DataFile, [104](#)
- oprocEnable
  - BMeasureApi::BMeasureUnits, [73](#)
- oprocRunning
  - BMeasureApi::BMeasureUnits, [73](#)
- osampleCount
  - BMeasureApi::BMeasureUnit, [57](#)
- osequenceNext
  - BMeasureApi::BMeasureUnit, [57](#)
- oserialNumber
  - BMeasureApi::BMeasureUnit1, [60](#)
- oserialPort
  - BMeasureApi::CommsSerial, [89](#)
- osocket
  - Bmdns, [31](#)
  - BMeasureApi::CommsNet, [86](#)
- osource
  - BMeasureApi::BMeasureUnit1, [60](#)
- ostartSample

- BMeasureApi::BMeasureUnits, 74
- oterminated
  - BMeasureApi::CommsUsb, 92
- otransactionId
  - BMdns, 31
- otriggered
  - BMeasureApi::BMeasureUnits, 74
- ounitMaster
  - BMeasureApi::BMeasureUnits, 74
- ounits
  - BMeasureApi::BMeasureUnits, 74
- output
  - BMeasureApi::AwgConfig, 28
- outputBlock
  - BMeasureApi::BMeasureUnits, 67
- overbose
  - Dfu, 109
- overview.dox, 149
- pageAddress
  - Dfu.cpp, 149
- pageNumber
  - Dfu.cpp, 149
- pgaGain
  - BMeasureApi::ChannelConfig, 83
- pollTimeout
  - DfuStatus, 110
- process
  - BMeasureApi::ChannelConfig, 83
- processRequest
  - BMeasureApi::BMeasure, 44
- processdataBlock
  - BMeasureApi::BMeasureUnit, 54
- program
  - BMeasureApi::Configuration, 97
- read
  - BMeasureApi::CommsNet, 85
  - BMeasureApi::CommsSerial, 88
  - BMeasureApi::CommsUsb, 90
- readAvailable
  - BMeasureApi::CommsNet, 85
  - BMeasureApi::CommsSerial, 88
  - BMeasureApi::CommsUsb, 91
- readChunk
  - BMeasureApi::CommsUsb, 91
- readData
  - BMeasureApi::DataFile, 104
- readInfo
  - BMeasureApi::DataFile, 104
- reset
  - Dfu, 108
- round512
  - BMeasureApi, 24
- rs485BaudRate
  - BMeasureApi::Configuration, 97
- rs485Bits
  - BMeasureApi::Configuration, 98
- rs485StopBits
  - BMeasureApi::Configuration, 98
- run
  - BMeasureApi::BMeasureUnit, 54
  - BMeasureApi::BMeasureUnits, 67
- runBoardTest
  - BMeasureApi::BMeasure, 44
- runBoardTestServe
  - BMeasureApi::BMeasure, 44
- STATE\_APP\_DETACH
  - Dfu.cpp, 147
- STATE\_APP\_IDLE
  - Dfu.cpp, 147
- STATE\_DFU\_DOWNLOAD\_BUSY
  - Dfu.cpp, 147
- STATE\_DFU\_DOWNLOAD\_IDLE
  - Dfu.cpp, 147
- STATE\_DFU\_DOWNLOAD\_SYNC
  - Dfu.cpp, 147
- STATE\_DFU\_ERROR
  - Dfu.cpp, 148
- STATE\_DFU\_IDLE
  - Dfu.cpp, 148
- STATE\_DFU\_MANIFEST\_SYNC
  - Dfu.cpp, 148
- STATE\_DFU\_MANIFEST\_WAIT\_RESET
  - Dfu.cpp, 148
- STATE\_DFU\_MANIFEST
  - Dfu.cpp, 148
- STATE\_DFU\_UPLOAD\_IDLE
  - Dfu.cpp, 148
- sampleFrequencyMode
  - BMeasureApi::Configuration, 98
- sampleRate
  - BMeasureApi::MeasurementConfig, 121
- SampleType
  - BMeasureApi, 21
- sampleType
  - BMeasureApi::ChannelConfig, 83
- sampleTypeString
  - BMeasureApi, 25
- scale
  - BMeasureApi::ChannelConfig, 83
- SecureMode
  - BMeasureApi, 22
- sendData
  - BMeasureApi::BMeasure, 44
- sendDataEnable
  - BMeasureApi::BMeasure, 45
  - BMeasureApi::BMeasureUnits, 67
- sendDataEnableServe
  - BMeasureApi::BMeasure, 45
- sendDataProcess
  - BMeasureApi::BMeasureUnits, 67
- sendDataProcessTrigger
  - BMeasureApi::BMeasureUnits, 68
- sendDataQueue
  - BMeasureApi::BMeasureUnits, 68
- sendDataServe

- BMeasureApi::BMeasure, 45
- BMeasureApi::BMeasureUnit, 54
- sendDataServe
  - BMeasureApi::BMeasureUnit, 54
  - BMeasureApi::BMeasureUnit1, 58
  - BMeasureApi::BMeasureUnits, 68
- sendInfo
  - BMeasureApi::BMeasure, 45
- sendInfoServe
  - BMeasureApi::BMeasure, 45
- sendMessage
  - BMeasureApi::BMeasure, 45
  - BMeasureApi::BMeasureUnits, 68
- sendMessageServe
  - BMeasureApi::BMeasure, 46
  - BMeasureApi::BMeasureUnit1, 58
  - BMeasureApi::BMeasureUnits, 68
- sendStatus
  - BMeasureApi::BMeasure, 46
- sendStatusServe
  - BMeasureApi::BMeasure, 46
- sendTime
  - BMeasureApi::BMeasure, 46
  - BMeasureApi::BMeasureUnits, 68
- sendTimeServe
  - BMeasureApi::BMeasure, 46
- sequence
  - BMeasureApi::DataBlock, 101
- serialNumber
  - BMeasureApi::BMeasureUnit, 55
  - BMeasureApi::BMeasureUnit1, 59
  - BMeasureApi::BMeasureUnitDevice, 61
  - BMeasureApi::BoardConfig, 78
  - BMeasureApi::NodeInfo, 123
- setAnalogueOut
  - BMeasureApi::BMeasure, 46
- setAnalogueOutServe
  - BMeasureApi::BMeasure, 47
- setAwgConfig
  - BMeasureApi::BMeasure, 47
  - BMeasureApi::BMeasureUnits, 68
- setAwgConfigServe
  - BMeasureApi::BMeasure, 47
- setAwgWaveform
  - BMeasureApi::BMeasure, 47
- setAwgWaveformServe
  - BMeasureApi::BMeasure, 47
- setBoardConfig
  - BMeasureApi::BMeasure, 47
- setBoardConfigServe
  - BMeasureApi::BMeasure, 48
- setChannelConfig
  - BMeasureApi::BMeasure, 48
  - BMeasureApi::BMeasureUnit, 55
  - BMeasureApi::BMeasureUnits, 69
- setChannelConfigFull
  - BMeasureApi::BMeasure, 48
- setChannelConfigFullServe
  - BMeasureApi::BMeasure, 48
- setChannelConfigServe
  - BMeasureApi::BMeasure, 48
- setConfig
  - BMeasureApi::BMeasure, 48
  - BMeasureApi::BMeasureUnits, 69
- setConfigServe
  - BMeasureApi::BMeasure, 49
- setDigital
  - BMeasureApi::BMeasure, 49
- setDigitalServe
  - BMeasureApi::BMeasure, 49
- setMeasurement
  - BMeasureApi::BMeasure, 49
  - BMeasureApi::BMeasureUnit, 55
  - BMeasureApi::BMeasureUnits, 69
- setMeasurementConfig
  - BMeasureApi::BMeasure, 49
  - BMeasureApi::BMeasureUnits, 69
- setMeasurementConfigServe
  - BMeasureApi::BMeasure, 49
- setMeasurementServe
  - BMeasureApi::BMeasure, 50
- setMode
  - BMeasureApi::BMeasure, 50
  - BMeasureApi::BMeasureUnits, 69
- setModeServe
  - BMeasureApi::BMeasure, 50
- setRelay
  - BMeasureApi::BMeasure, 50
- setRelayServe
  - BMeasureApi::BMeasure, 50
- setSecureMode
  - BMeasureApi::BMeasure, 50
- setSecureModeServe
  - BMeasureApi::BMeasure, 51
- setSerialNumber
  - BMeasureApi::BMeasureUnit1, 59
- siUnits
  - BMeasureApi::ChannelConfig, 83
- size
  - BMeasureApi::FilesysInfo, 114
- softwareVersion
  - BMeasureApi::NodeInfo, 123
- source
  - BMeasureApi::Configuration, 98
  - BMeasureApi::DataBlock, 101
  - BMeasureApi::InfoBlock, 116
- spare
  - BMeasureApi::AwgConfig, 28
  - BMeasureApi::ConfigItem, 93
  - BMeasureApi::DataBlock, 101
  - BMeasureApi::FileInfo, 113
  - BMeasureApi::NodeStatus, 125
- spare0
  - BMeasureApi::BoardConfig, 78
  - BMeasureApi::ChannelConfig, 83
  - BMeasureApi::InfoBlock, 116

- BMeasureApi::Information, 119
- spare1
  - BMeasureApi::Configuration, 98
  - BMeasureApi::Information, 119
- spare3
  - BMeasureApi::Configuration, 98
- spare4
  - BMeasureApi::Configuration, 99
- stage
  - BMeasureApi::CalibrateInfo, 79
- state
  - DfuStatus, 110
- Status
  - BMeasureApi, 22
- status
  - BMeasureApi::DataBlock, 101
  - BMeasureApi::NodeStatus, 125
  - DfuStatus, 110
- SyncMode
  - BMeasureApi, 22
- TdsDataType
  - BMeasureApi, 23
- testMode
  - BMeasureApi::BoardConfig, 78
- time
  - BMeasureApi::DataBlock, 101
  - BMeasureApi::FileInfo, 113
  - BMeasureApi::InfoBlock, 116
  - BMeasureApi::Information, 119
  - BMeasureApi::NodeStatus, 125
- toFloat
  - BMeasureApi, 26
- TocBigEndian
  - BMeasureApi, 25
- TocDaqRawData
  - BMeasureApi, 25
- TocInterleavedData
  - BMeasureApi, 25
- TocMetaData
  - BMeasureApi, 25
- TocNewObjList
  - BMeasureApi, 25
- TocRawData
  - BMeasureApi, 25
- triggerChannel
  - BMeasureApi::MeasurementConfig, 121
- TriggerConfig
  - BMeasureApi, 23
- triggerConfig
  - BMeasureApi::MeasurementConfig, 121
- triggerDelay
  - BMeasureApi::MeasurementConfig, 122
- triggerLevel
  - BMeasureApi::MeasurementConfig, 122
- TriggerMode
  - BMeasureApi, 24
- triggerMode
  - BMeasureApi::MeasurementConfig, 122
- type
  - BFirmwareInfo, 29
  - BMeasureApi::ChannelConfig, 83
  - BMeasureApi::ConfigItem, 93
  - BMeasureApi::DataBlock, 102
  - BMeasureApi::Version, 126
- unit
  - BMeasureApi::BMeasureUnits, 69
- unitAdd
  - BMeasureApi::BMeasureUnits, 70
- unitDelete
  - BMeasureApi::BMeasureUnits, 70
- unitMaster
  - BMeasureApi::BMeasureUnits, 70
- unitSetEnabled
  - BMeasureApi::BMeasureUnits, 71
- unitSetOrder
  - BMeasureApi::BMeasureUnits, 71
- unitSort
  - BMeasureApi, 26
- unitsConnect
  - BMeasureApi::BMeasureUnits, 70
- unitsConnected
  - BMeasureApi::BMeasureUnits, 70
- unitsConnectedNum
  - BMeasureApi::BMeasureUnits, 70
- unitsDisconnect
  - BMeasureApi::BMeasureUnits, 70
- unitsFind
  - BMeasureApi::BMeasureUnits, 71
- unitsNum
  - BMeasureApi::BMeasureUnits, 71
- upload
  - Dfu, 108
- upload\_cmd
  - Dfu, 108
- usbaEnable
  - BMeasureApi::Configuration, 99
- usbbEnable
  - BMeasureApi::Configuration, 99
- validateFile
  - Dfu, 109
- validateFormat
  - BMeasureApi::DataFile, 104
- value
  - BMeasureApi::CalibrateInfo, 80
  - BMeasureApi::ConfigItem, 93
- ver0
  - BFirmwareInfo, 30
  - BMeasureApi::Version, 126
- ver1
  - BFirmwareInfo, 30
  - BMeasureApi::Version, 126
- ver2
  - BFirmwareInfo, 30
  - BMeasureApi::Version, 126
- version



- BMeasureApi::Configuration, [99](#)
- BMeasureApi::InfoBlock, [116](#)
- wait
  - BMeasureApi::CommsNet, [86](#)
  - BMeasureApi::CommsSerial, [88](#)
  - BMeasureApi::CommsUsb, [91](#)
- Waveform
  - BMeasureApi, [24](#)
- waveform
  - BMeasureApi::AwgConfig, [28](#)
- wifiEnable
  - BMeasureApi::Configuration, [99](#)
- write
  - BMeasureApi::CommsNet, [86](#)
  - BMeasureApi::CommsSerial, [88](#)
  - BMeasureApi::CommsUsb, [91](#)
- writeAvailable
  - BMeasureApi::CommsNet, [86](#)
- writeChunks
  - BMeasureApi::CommsNet, [86](#)
- writeData
  - BMeasureApi::DataFile, [104](#)
- writeEnd
  - BMeasureApi::DataFile, [104](#)
- writeInfo
  - BMeasureApi::DataFile, [105](#)
- writeInfoBMeas
  - BMeasureApi::DataFile, [105](#)
- writeInfoTdms
  - BMeasureApi::DataFile, [105](#)