

BMeasure-lib

0.8.4

Generated by Doxygen 1.8.15



---

<b>1 Main Page</b>	<b>1</b>
1.1 Introduction	1
1.2 Overview	2
1.3 API Usage	2
1.4 API Usage	3
<b>2 Namespace Index</b>	<b>7</b>
2.1 Namespace List	7
<b>3 Hierarchical Index</b>	<b>9</b>
3.1 Class Hierarchy	9
<b>4 Class Index</b>	<b>11</b>
4.1 Class List	11
<b>5 File Index</b>	<b>13</b>
5.1 File List	13
<b>6 Namespace Documentation</b>	<b>15</b>
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 AlarmMode	17
6.1.2.2 AlarmOutput	18
6.1.2.3 AwgOutput	18
6.1.2.4 BlockTypes	18
6.1.2.5 CalibrateStage	19
6.1.2.6 ChannelType	19
6.1.2.7 DataBlockType	19
6.1.2.8 DataSend	19
6.1.2.9 DigitalMode	20
6.1.2.10 ErrorNum	20
6.1.2.11 EventMode	20
6.1.2.12 FilesysDeleteType	21
6.1.2.13 FileType	21
6.1.2.14 LogDataMode	21
6.1.2.15 MeasureMode	21
6.1.2.16 MessageSource	22
6.1.2.17 Mode	22
6.1.2.18 NetworkMode	22
6.1.2.19 NodeType	22
6.1.2.20 Rs485Mode	23
6.1.2.21 SampleType	23

---

6.1.2.22 SecureMode . . . . .	23
6.1.2.23 Status . . . . .	24
6.1.2.24 SyncMode . . . . .	24
6.1.2.25 TdsDataType . . . . .	24
6.1.2.26 TriggerConfig . . . . .	25
6.1.2.27 TriggerMode . . . . .	25
6.1.2.28 Waveform . . . . .	25
6.1.2.29 WifiMode . . . . .	26
6.1.3 Function Documentation . . . . .	26
6.1.3.1 channelTypeString() . . . . .	26
6.1.3.2 round512() . . . . .	26
6.1.3.3 roundDown512() . . . . .	26
6.1.3.4 sampleTypeString() . . . . .	27
6.1.3.5 TocBigEndian() . . . . .	27
6.1.3.6 TocDaqRawData() . . . . .	27
6.1.3.7 TocInterleavedData() . . . . .	27
6.1.3.8 TocMetaData() . . . . .	27
6.1.3.9 TocNewObjList() . . . . .	27
6.1.3.10 TocRawData() . . . . .	27
6.1.3.11 toFloat() . . . . .	28
6.1.3.12 unitSort() . . . . .	28
6.1.4 Variable Documentation . . . . .	28
6.1.4.1 apiVersion . . . . .	28
<b>7 Class Documentation</b> . . . . .	<b>29</b>
7.1 BMeasureApi::AlarmConfig Class Reference . . . . .	29
7.1.1 Member Function Documentation . . . . .	29
7.1.1.1 getMembers() . . . . .	29
7.1.2 Member Data Documentation . . . . .	30
7.1.2.1 channel . . . . .	30
7.1.2.2 level . . . . .	30
7.1.2.3 mode . . . . .	30
7.1.2.4 output . . . . .	30
7.1.2.5 outputChannel . . . . .	30
7.2 BMeasureApi::AwgConfig Class Reference . . . . .	31
7.2.1 Member Function Documentation . . . . .	31
7.2.1.1 getMembers() . . . . .	31
7.2.2 Member Data Documentation . . . . .	31
7.2.2.1 amplitude . . . . .	31
7.2.2.2 duty . . . . .	32
7.2.2.3 frequency . . . . .	32
7.2.2.4 numSamples . . . . .	32

---

7.2.2.5 offset	32
7.2.2.6 output	32
7.2.2.7 spare	32
7.2.2.8 waveform	33
7.3 BFirmwareInfo Struct Reference	33
7.3.1 Member Data Documentation	33
7.3.1.1 checksum	33
7.3.1.2 length	33
7.3.1.3 magic	33
7.3.1.4 type	34
7.3.1.5 ver0	34
7.3.1.6 ver1	34
7.3.1.7 ver2	34
7.4 BMdns Class Reference	34
7.4.1 Constructor & Destructor Documentation	34
7.4.1.1 BMdns()	35
7.4.1.2 ~BMdns()	35
7.4.2 Member Function Documentation	35
7.4.2.1 findServices()	35
7.4.2.2 init()	35
7.4.3 Member Data Documentation	35
7.4.3.1 osocket	35
7.4.3.2 otransactionId	35
7.5 BMdnsService Class Reference	36
7.5.1 Member Data Documentation	36
7.5.1.1 address	36
7.5.1.2 extra	36
7.5.1.3 hostname	36
7.5.1.4 name	36
7.6 BMeasureApi::BMeasure Class Reference	37
7.6.1 Constructor & Destructor Documentation	39
7.6.1.1 BMeasure()	40
7.6.2 Member Function Documentation	40
7.6.2.1 calibrate()	40
7.6.2.2 calibrateServe()	40
7.6.2.3 factoryReset()	40
7.6.2.4 factoryResetServe()	40
7.6.2.5 fileClose()	40
7.6.2.6 fileCloseServe()	41
7.6.2.7 fileDelete()	41
7.6.2.8 fileDeleteServe()	41
7.6.2.9 fileList()	41

---

7.6.2.10	fileListServe()	41
7.6.2.11	fileOpen()	41
7.6.2.12	fileOpenServe()	42
7.6.2.13	fileRead()	42
7.6.2.14	fileReadServe()	42
7.6.2.15	fileSysDelete()	42
7.6.2.16	fileSysDeleteServe()	42
7.6.2.17	fileSysInfo()	42
7.6.2.18	fileSysInfoServe()	43
7.6.2.19	fileWrite()	43
7.6.2.20	fileWriteServe()	43
7.6.2.21	functionUnLock()	43
7.6.2.22	functionUnLockServe()	43
7.6.2.23	getAwgConfig()	43
7.6.2.24	getAwgConfigServe()	44
7.6.2.25	getBoardConfig()	44
7.6.2.26	getBoardConfigServe()	44
7.6.2.27	getChannelConfig()	44
7.6.2.28	getChannelConfigServe()	44
7.6.2.29	getConfig()	44
7.6.2.30	getConfigServe()	45
7.6.2.31	getDigital()	45
7.6.2.32	getDigitalServe()	45
7.6.2.33	getInfoBlock()	45
7.6.2.34	getInfoBlockServe()	45
7.6.2.35	getInformation()	45
7.6.2.36	getInformationServe()	46
7.6.2.37	getMeasurement()	46
7.6.2.38	getMeasurementConfig()	46
7.6.2.39	getMeasurementConfigServe()	46
7.6.2.40	getMeasurementServe()	46
7.6.2.41	getNodeInfo()	46
7.6.2.42	getNodeInfoServe()	47
7.6.2.43	getStatus()	47
7.6.2.44	getStatusServe()	47
7.6.2.45	getSwitch()	47
7.6.2.46	getSwitchServe()	47
7.6.2.47	login()	47
7.6.2.48	loginServe()	48
7.6.2.49	measure()	48
7.6.2.50	measureServe()	48
7.6.2.51	processRequest()	48

---

7.6.2.52 runBoardTest()	48
7.6.2.53 runBoardTestServe()	48
7.6.2.54 sendData()	49
7.6.2.55 sendDataEnable()	49
7.6.2.56 sendDataEnableServe()	49
7.6.2.57 sendDataServe()	49
7.6.2.58 sendInfo()	49
7.6.2.59 sendInfoServe()	49
7.6.2.60 sendMessage()	50
7.6.2.61 sendMessageServe()	50
7.6.2.62 sendStatus()	50
7.6.2.63 sendStatusServe()	50
7.6.2.64 sendTime()	50
7.6.2.65 sendTimeServe()	50
7.6.2.66 setAnalogueOut()	51
7.6.2.67 setAnalogueOutServe()	51
7.6.2.68 setAwgConfig()	51
7.6.2.69 setAwgConfigServe()	51
7.6.2.70 setAwgWaveform()	51
7.6.2.71 setAwgWaveformServe()	51
7.6.2.72 setBoardConfig()	52
7.6.2.73 setBoardConfigServe()	52
7.6.2.74 setChannelConfig()	52
7.6.2.75 setChannelConfigFull()	52
7.6.2.76 setChannelConfigFullServe()	52
7.6.2.77 setChannelConfigServe()	52
7.6.2.78 setConfig()	53
7.6.2.79 setConfigServe()	53
7.6.2.80 setDigital()	53
7.6.2.81 setDigitalServe()	53
7.6.2.82 setMeasurement()	53
7.6.2.83 setMeasurementConfig()	53
7.6.2.84 setMeasurementConfigServe()	54
7.6.2.85 setMeasurementServe()	54
7.6.2.86 setMode()	54
7.6.2.87 setModeServe()	54
7.6.2.88 setRelay()	54
7.6.2.89 setRelayServe()	54
7.6.2.90 setSecureKey()	55
7.6.2.91 setSecureKeyServe()	55
7.6.2.92 setSecureMode()	55
7.6.2.93 setSecureModeServe()	55

---

7.7 BMeasureApi::BMeasureUnit Class Reference	55
7.7.1 Constructor & Destructor Documentation	57
7.7.1.1 BMeasureUnit()	57
7.7.1.2 ~BMeasureUnit()	57
7.7.2 Member Function Documentation	57
7.7.2.1 connect()	57
7.7.2.2 device()	57
7.7.2.3 disconnect()	57
7.7.2.4 disconnected()	58
7.7.2.5 findDevices()	58
7.7.2.6 findDevicesNetwork()	58
7.7.2.7 findDevicesUsb()	58
7.7.2.8 info()	58
7.7.2.9 numChannels()	58
7.7.2.10 processDataBlock()	59
7.7.2.11 run()	59
7.7.2.12 sendDataServe()	59
7.7.2.13 sendDataServe1()	59
7.7.2.14 serialNumber()	59
7.7.2.15 setChannelConfig()	59
7.7.2.16 setMeasurement()	60
7.7.3 Member Data Documentation	60
7.7.3.1 blockNumChannels	60
7.7.3.2 blockNumSamples	60
7.7.3.3 oblockCount	60
7.7.3.4 ochannels	60
7.7.3.5 oconfigMeasurement	60
7.7.3.6 odataBlock	60
7.7.3.7 odevice	61
7.7.3.8 odisconnecting	61
7.7.3.9 oinfo	61
7.7.3.10 onodeInfo	61
7.7.3.11 osampleCount	61
7.7.3.12 osequenceNext	61
7.8 BMeasureApi::BMeasureUnit1 Class Reference	62
7.8.1 Constructor & Destructor Documentation	62
7.8.1.1 BMeasureUnit1()	62
7.8.2 Member Function Documentation	63
7.8.2.1 disconnected()	63
7.8.2.2 sendDataServe1()	63
7.8.2.3 sendMessageServe()	63
7.8.2.4 serialNumber()	63

---



---

7.8.2.5 setSerialNumber()	63
7.8.3 Member Data Documentation	63
7.8.3.1 oconnected	64
7.8.3.2 oenabled	64
7.8.3.3 omeasureUnits	64
7.8.3.4 oorder	64
7.8.3.5 oserialNumber	64
7.8.3.6 osource	64
7.9 BMeasureApi::BMeasureUnitDevice Class Reference	64
7.9.1 Constructor & Destructor Documentation	65
7.9.1.1 BMeasureUnitDevice()	65
7.9.2 Member Data Documentation	65
7.9.2.1 device	65
7.9.2.2 serialNumber	65
7.10 BMeasureApi::BMeasureUnits Class Reference	65
7.10.1 Constructor & Destructor Documentation	67
7.10.1.1 BMeasureUnits()	68
7.10.1.2 ~BMeasureUnits()	68
7.10.2 Member Function Documentation	68
7.10.2.1 clear()	68
7.10.2.2 dataAvailable()	68
7.10.2.3 dataClear()	68
7.10.2.4 dataDone()	68
7.10.2.5 dataEvent()	68
7.10.2.6 dataProcessEnable()	69
7.10.2.7 dataRead()	69
7.10.2.8 dataSetNumStreams()	69
7.10.2.9 dataWait()	69
7.10.2.10 debugPrint()	69
7.10.2.11 disconnected()	69
7.10.2.12 getAwgConfig()	70
7.10.2.13 getChannelConfig()	70
7.10.2.14 getConfig()	70
7.10.2.15 getFreeBlock()	70
7.10.2.16 getInfoBlock()	70
7.10.2.17 getInformation()	70
7.10.2.18 getMeasurement()	71
7.10.2.19 getMeasurementConfig()	71
7.10.2.20 getStatus()	71
7.10.2.21 numChannels()	71
7.10.2.22 outputBlock()	71
7.10.2.23 run()	71

---

7.10.2.24 sendDataEnable()	72
7.10.2.25 sendDataProcess()	72
7.10.2.26 sendDataProcessTrigger()	72
7.10.2.27 sendDataQueue()	72
7.10.2.28 sendDataServe1()	72
7.10.2.29 sendMessage()	72
7.10.2.30 sendMessageServe()	72
7.10.2.31 sendTime()	73
7.10.2.32 setAwgConfig()	73
7.10.2.33 setChannelConfig()	73
7.10.2.34 setConfig()	73
7.10.2.35 setMeasurement()	73
7.10.2.36 setMeasurementConfig()	73
7.10.2.37 setMode()	74
7.10.2.38 unit()	74
7.10.2.39 unitAdd()	74
7.10.2.40 unitDelete()	74
7.10.2.41 unitMaster()	74
7.10.2.42 unitsConnect()	74
7.10.2.43 unitsConnected()	74
7.10.2.44 unitsConnectedNum()	75
7.10.2.45 unitsDisconnect()	75
7.10.2.46 unitSetEnabled()	75
7.10.2.47 unitSetOrder()	75
7.10.2.48 unitsFind()	75
7.10.2.49 unitsNum()	75
7.10.3 Member Data Documentation	75
7.10.3.1 odataBlocksFree	76
7.10.3.2 odataBlocksIn	76
7.10.3.3 odataBlocksOut	76
7.10.3.4 odataBlocksOutCount	76
7.10.3.5 odataBlocksProcess	76
7.10.3.6 odataBlocksProcessNum	76
7.10.3.7 odataStreamNum	76
7.10.3.8 ofill	76
7.10.3.9 olocalTrigger	77
7.10.3.10 olockInput	77
7.10.3.11 olockOutput	77
7.10.3.12 olockUnits	77
7.10.3.13 onumBlocks	77
7.10.3.14 onumChannels	77
7.10.3.15 onumConnected	77

---

7.10.3.16 oprocEnable	78
7.10.3.17 oprocRunning	78
7.10.3.18 ostartSample	78
7.10.3.19 otriggered	78
7.10.3.20 ounitMaster	78
7.10.3.21 ounits	78
7.11 BMeasureApi::BMeasureUnitsDataBlock Class Reference	78
7.11.1 Constructor & Destructor Documentation	79
7.11.1.1 BMeasureUnitsDataBlock()	79
7.11.1.2 ~BMeasureUnitsDataBlock()	79
7.11.2 Member Function Documentation	79
7.11.2.1 init()	79
7.11.3 Member Data Documentation	79
7.11.3.1 odataBlock	80
7.11.3.2 ofill	80
7.11.3.3 oinUse	80
7.12 BMeasureApi::BoardConfig Class Reference	80
7.12.1 Member Function Documentation	81
7.12.1.1 getMembers()	81
7.12.2 Member Data Documentation	81
7.12.2.1 buildTime	81
7.12.2.2 calibAdcOffsets	81
7.12.2.3 calibAdcScales	81
7.12.2.4 calibAttenScales	81
7.12.2.5 calibDacOffsets	81
7.12.2.6 calibDacScales	82
7.12.2.7 calibFiveVolts	82
7.12.2.8 calibTemp	82
7.12.2.9 calibTime	82
7.12.2.10 hardwareVersion	82
7.12.2.11 macAddress	82
7.12.2.12 magic	82
7.12.2.13 serialNumber	82
7.12.2.14 spare0	83
7.12.2.15 testMode	83
7.13 BMeasureApi::CalibrateInfo Class Reference	83
7.13.1 Member Function Documentation	83
7.13.1.1 getMembers()	83
7.13.2 Member Data Documentation	84
7.13.2.1 calibrateAmplitude	84
7.13.2.2 calibrateFrequency	84
7.13.2.3 calibrateTime	84

---

7.13.2.4 stage . . . . .	84
7.13.2.5 value . . . . .	84
7.14 BMeasureApi::ChannelConfig Class Reference . . . . .	85
7.14.1 Member Function Documentation . . . . .	85
7.14.1.1 getMembers() . . . . .	85
7.14.2 Member Data Documentation . . . . .	86
7.14.2.1 attenuator . . . . .	86
7.14.2.2 calibOffset . . . . .	86
7.14.2.3 calibScale . . . . .	86
7.14.2.4 calibScaleAtten1 . . . . .	86
7.14.2.5 dataChannel . . . . .	86
7.14.2.6 enabled . . . . .	87
7.14.2.7 id . . . . .	87
7.14.2.8 name . . . . .	87
7.14.2.9 number . . . . .	87
7.14.2.10 offset . . . . .	87
7.14.2.11 pgaGain . . . . .	87
7.14.2.12 process . . . . .	87
7.14.2.13 sampleType . . . . .	88
7.14.2.14 scale . . . . .	88
7.14.2.15 siUnits . . . . .	88
7.14.2.16 spare0 . . . . .	88
7.14.2.17 type . . . . .	88
7.15 BMeasureApi::CommsNet Class Reference . . . . .	88
7.15.1 Constructor & Destructor Documentation . . . . .	89
7.15.1.1 CommsNet() . . . . .	89
7.15.1.2 ~CommsNet() . . . . .	89
7.15.2 Member Function Documentation . . . . .	89
7.15.2.1 connect() . . . . .	89
7.15.2.2 disconnect() . . . . .	90
7.15.2.3 init() . . . . .	90
7.15.2.4 read() . . . . .	90
7.15.2.5 readAvailable() . . . . .	90
7.15.2.6 wait() . . . . .	90
7.15.2.7 write() . . . . .	90
7.15.2.8 writeAvailable() . . . . .	91
7.15.2.9 writeChunks() . . . . .	91
7.15.3 Member Data Documentation . . . . .	91
7.15.3.1 oinWait . . . . .	91
7.15.3.2 osocket . . . . .	91
7.15.3.3 oterminating . . . . .	91
7.16 BMeasureApi::CommsSerial Class Reference . . . . .	92

---

---

7.16.1 Constructor & Destructor Documentation	92
7.16.1.1 CommsSerial()	92
7.16.1.2 ~CommsSerial()	92
7.16.2 Member Function Documentation	93
7.16.2.1 connect()	93
7.16.2.2 disconnect()	93
7.16.2.3 read()	93
7.16.2.4 readAvailable()	93
7.16.2.5 wait()	93
7.16.2.6 write()	94
7.16.3 Member Data Documentation	94
7.16.3.1 odevice	94
7.16.3.2 oserialPort	94
7.17 BMeasureApi::CommsUsb Class Reference	94
7.17.1 Constructor & Destructor Documentation	95
7.17.1.1 CommsUsb()	95
7.17.1.2 ~CommsUsb()	95
7.17.2 Member Function Documentation	95
7.17.2.1 connect()	95
7.17.2.2 disconnect()	95
7.17.2.3 read()	96
7.17.2.4 readAvailable()	96
7.17.2.5 readChunk()	96
7.17.2.6 wait()	96
7.17.2.7 write()	96
7.17.3 Member Data Documentation	96
7.17.3.1 obuffer	97
7.17.3.2 ocontext	97
7.17.3.3 odev	97
7.17.3.4 odevice	97
7.17.3.5 onum	97
7.17.3.6 oterminated	97
7.17.3.7 oterminating	97
7.18 BMeasureApi::ConfigItem Class Reference	98
7.18.1 Member Function Documentation	98
7.18.1.1 getMembers()	98
7.18.2 Member Data Documentation	98
7.18.2.1 name	98
7.18.2.2 spare	98
7.18.2.3 type	99
7.18.2.4 value	99
7.19 BMeasureApi::Configuration Class Reference	99

---

7.19.1 Member Function Documentation	100
7.19.1.1 getMembers()	100
7.19.2 Member Data Documentation	101
7.19.2.1 alarms	101
7.19.2.2 digitalMode	101
7.19.2.3 digitalPins	101
7.19.2.4 emailAddress	101
7.19.2.5 emailMode	101
7.19.2.6 location	101
7.19.2.7 logData	102
7.19.2.8 logDataDevice	102
7.19.2.9 logDataMode	102
7.19.2.10 mode	102
7.19.2.11 mqttMode	102
7.19.2.12 mqttPort	102
7.19.2.13 mqttServer	103
7.19.2.14 name	103
7.19.2.15 networkAddress	103
7.19.2.16 networkGateway	103
7.19.2.17 networkMask	103
7.19.2.18 networkMode	103
7.19.2.19 networkNameServer0	104
7.19.2.20 networkTimeServer	104
7.19.2.21 program	104
7.19.2.22 rs485BaudRate	104
7.19.2.23 rs485Bits	104
7.19.2.24 rs485Mode	104
7.19.2.25 rs485StopBits	105
7.19.2.26 sampleFrequencyMode	105
7.19.2.27 source	105
7.19.2.28 spare0	105
7.19.2.29 spare1	105
7.19.2.30 spare2	105
7.19.2.31 spare3	105
7.19.2.32 spare4	106
7.19.2.33 spare5	106
7.19.2.34 version	106
7.19.2.35 wifiAp0	106
7.19.2.36 wifiAp0Password	106
7.19.2.37 wifiMode	106
7.20 BMeasureApi::DataBlock Class Reference	106
7.20.1 Member Function Documentation	107

---

7.20.1.1	getMembers()	107
7.20.2	Member Data Documentation	107
7.20.2.1	data	107
7.20.2.2	numChannels	107
7.20.2.3	numSamples	108
7.20.2.4	sequence	108
7.20.2.5	source	108
7.20.2.6	spare	108
7.20.2.7	status	108
7.20.2.8	time	108
7.20.2.9	type	109
7.21	BMeasureApi::DataFile Class Reference	109
7.21.1	Constructor & Destructor Documentation	110
7.21.1.1	DataFile()	110
7.21.1.2	~DataFile()	110
7.21.2	Member Function Documentation	110
7.21.2.1	close()	110
7.21.2.2	getFileName()	110
7.21.2.3	init()	110
7.21.2.4	open()	111
7.21.2.5	readData()	111
7.21.2.6	readInfo()	111
7.21.2.7	validateFormat()	111
7.21.2.8	writeData()	111
7.21.2.9	writeEnd()	111
7.21.2.10	writeInfo()	112
7.21.2.11	writeInfoBMeas()	112
7.21.2.12	writeInfoCsv()	112
7.21.2.13	writeInfoTdms()	112
7.21.3	Member Data Documentation	112
7.21.3.1	ofile	112
7.21.3.2	ofilename	112
7.21.3.3	offormat	113
7.21.3.4	omode	113
7.21.3.5	opacket	113
7.21.3.6	opacketLen	113
7.22	Dfu Class Reference	113
7.22.1	Detailed Description	114
7.22.2	Constructor & Destructor Documentation	114
7.22.2.1	Dfu()	114
7.22.2.2	~Dfu()	114
7.22.3	Member Function Documentation	114

---

7.22.3.1 clearStatus()	114
7.22.3.2 connect()	114
7.22.3.3 detectDevice()	115
7.22.3.4 disconnect()	115
7.22.3.5 getStatus()	115
7.22.3.6 init()	115
7.22.3.7 reset()	115
7.22.3.8 upload()	115
7.22.3.9 upload_cmd()	116
7.22.3.10 validateFile()	116
7.22.4 Member Data Documentation	116
7.22.4.1 oconnected	116
7.22.4.2 ocontext	116
7.22.4.3 odev	116
7.22.4.4 overbose	116
7.23 DfuStatus Struct Reference	117
7.23.1 Member Data Documentation	117
7.23.1.1 iString	117
7.23.1.2 pollTimeout	117
7.23.1.3 state	117
7.23.1.4 status	117
7.24 BMeasureApi::FileData Class Reference	117
7.24.1 Member Function Documentation	118
7.24.1.1 getMembers()	118
7.24.2 Member Data Documentation	118
7.24.2.1 data	118
7.24.2.2 length	118
7.25 BMeasureApi::FileInfo Class Reference	118
7.25.1 Member Function Documentation	119
7.25.1.1 getMembers()	119
7.25.2 Member Data Documentation	119
7.25.2.1 fileLength	119
7.25.2.2 fileType	119
7.25.2.3 name	120
7.25.2.4 spare	120
7.25.2.5 time	120
7.26 BMeasureApi::FilesysInfo Class Reference	120
7.26.1 Member Function Documentation	120
7.26.1.1 getMembers()	121
7.26.2 Member Data Documentation	121
7.26.2.1 free	121
7.26.2.2 name	121



---

7.26.2.3 size	121
7.27 BMeasureApi::InfoBlock Class Reference	121
7.27.1 Member Function Documentation	122
7.27.1.1 getMembers()	122
7.27.2 Member Data Documentation	122
7.27.2.1 location	122
7.27.2.2 measureConfig	122
7.27.2.3 name	123
7.27.2.4 nodeInfo	123
7.27.2.5 numChannels	123
7.27.2.6 source	123
7.27.2.7 spare0	123
7.27.2.8 time	123
7.27.2.9 version	124
7.28 BMeasureApi::Information Class Reference	124
7.28.1 Member Function Documentation	124
7.28.1.1 getMembers()	125
7.28.2 Member Data Documentation	125
7.28.2.1 networkAddress	125
7.28.2.2 networkGateway	125
7.28.2.3 networkMask	125
7.28.2.4 networkMode	125
7.28.2.5 networkNameServer0	125
7.28.2.6 networkTimeServer	126
7.28.2.7 nodeInfo	126
7.28.2.8 numChannels	126
7.28.2.9 numConfigItems	126
7.28.2.10 spare0	126
7.28.2.11 spare1	126
7.28.2.12 time	127
7.29 BMeasureApi::MeasurementConfig Class Reference	127
7.29.1 Member Function Documentation	127
7.29.1.1 getMembers()	127
7.29.2 Member Data Documentation	128
7.29.2.1 description	128
7.29.2.2 measureMode	128
7.29.2.3 measurePeriod	128
7.29.2.4 numSamples0	128
7.29.2.5 numSamples1	128
7.29.2.6 numSamplesBlock	128
7.29.2.7 sampleRate	129
7.29.2.8 triggerChannel	129

---

7.29.2.9 triggerConfig	129
7.29.2.10 triggerDelay	129
7.29.2.11 triggerLevel	129
7.29.2.12 triggerMode	129
7.30 BMeasureApi::NodeInfo Class Reference	129
7.30.1 Member Function Documentation	130
7.30.1.1 getMembers()	130
7.30.2 Member Data Documentation	130
7.30.2.1 apiVersion	130
7.30.2.2 fpgaVersion	130
7.30.2.3 hardwareVersion	130
7.30.2.4 serialNumber	131
7.30.2.5 softwareVersion	131
7.30.2.6 wifiVersion	131
7.31 BMeasureApi::NodeStatus Class Reference	131
7.31.1 Member Function Documentation	131
7.31.1.1 getMembers()	131
7.31.2 Member Data Documentation	132
7.31.2.1 error	132
7.31.2.2 errorStr	132
7.31.2.3 mode	132
7.31.2.4 spare	132
7.31.2.5 status	132
7.31.2.6 time	132
7.32 BMeasureApi::Version Class Reference	133
7.32.1 Member Function Documentation	133
7.32.1.1 getMembers()	133
7.32.2 Member Data Documentation	133
7.32.2.1 type	133
7.32.2.2 ver0	133
7.32.2.3 ver1	133
7.32.2.4 ver2	133
<b>8 File Documentation</b>	<b>135</b>
8.1 BMdns.cpp File Reference	135
8.1.1 Macro Definition Documentation	136
8.1.1.1 BDEBUGL1	136
8.1.2 Enumeration Type Documentation	136
8.1.2.1 MdnsClass	136
8.1.2.2 MdnsEntryType	136
8.1.2.3 MdnsRecordType	136
8.1.3 Function Documentation	137

---

---

8.1.3.1 mdns_read_string()	137
8.1.3.2 mdns_read_strings()	137
8.1.3.3 mdns_write_string()	137
8.2 BMdns.h File Reference	137
8.3 BMeasureB-1.cpp File Reference	137
8.4 BMeasureB.cpp File Reference	138
8.5 BMeasureB.h File Reference	138
8.6 BMeasureD.cpp File Reference	138
8.6.1 Macro Definition Documentation	139
8.6.1.1 boffsetof	139
8.7 BMeasureD.h File Reference	139
8.8 BMeasureLib.cpp File Reference	141
8.8.1 Macro Definition Documentation	141
8.8.1.1 BDEBUGL1	141
8.8.1.2 BDEBUGL2	142
8.9 BMeasureLib.h File Reference	142
8.10 BMeasureS.cpp File Reference	142
8.11 BMeasureUnit.cpp File Reference	142
8.11.1 Macro Definition Documentation	143
8.11.1.1 BDEBUGL1	143
8.11.1.2 BDEBUGL2	143
8.11.1.3 BDEBUGL3	143
8.11.1.4 CONVERT_FLOAT	143
8.12 BMeasureUnit.h File Reference	144
8.13 BMeasureUnits.cpp File Reference	144
8.13.1 Macro Definition Documentation	144
8.13.1.1 BDEBUGL1	145
8.13.1.2 BDEBUGL2	145
8.13.1.3 BDEBUGL3	145
8.14 BMeasureUnits.h File Reference	145
8.15 CommsNet.cpp File Reference	145
8.15.1 Macro Definition Documentation	146
8.15.1.1 BDEBUGL1	146
8.15.1.2 BDEBUGL2	146
8.15.1.3 BDEBUGL3	146
8.16 CommsNet.h File Reference	146
8.17 CommsSerial.cpp File Reference	147
8.18 CommsSerial.h File Reference	147
8.19 CommsUsb.cpp File Reference	147
8.19.1 Macro Definition Documentation	147
8.19.1.1 BDEBUGL1	148
8.19.1.2 BDEBUGL2	148

---

8.20 CommsUsb.h File Reference . . . . .	148
8.21 DataFile.cpp File Reference . . . . .	148
8.21.1 Macro Definition Documentation . . . . .	149
8.21.1.1 BDEBUGL1 . . . . .	149
8.21.1.2 BDEBUGL2 . . . . .	149
8.22 DataFile.h File Reference . . . . .	149
8.23 Dfu.cpp File Reference . . . . .	150
8.23.1 Macro Definition Documentation . . . . .	151
8.23.1.1 BDEBUGL1 . . . . .	151
8.23.1.2 BDEBUGL2 . . . . .	151
8.23.1.3 DFU_ABORT . . . . .	152
8.23.1.4 DFU_CLRSTATUS . . . . .	152
8.23.1.5 DFU_DETACH . . . . .	152
8.23.1.6 DFU_DNLOAD . . . . .	152
8.23.1.7 DFU_GETSTATE . . . . .	152
8.23.1.8 DFU_GETSTATUS . . . . .	152
8.23.1.9 DFU_IFF_ALT . . . . .	152
8.23.1.10 DFU_IFF_CONFIG . . . . .	152
8.23.1.11 DFU_IFF_DEVNUM . . . . .	153
8.23.1.12 DFU_IFF_DFU . . . . .	153
8.23.1.13 DFU_IFF_IFACE . . . . .	153
8.23.1.14 DFU_IFF_PATH . . . . .	153
8.23.1.15 DFU_IFF_PRODUCT . . . . .	153
8.23.1.16 DFU_IFF_VENDOR . . . . .	153
8.23.1.17 DFU_STATUS_ERROR_ADDRESS . . . . .	153
8.23.1.18 DFU_STATUS_ERROR_CHECK_ERASED . . . . .	153
8.23.1.19 DFU_STATUS_ERROR_ERASE . . . . .	154
8.23.1.20 DFU_STATUS_ERROR_FILE . . . . .	154
8.23.1.21 DFU_STATUS_ERROR_FIRMWARE . . . . .	154
8.23.1.22 DFU_STATUS_ERROR_NOTDONE . . . . .	154
8.23.1.23 DFU_STATUS_ERROR_POR . . . . .	154
8.23.1.24 DFU_STATUS_ERROR_PROG . . . . .	154
8.23.1.25 DFU_STATUS_ERROR_STALLEDPKT . . . . .	154
8.23.1.26 DFU_STATUS_ERROR_TARGET . . . . .	154
8.23.1.27 DFU_STATUS_ERROR_UNKNOWN . . . . .	155
8.23.1.28 DFU_STATUS_ERROR_USBR . . . . .	155
8.23.1.29 DFU_STATUS_ERROR_VENDOR . . . . .	155
8.23.1.30 DFU_STATUS_ERROR_VERIFY . . . . .	155
8.23.1.31 DFU_STATUS_ERROR_WRITE . . . . .	155
8.23.1.32 DFU_STATUS_OK . . . . .	155
8.23.1.33 DFU_UPLOAD . . . . .	155
8.23.1.34 STATE_APP_DETACH . . . . .	155

---

---

8.23.1.35 STATE_APP_IDLE . . . . .	156
8.23.1.36 STATE_DFU_DOWNLOAD_BUSY . . . . .	156
8.23.1.37 STATE_DFU_DOWNLOAD_IDLE . . . . .	156
8.23.1.38 STATE_DFU_DOWNLOAD_SYNC . . . . .	156
8.23.1.39 STATE_DFU_ERROR . . . . .	156
8.23.1.40 STATE_DFU_IDLE . . . . .	156
8.23.1.41 STATE_DFU_MANIFEST . . . . .	156
8.23.1.42 STATE_DFU_MANIFEST_SYNC . . . . .	156
8.23.1.43 STATE_DFU_MANIFEST_WAIT_RESET . . . . .	157
8.23.1.44 STATE_DFU_UPLOAD_IDLE . . . . .	157
8.23.2 Enumeration Type Documentation . . . . .	157
8.23.2.1 dfuse_command . . . . .	157
8.23.3 Function Documentation . . . . .	157
8.23.3.1 pageAddress() . . . . .	157
8.23.3.2 pageNumber() . . . . .	157
8.23.4 Variable Documentation . . . . .	157
8.23.4.1 BFirmwareInfoEncrypt1 . . . . .	158
8.23.4.2 BFirmwareInfoMagic . . . . .	158
8.24 Dfu.h File Reference . . . . .	158
8.25 overview.dox File Reference . . . . .	158
<b>Index</b>	<b>159</b>



# Chapter 1

## Main Page

### Author

Dr Terry Barnaby

### Version

0.8.4

### Date

2020-01-17

## 1.1 Introduction

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

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

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

## 1.2 Overview

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

The API provides the following functionality:

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

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

The API supports threaded and non-threaded operation.

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

## 1.3 API Usage

To use the API the core procedure is:

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

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



## 1.4 API Usage

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

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

```

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

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

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

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

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

    printf("NumChannels: %d\n", info.numChannels);
    //printf("Exit\n"); return err;
    printf("Configure measurement\n");
    mc.measureMode = MeasureModeOneShot;
    mc.triggerMode = TriggerModeOff;
    mc.triggerConfig = TriggerConfigNone;
    mc.triggerChannel = 0;
    mc.triggerLevel = 0;
    mc.triggerDelay = 0;
    mc.sampleRate = 8000.0;
    mc.measurePeriod = 0;
    mc.numSamples0 = 1;
    mc.numSamples1 = 0;
    if(err = bmeasure.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::AlarmConfig . . . . .	29
BMeasureApi::AwgConfig . . . . .	31
BComms [external]	
BMeasureApi::CommsNet . . . . .	88
BMeasureApi::CommsSerial . . . . .	92
BMeasureApi::CommsUsb . . . . .	94
BFirmwareInfo . . . . .	33
BMdns . . . . .	34
BMdnsService . . . . .	36
BMeasureApi::BMeasureUnitDevice . . . . .	64
BMeasureApi::BMeasureUnitsDataBlock . . . . .	78
BoapMc1Comms [external]	
BMeasureApi::BMeasure . . . . .	37
BMeasureApi::BMeasureUnit . . . . .	55
BMeasureApi::BMeasureUnit1 . . . . .	62
BMeasureApi::BoardConfig . . . . .	80
BTask [external]	
BMeasureApi::BMeasureUnit . . . . .	55
BMeasureApi::BMeasureUnits . . . . .	65
BMeasureApi::CalibrateInfo . . . . .	83
BMeasureApi::ChannelConfig . . . . .	85
BMeasureApi::ConfigItem . . . . .	98
BMeasureApi::Configuration . . . . .	99
BMeasureApi::DataBlock . . . . .	106
BMeasureApi::DataFile . . . . .	109
Dfu . . . . .	113
DfuStatus . . . . .	117
BMeasureApi::FileData . . . . .	117
BMeasureApi::FileInfo . . . . .	118
BMeasureApi::FilesysInfo . . . . .	120
BMeasureApi::InfoBlock . . . . .	121
BMeasureApi::Information . . . . .	124
BMeasureApi::MeasurementConfig . . . . .	127
BMeasureApi::NodeInfo . . . . .	129
BMeasureApi::NodeStatus . . . . .	131
BMeasureApi::Version . . . . .	133





# Chapter 4

## Class Index

### 4.1 Class List

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

- BMeasureApi::AlarmConfig . . . . . 29
- BMeasureApi::AwgConfig . . . . . 31
- BFirmwareInfo . . . . . 33
- BMdns . . . . . 34
- BMdnsService . . . . . 36
- BMeasureApi::BMeasure . . . . . 37
- BMeasureApi::BMeasureUnit . . . . . 55
- BMeasureApi::BMeasureUnit1 . . . . . 62
- BMeasureApi::BMeasureUnitDevice . . . . . 64
- BMeasureApi::BMeasureUnits . . . . . 65
- BMeasureApi::BMeasureUnitsDataBlock . . . . . 78
- BMeasureApi::BoardConfig . . . . . 80
- BMeasureApi::CalibrateInfo . . . . . 83
- BMeasureApi::ChannelConfig . . . . . 85
- BMeasureApi::CommsNet . . . . . 88
- BMeasureApi::CommsSerial . . . . . 92
- BMeasureApi::CommsUsb . . . . . 94
- BMeasureApi::ConfigItem . . . . . 98
- BMeasureApi::Configuration . . . . . 99
- BMeasureApi::DataBlock . . . . . 106
- BMeasureApi::DataFile . . . . . 109
- Dfu
  - The Dfu access class . . . . . 113
- DfuStatus . . . . . 117
- BMeasureApi::FileData . . . . . 117
- BMeasureApi::FileInfo . . . . . 118
- BMeasureApi::FilesysInfo . . . . . 120
- BMeasureApi::InfoBlock . . . . . 121
- BMeasureApi::Information . . . . . 124
- BMeasureApi::MeasurementConfig . . . . . 127
- BMeasureApi::NodeInfo . . . . . 129
- BMeasureApi::NodeStatus . . . . . 131
- BMeasureApi::Version . . . . . 133



# Chapter 5

## File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

BMdns.cpp	135
BMdns.h	137
BMeasureB-1.cpp	137
BMeasureB.cpp	138
BMeasureB.h	138
BMeasureD.cpp	138
BMeasureD.h	139
BMeasureLib.cpp	141
BMeasureLib.h	142
BMeasureS.cpp	142
BMeasureUnit.cpp	142
BMeasureUnit.h	144
BMeasureUnits.cpp	144
BMeasureUnits.h	145
CommsNet.cpp	145
CommsNet.h	146
CommsSerial.cpp	147
CommsSerial.h	147
CommsUsb.cpp	147
CommsUsb.h	148
DataFile.cpp	148
DataFile.h	149
Dfu.cpp	150
Dfu.h	158



# Chapter 6

## Namespace Documentation

### 6.1 BMeasureApi Namespace Reference

#### Classes

- class [AlarmConfig](#)
- class [AwgConfig](#)
- class [BMeasure](#)
- class [BMeasureUnit](#)
- class [BMeasureUnit1](#)
- class [BMeasureUnitDevice](#)
- class [BMeasureUnits](#)
- class [BMeasureUnitsDataBlock](#)
- class [BoardConfig](#)
- class [CalibrateInfo](#)
- class [ChannelConfig](#)
- class [CommsNet](#)
- class [CommsSerial](#)
- class [CommsUsb](#)
- class [ConfigItem](#)
- class [Configuration](#)
- class [DataBlock](#)
- class [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, `ErrorToFast` = 66 }
- 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, `StatusAlarm` = 0x0100 }
- 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,  
`CalibrateStageAttenScaling` = 8, `CalibrationStageFiveVolts` = 9 }
- enum `MessageSource` {  
`MessageSourceGeneral` = 0, `MessageSourceDebug` = 1, `MessageSourceTest` = 2, `MessageSourceWifi` = 3,  
`MessageSourceWifiTest` = 4 }
- enum `NetworkMode` { `NetworkModeOff` = 0, `NetworkModeDhcp` = 1, `NetworkModeManual` = 2 }
- enum `WifiMode` { `WifiModeOff`, `WifiModeClient`, `WifiModeAp` }
- enum `AlarmMode` { `AlarmModeOff`, `AlarmModeHigher`, `AlarmModeLower` }
- enum `AlarmOutput` {  
`AlarmOutputOff`, `AlarmOutputDioHigh`, `AlarmOutputDioLow`, `AlarmOutputRelayOn`,  
`AlarmOutputRelayOff` }
- enum `EventMode` { `EventModeOff`, `EventModeAlarm`, `EventModeSecond` }
- enum `Rs485Mode` { `Rs485ModeOff`, `Rs485ModeBoap` }

- 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)
- static `BUInt32 roundDown512` ( `BUInt32` size)
- const `BUInt32 TocMetaData` (1<< 1)
- const `BUInt32 TocNewObjList` (1<< 2)
- const `BUInt32 TocRawData` (1<< 3)
- const `BUInt32 TocInterleavedData` (1<< 5)
- const `BUInt32 TocBigEndian` (1<< 6)
- const `BUInt32 TocDaqRawData` (1<< 7)
- `BUInt32 round512` ( `BUInt32` s)

## Variables

- const `BUInt32 apiVersion` = 0

### 6.1.1 Typedef Documentation

#### 6.1.1.1 ChannelConfigs

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

### 6.1.2 Enumeration Type Documentation

#### 6.1.2.1 AlarmMode

```
enum BMeasureApi::AlarmMode
```

**Enumerator**

AlarmModeOff	
AlarmModeHigher	
AlarmModeLower	

**6.1.2.2 AlarmOutput**

enum `BMeasureApi::AlarmOutput`

**Enumerator**

AlarmOutputOff	
AlarmOutputDioHigh	
AlarmOutputDioLow	
AlarmOutputRelayOn	
AlarmOutputRelayOff	

**6.1.2.3 AwgOutput**

enum `BMeasureApi::AwgOutput`

**Enumerator**

AwgOutputNone	
AwgOutputAO0	
AwgOutputAO1	
AwgOutputAO01	

**6.1.2.4 BlockTypes**

enum `BMeasureApi::BlockTypes`

**Enumerator**

BlockTypeInfo	
BlockTypeData	



### 6.1.2.5 CalibrateStage

enum `BMeasureApi::CalibrateStage`

#### Enumerator

CalibrateStageNone	
CalibrateStageClear	
CalibrateStageSettle	
CalibrateStageAdcOffsets	
CalibrateStageDacOffsets	
CalibrateStageDacScaling0	
CalibrateStageDacScaling1	
CalibrateStageAdcScaling	
CalibrateStageAttenScaling	
CalibrationStageFiveVolts	

### 6.1.2.6 ChannelType

enum `BMeasureApi::ChannelType`

#### Enumerator

ChannelTypeNone	
ChannelTypeAnalogueIn	
ChannelTypeAnalogueOut	
ChannelTypeDigitalIn	
ChannelTypeDigitalOut	

### 6.1.2.7 DataBlockType

enum `BMeasureApi::DataBlockType`

#### Enumerator

DataBlockTypeFloat32	
DataBlockType125i	

### 6.1.2.8 DataSend

enum `BMeasureApi::DataSend`

**Enumerator**

DataSendOff	
DataSendOn	

**6.1.2.9 DigitalMode**

```
enum BMeasureApi::DigitalMode
```

**Enumerator**

DigitalModeInput	
DigitalModeOutput	
DigitalInOut	
DigitalModeSyncMaster	
DigitalModeSyncSlave	

**6.1.2.10 ErrorNum**

```
enum BMeasureApi::ErrorNum
```

**Enumerator**

ErrorSystem	
ErrorDataOverrun	
ErrorToFast	

**6.1.2.11 EventMode**

```
enum BMeasureApi::EventMode
```

**Enumerator**

EventModeOff	
EventModeAlarm	
EventModeSecond	

### 6.1.2.12 FilesysDeleteType

enum `BMeasureApi::FilesysDeleteType`

#### Enumerator

FilesysDeleteTypeNone	
FilesysDeleteTypeData	
FilesysDeleteTypeFormat	

### 6.1.2.13 FileType

enum `BMeasureApi::FileType`

#### Enumerator

FileTypeNone	
FileTypeFile	
FileTypeDir	

### 6.1.2.14 LogDataMode

enum `BMeasureApi::LogDataMode`

#### Enumerator

LogDataModeNormal	
LogDataModeDeleteOld	

### 6.1.2.15 MeasureMode

enum `BMeasureApi::MeasureMode`

#### Enumerator

MeasureModeOff	
MeasureModeOneShot	
MeasureModeRepeat	
MeasureModeContinuous	

### 6.1.2.16 MessageSource

enum [BMeasureApi::MessageSource](#)

#### Enumerator

MessageSourceGeneral	
MessageSourceDebug	
MessageSourceTest	
MessageSourceWifi	
MessageSourceWifiTest	

### 6.1.2.17 Mode

enum [BMeasureApi::Mode](#)

#### Enumerator

ModeIdle	
ModeRun	
ModeRunProgram	
ModeInternal	
ModeSleep	
ModeDemo1	

### 6.1.2.18 NetworkMode

enum [BMeasureApi::NetworkMode](#)

#### Enumerator

NetworkModeOff	
NetworkModeDhcp	
NetworkModeManual	

### 6.1.2.19 NodeType

enum [BMeasureApi::NodeType](#)

## Enumerator

NodeTypeNone	
NodeTypeBMeasure1	

## 6.1.2.20 Rs485Mode

```
enum BMeasureApi::Rs485Mode
```

## Enumerator

Rs485ModeOff	
Rs485ModeBoap	

## 6.1.2.21 SampleType

```
enum BMeasureApi::SampleType
```

## Enumerator

SampleTypeNone	
SampleTypeBool	
SampleTypeInt8	
SampleTypeInt16	
SampleTypeInt32	
SampleTypeFloat32	
SampleTypeFloat64	

## 6.1.2.22 SecureMode

```
enum BMeasureApi::SecureMode
```

## Enumerator

SecureModeOpen	
SecureMoteRemote	
SecureModeFull	

### 6.1.2.23 Status

enum `BMeasureApi::Status`

#### Enumerator

StatusNone	
StatusError	
StatusWarning	
StatusRun	
StatusTriggerWait	
StatusEnd0	
StatusEnd1	
StatusDataOverrun	
StatusFpgaOverrun	
StatusAlarm	

### 6.1.2.24 SyncMode

enum `BMeasureApi::SyncMode`

#### Enumerator

SyncModeOff	
SyncModeMaster	
SyncModeSlave	

### 6.1.2.25 TdsDataType

enum `BMeasureApi::TdsDataType`

#### Enumerator

TdsTypeVoid	
TdsTypeI8	
TdsTypeI16	
TdsTypeI32	
TdsTypeI64	
TdsTypeU8	
TdsTypeU16	
TdsTypeU32	
TdsTypeU64	
TdsTypeSingleFloat	
TdsTypeDoubleFloat	

## Enumerator

TdsTypeExtendedFloat	
TdsTypeSingleFloatWithUnit	
TdsTypeDoubleFloatWithUnit	
TdsTypeExtendedFloatWithUnit	
TdsTypeString	
TdsTypeBoolean	
TdsTypeTimeStamp	
TdsTypeFixedPoint	
TdsTypeComplexSingleFloat	
TdsTypeComplexDoubleFloat	
TdsTypeDAQmxRawData	

## 6.1.2.26 TriggerConfig

```
enum BMeasureApi::TriggerConfig
```

## Enumerator

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

## 6.1.2.27 TriggerMode

```
enum BMeasureApi::TriggerMode
```

## Enumerator

TriggerModeOff	
TriggerModePositive	
TriggerModeNegative	

## 6.1.2.28 Waveform

```
enum BMeasureApi::Waveform
```

## Enumerator

WaveformNone	
WaveformDc	

## Enumerator

WaveformSine	
WaveformSquare	
WaveformTriangle	
WaveformNoise	
WaveformArbitrary	

## 6.1.2.29 WifiMode

```
enum BMeasureApi::WifiMode
```

## Enumerator

WifiModeOff	
WifiModeClient	
WifiModeAp	

## 6.1.3 Function Documentation

## 6.1.3.1 channelTypeString()

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

## 6.1.3.2 round512()

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

## 6.1.3.3 roundDown512()

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



#### 6.1.3.4 sampleTypeString()

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

#### 6.1.3.5 TocBigEndian()

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

#### 6.1.3.6 TocDaqRawData()

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

#### 6.1.3.7 TocInterleavedData()

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

#### 6.1.3.8 TocMetaData()

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

#### 6.1.3.9 TocNewObjList()

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

#### 6.1.3.10 TocRawData()

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

#### 6.1.3.11 toFloat()

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

#### 6.1.3.12 unitSort()

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

### 6.1.4 Variable Documentation

#### 6.1.4.1 apiVersion

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

# Chapter 7

## Class Documentation

### 7.1 BMeasureApi::AlarmConfig Class Reference

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

#### Static Public Member Functions

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

#### Public Attributes

- [AlarmMode mode](#)  
*Alarm mode.*
- **BUInt8** [channel](#)  
*Alarm channel.*
- **BUInt8** [output](#)  
*Alarm output bitset.*
- **BUInt8** [outputChannel](#)  
*Alarm output channel.*
- **BFloat32** [level](#)  
*Alarm level.*

#### 7.1.1 Member Function Documentation

##### 7.1.1.1 [getMembers\(\)](#)

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

## 7.1.2 Member Data Documentation

### 7.1.2.1 channel

**BUInt8** BMeasureApi::AlarmConfig::channel

Alarm channel.

### 7.1.2.2 level

**BFloat32** BMeasureApi::AlarmConfig::level

Alarm level.

### 7.1.2.3 mode

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

Alarm mode.

### 7.1.2.4 output

**BUInt8** BMeasureApi::AlarmConfig::output

Alarm output bitset.

### 7.1.2.5 outputChannel

**BUInt8** BMeasureApi::AlarmConfig::outputChannel

Alarm output channel.

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

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

## 7.2 BMeasureApi::AwgConfig Class Reference

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

### Static Public Member Functions

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

### Public Attributes

- [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 %.*
- **BUInt32 numSamples**  
*The number of samples when using arbitrary waveforms.*

### 7.2.1 Member Function Documentation

#### 7.2.1.1 getMembers()

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

### 7.2.2 Member Data Documentation

#### 7.2.2.1 amplitude

```
BFloat32 BMeasureApi::AwgConfig::amplitude
```

The peak amplitude in Volts.

### 7.2.2.2 duty

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

The Duty cycle in %.

### 7.2.2.3 frequency

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

The frequency.

### 7.2.2.4 numSamples

**BUInt32** BMeasureApi::AwgConfig::numSamples

The number of samples when using arbitrary waveforms.

### 7.2.2.5 offset

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

The DC offset in volts.

### 7.2.2.6 output

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

The output channels.

### 7.2.2.7 spare

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

### 7.2.2.8 waveform

`Waveform` BMeasureApi::AwgConfig::waveform

The waveform.

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

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

## 7.3 BFirmwareInfo Struct Reference

### Public Attributes

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

### 7.3.1 Member Data Documentation

#### 7.3.1.1 checksum

**BUInt32** BFirmwareInfo::checksum

#### 7.3.1.2 length

**BUInt32** BFirmwareInfo::length

#### 7.3.1.3 magic

**BUInt32** BFirmwareInfo::magic

#### 7.3.1.4 type

**BUInt8** BFirmwareInfo::type

#### 7.3.1.5 ver0

**BUInt8** BFirmwareInfo::ver0

#### 7.3.1.6 ver1

**BUInt8** BFirmwareInfo::ver1

#### 7.3.1.7 ver2

**BUInt8** BFirmwareInfo::ver2

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

- [Dfu.cpp](#)

## 7.4 BMdns Class Reference

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

### Public Member Functions

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

### Private Attributes

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

#### 7.4.1 Constructor & Destructor Documentation



### 7.4.1.1 BMdns()

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

### 7.4.1.2 ~BMdns()

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

## 7.4.2 Member Function Documentation

### 7.4.2.1 findServices()

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

Unicast response, class IN

### 7.4.2.2 init()

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

## 7.4.3 Member Data Documentation

### 7.4.3.1 osocket

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

### 7.4.3.2 otransactionId

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

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

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

## 7.5 BMdnsService Class Reference

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

### Public Attributes

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

### 7.5.1 Member Data Documentation

#### 7.5.1.1 address

```
BSocketAddressINET BMdnsService::address
```

#### 7.5.1.2 extra

```
BStringList BMdnsService::extra
```

#### 7.5.1.3 hostname

```
BString BMdnsService::hostname
```

#### 7.5.1.4 name

```
BString BMdnsService::name
```

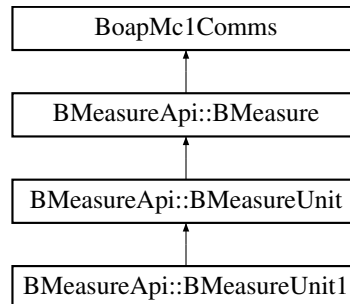
The documentation for this class was generated from the following file:

- [BMdns.h](#)

## 7.6 BMeasureApi::BMeasure Class Reference

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

Inheritance diagram for BMeasureApi::BMeasure:



### Public Member Functions

- [BMeasure](#) ( **Bool** threaded=0, **BUInt** reqSize=512)
- **BError** [getNodeInfo](#) ([NodeInfo](#) &nodeInfo)  
*Get node information.*
- 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** [setMode](#) (const [Mode](#) &mode)  
*Set the current operational mode.*
- **BError** [setSecureMode](#) (const [SecureMode](#) &secureMode)  
*Set the security mode.*
- **BError** [setSecureKey](#) (const **BString** &key)  
*Set the security key.*
- **BError** [login](#) (const **BString** &user, const **BString** &password)  
*Provides user/password information for secure connection.*
- **BError** [getInformation](#) ([Information](#) &info)
- **BError** [getInfoBlock](#) ([InfoBlock](#) &infoBlock)
- **BError** [getChannelConfig](#) (const **BUInt32** &channelNumber, [ChannelConfig](#) &channelConfig)
- **BError** [setChannelConfig](#) (const **BUInt32** &channelNumber, const [ChannelConfig](#) &channelConfig)
- **BError** [setChannelConfigFull](#) (const **BUInt64** &key, const **BUInt32** &channelNumber, const [ChannelConfig](#) &channelConfig)
- **BError** [getConfig](#) ([Configuration](#) &config)  
*Return units configuration.*
- **BError** [setConfig](#) (const [Configuration](#) &config)  
*Set units configuration.*
- **BError** [getMeasurementConfig](#) ([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 **BUInt32** &chan, const **BUInt32** &pos, const [FileData](#) &dataBlock)

*Configure AWG Arbitrary waveform.*
- **BError** [setAnalogueOut](#) (const **BUInt32** &chan, const **BFloat32** &value)

*Set analogue output value.*
- **BError** [setDigital](#) (const **BUInt32** &bits)

*Set digital bits.*
- **BError** [getDigital](#) (**BUInt32** &bits)

*Get digital bits.*
- **BError** [setRelay](#) (const **BUInt32** &relayNum, const **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** [setModeServe](#) (const [Mode](#) &mode)
- virtual **BError** [setSecureModeServe](#) (const [SecureMode](#) &secureMode)
- virtual **BError** [setSecureKeyServe](#) (const **BString** &key)
- virtual **BError** [loginServe](#) (const **BString** &user, const **BString** &password)
- virtual **BError** [getInformationServe](#) ([Information](#) &info)
- virtual **BError** [getInfoBlockServe](#) ([InfoBlock](#) &infoBlock)
- virtual **BError** [getChannelConfigServe](#) (const **BUInt32** &channelNumber, [ChannelConfig](#) &channelConfig)
- virtual **BError** [setChannelConfigServe](#) (const **BUInt32** &channelNumber, const [ChannelConfig](#) &channelConfig)
- virtual **BError** [setChannelConfigFullServe](#) (const **BUInt64** &key, const **BUInt32** &channelNumber, const [ChannelConfig](#) &channelConfig)
- virtual **BError** [getConfigServe](#) ([Configuration](#) &config)
- virtual **BError** [setConfigServe](#) (const [Configuration](#) &config)
- virtual **BError** [getMeasurementConfigServe](#) ([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 **BUInt32** &chan, const **BUInt32** &pos, const [FileData](#) &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 [CalibrateInfo](#) &calibInfo)
- virtual void [sendMessageServe](#) (const **BUInt32** &source, const **BString** &message)

## Additional Inherited Members

### 7.6.1 Constructor & Destructor Documentation

### 7.6.1.1 BMeasure()

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

## 7.6.2 Member Function Documentation

### 7.6.2.1 calibrate()

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

Calibrate system.

### 7.6.2.2 calibrateServe()

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

### 7.6.2.3 factoryReset()

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

Factory reset.

### 7.6.2.4 factoryResetServe()

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

### 7.6.2.5 fileClose()

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

### 7.6.2.6 fileCloseServe()

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

### 7.6.2.7 fileDelete()

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

### 7.6.2.8 fileDeleteServe()

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

### 7.6.2.9 fileList()

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

### 7.6.2.10 fileListServe()

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

### 7.6.2.11 fileOpen()

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

### 7.6.2.12 fileOpenServe()

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

### 7.6.2.13 fileRead()

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

### 7.6.2.14 fileReadServe()

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

### 7.6.2.15 fileSysDelete()

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

### 7.6.2.16 fileSysDeleteServe()

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

### 7.6.2.17 fileSysInfo()

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



### 7.6.2.18 fileSysInfoServe()

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

### 7.6.2.19 fileWrite()

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

### 7.6.2.20 fileWriteServe()

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

### 7.6.2.21 functionUnlock()

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

Unlock/Lock special functions.

### 7.6.2.22 functionUnlockServe()

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

### 7.6.2.23 getAwgConfig()

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

Get AWG [Configuration](#).

#### 7.6.2.24 getAwgConfigServe()

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

#### 7.6.2.25 getBoardConfig()

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

Get the boards configuration.

#### 7.6.2.26 getBoardConfigServe()

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

#### 7.6.2.27 getChannelConfig()

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

#### 7.6.2.28 getChannelConfigServe()

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

#### 7.6.2.29 getConfig()

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

Return units configuration.

### 7.6.2.30 getConfigServe()

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

### 7.6.2.31 getDigital()

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

Get digital bits.

### 7.6.2.32 getDigitalServe()

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

### 7.6.2.33 getInfoBlock()

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

### 7.6.2.34 getInfoBlockServe()

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

### 7.6.2.35 getInformation()

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

#### 7.6.2.36 `getInformationServe()`

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

#### 7.6.2.37 `getMeasurement()`

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

Get measurement config.

#### 7.6.2.38 `getMeasurementConfig()`

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

Get measurement config.

#### 7.6.2.39 `getMeasurementConfigServe()`

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

#### 7.6.2.40 `getMeasurementServe()`

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

#### 7.6.2.41 `getNodeInfo()`

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

Get node information.

#### 7.6.2.42 getNodeInfoServe()

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

#### 7.6.2.43 getStatus()

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

Get the node status.

#### 7.6.2.44 getStatusServe()

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

#### 7.6.2.45 getSwitch()

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

Get digital bits.

#### 7.6.2.46 getSwitchServe()

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

#### 7.6.2.47 login()

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

Provides user/password information for secure connection.

#### 7.6.2.48 loginServe()

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

#### 7.6.2.49 measure()

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

Performs a single measurement.

#### 7.6.2.50 measureServe()

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

#### 7.6.2.51 processRequest()

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

Reimplemented from **BoapMc1Comms**.

#### 7.6.2.52 runBoardTest()

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

Runs the given board test.

#### 7.6.2.53 runBoardTestServe()

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

#### 7.6.2.54 sendData()

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

Sends a data block.

#### 7.6.2.55 sendDataEnable()

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

Enable the sending of data.

#### 7.6.2.56 sendDataEnableServe()

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

#### 7.6.2.57 sendDataServe()

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

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

#### 7.6.2.58 sendInfo()

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

Sends an info block.

#### 7.6.2.59 sendInfoServe()

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

### 7.6.2.60 sendMessage()

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

Sends text messages.

### 7.6.2.61 sendMessageServe()

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

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

### 7.6.2.62 sendStatus()

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

Sends the current status.

### 7.6.2.63 sendStatusServe()

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

### 7.6.2.64 sendTime()

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

Sends the current time.

### 7.6.2.65 sendTimeServe()

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



#### 7.6.2.66 setAnalogueOut()

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

Set analogue output value.

#### 7.6.2.67 setAnalogueOutServe()

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

#### 7.6.2.68 setAwgConfig()

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

Configure AWG.

#### 7.6.2.69 setAwgConfigServe()

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

#### 7.6.2.70 setAwgWaveform()

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

Configure AWG Arbitrary waveform.

#### 7.6.2.71 setAwgWaveformServe()

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

### 7.6.2.72 setBoardConfig()

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

Sets the boards configuration, requires key.

### 7.6.2.73 setBoardConfigServe()

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

### 7.6.2.74 setChannelConfig()

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

### 7.6.2.75 setChannelConfigFull()

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

### 7.6.2.76 setChannelConfigFullServe()

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

### 7.6.2.77 setChannelConfigServe()

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

#### 7.6.2.78 setConfig()

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

Set units configuration.

#### 7.6.2.79 setConfigServe()

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

#### 7.6.2.80 setDigital()

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

Set digital bits.

#### 7.6.2.81 setDigitalServe()

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

#### 7.6.2.82 setMeasurement()

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

Set measurement config.

#### 7.6.2.83 setMeasurementConfig()

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

Set measurement config.

#### 7.6.2.84 setMeasurementConfigServe()

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

#### 7.6.2.85 setMeasurementServe()

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

#### 7.6.2.86 setMode()

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

Set the current operational mode.

#### 7.6.2.87 setModeServe()

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

#### 7.6.2.88 setRelay()

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

Set relay.

#### 7.6.2.89 setRelayServe()

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

## 7.6.2.90 setSecureKey()

```
BError BMeasureApi::BMeasure::setSecureKey (
    const BString & key )
```

Set the security key.

## 7.6.2.91 setSecureKeyServe()

```
BError BMeasureApi::BMeasure::setSecureKeyServe (
    const BString & key ) [virtual]
```

## 7.6.2.92 setSecureMode()

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

Set the security mode.

## 7.6.2.93 setSecureModeServe()

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

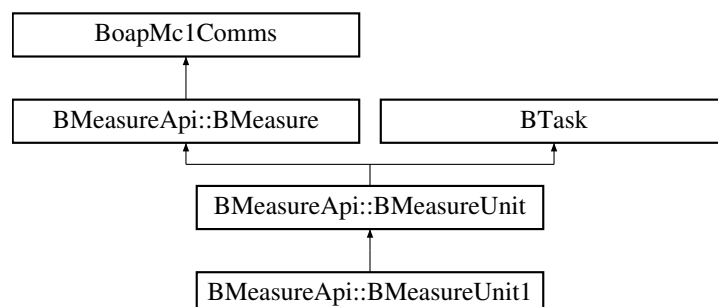
The documentation for this class was generated from the following files:

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

## 7.7 BMeasureApi::BMeasureUnit Class Reference

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

Inheritance diagram for BMeasureApi::BMeasureUnit:



## Public Member Functions

- **BMeasureUnit** ( **Bool** threaded=0, **BUInt** reqSize=2048)
- virtual **~BMeasureUnit** ()
- **BError** **connect** ( **BString** device)
  - Connect to a device.*
- void **disconnect** ()
- **BString** **device** ()
- **BString** **serialNumber** ()
- **BString** **info** ()
- **BUInt** **numChannels** ()
  - The number of channels of data.*
- void **run** ()
  - Threaded run mode.*
- virtual void **disconnected** ()
- virtual void **sendDataServe** (const **DataBlock** &dataBlock)
- virtual void **sendDataServe1** (const **DataBlock** &dataBlock)
- virtual **BError** **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

### 7.7.1 Constructor & Destructor Documentation

#### 7.7.1.1 BMeasureUnit()

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

#### 7.7.1.2 ~BMeasureUnit()

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

### 7.7.2 Member Function Documentation

#### 7.7.2.1 connect()

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

Connect to a device.

#### 7.7.2.2 device()

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

#### 7.7.2.3 disconnect()

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

#### 7.7.2.4 disconnected()

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

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

#### 7.7.2.5 findDevices()

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

Find available devices.

#### 7.7.2.6 findDevicesNetwork()

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

Find available devices on Network.

#### 7.7.2.7 findDevicesUsb()

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

Find available devices on USB bus.

#### 7.7.2.8 info()

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

#### 7.7.2.9 numChannels()

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

The number of channels of data.



#### 7.7.2.10 processdataBlock()

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

#### 7.7.2.11 run()

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

Threaded run mode.

Reimplemented from [BTask](#).

#### 7.7.2.12 sendDataServe()

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

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

#### 7.7.2.13 sendDataServe1()

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

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

#### 7.7.2.14 serialNumber()

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

#### 7.7.2.15 setChannelConfig()

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

### 7.7.2.16 setMeasurement()

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

## 7.7.3 Member Data Documentation

### 7.7.3.1 blockNumChannels

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

### 7.7.3.2 blockNumSamples

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

### 7.7.3.3 oblockCount

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

### 7.7.3.4 ochannels

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

### 7.7.3.5 oconfigMeasurement

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

### 7.7.3.6 odataBlock

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

### 7.7.3.7 odevice

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

### 7.7.3.8 odisconnecting

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

### 7.7.3.9 oinfo

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

Instrument info.

### 7.7.3.10 onodeInfo

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

### 7.7.3.11 osampleCount

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

### 7.7.3.12 osequenceNext

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

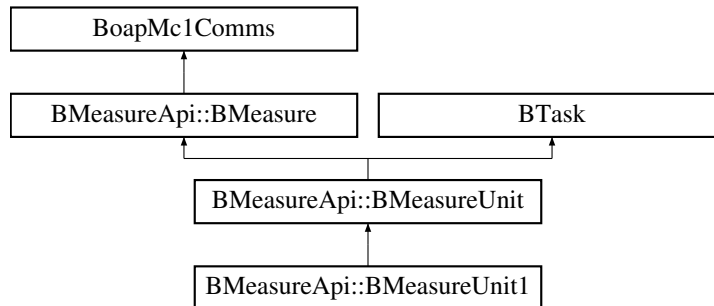
The documentation for this class was generated from the following files:

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

## 7.8 BMeasureApi::BMeasureUnit1 Class Reference

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

Inheritance diagram for BMeasureApi::BMeasureUnit1:



### Public Member Functions

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

### Public Attributes

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

### Additional Inherited Members

#### 7.8.1 Constructor & Destructor Documentation

##### 7.8.1.1 BMeasureUnit1()

```

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

## 7.8.2 Member Function Documentation

### 7.8.2.1 disconnected()

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

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

### 7.8.2.2 sendDataServe1()

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

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

### 7.8.2.3 sendMessageServe()

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

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

### 7.8.2.4 serialNumber()

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

### 7.8.2.5 setSerialNumber()

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

## 7.8.3 Member Data Documentation

### 7.8.3.1 oconnected

**Bool** BMeasureApi::BMeasureUnit1::oconnected

### 7.8.3.2 oenabled

**Bool** BMeasureApi::BMeasureUnit1::oenabled

### 7.8.3.3 omeasureUnits

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

### 7.8.3.4 oorder

**UInt** BMeasureApi::BMeasureUnit1::oorder

### 7.8.3.5 oserialNumber

**String** BMeasureApi::BMeasureUnit1::oserialNumber

### 7.8.3.6 osource

**UInt** BMeasureApi::BMeasureUnit1::osource

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

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

## 7.9 BMeasureApi::BMeasureUnitDevice Class Reference

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

## Public Member Functions

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

## Public Attributes

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

## 7.9.1 Constructor & Destructor Documentation

### 7.9.1.1 BMeasureUnitDevice()

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

## 7.9.2 Member Data Documentation

### 7.9.2.1 device

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

### 7.9.2.2 serialNumber

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

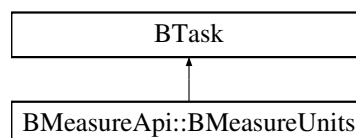
The documentation for this class was generated from the following file:

- [BMeasureUnit.h](#)

## 7.10 BMeasureApi::BMeasureUnits Class Reference

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

Inheritance diagram for BMeasureApi::BMeasureUnits:



## Public Member Functions

- [BMeasureUnits](#) ( **Bool** threaded=0)
- virtual [~BMeasureUnits](#) ()
- void [clear](#) ()
- **BError** [unitsFind](#) ()
- **BError** [unitAdd](#) ( **BString** serialNumber, **BString** device)
- **BError** [unitDelete](#) ( **BString** device)
- **BUInt32** [unitsNum](#) ()
- **BUInt32** [unitsConnectedNum](#) ()
- [BMeasureUnit1](#) & [unit](#) ( **BUInt** u)
- [BMeasureUnit1](#) & [unitMaster](#) ()
- **BError** [unitsConnect](#) ()
- **Bool** [unitsConnected](#) ()
- **BError** [unitsDisconnect](#) ()
- virtual void [disconnected](#) ()
- **BError** [unitSetOrder](#) ( **BUInt** u, **BUInt** order, **Bool** move)
- **BError** [unitSetEnabled](#) ( **BUInt** u, **Bool** enable)
- **BError** [dataSetNumStreams](#) ( **BUInt** num)  
*Set the number of data output channels.*
- void [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

#### 7.10.1 Constructor & Destructor Documentation

### 7.10.1.1 BMeasureUnits()

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

### 7.10.1.2 ~BMeasureUnits()

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

## 7.10.2 Member Function Documentation

### 7.10.2.1 clear()

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

### 7.10.2.2 dataAvailable()

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

### 7.10.2.3 dataClear()

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

### 7.10.2.4 dataDone()

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

### 7.10.2.5 dataEvent()

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

### 7.10.2.6 dataProcessEnable()

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

Enable the processing of data.

### 7.10.2.7 dataRead()

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

### 7.10.2.8 dataSetNumStreams()

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

Set the number of data output channels.

### 7.10.2.9 dataWait()

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

### 7.10.2.10 debugPrint()

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

### 7.10.2.11 disconnected()

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

#### 7.10.2.12 getAwgConfig()

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

Get AWG [Configuration](#).

#### 7.10.2.13 getChannelConfig()

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

#### 7.10.2.14 getConfig()

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

Should we have this, not generic for different instruments ?

#### 7.10.2.15 getFreeBlock()

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

#### 7.10.2.16 getInfoBlock()

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

#### 7.10.2.17 getInformation()

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

#### 7.10.2.18 getMeasurement()

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

Get measurement settings.

#### 7.10.2.19 getMeasurementConfig()

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

Get measurement config.

#### 7.10.2.20 getStatus()

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

#### 7.10.2.21 numChannels()

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

The number of channels of data.

#### 7.10.2.22 outputBlock()

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

#### 7.10.2.23 run()

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

Threaded run mode.

Reimplemented from **BTask**.

#### 7.10.2.24 sendDataEnable()

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

Enables the sending of data.

#### 7.10.2.25 sendDataProcess()

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

#### 7.10.2.26 sendDataProcessTrigger()

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

#### 7.10.2.27 sendDataQueue()

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

#### 7.10.2.28 sendDataServe1()

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

#### 7.10.2.29 sendMessage()

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

#### 7.10.2.30 sendMessageServe()

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

### 7.10.2.31 sendTime()

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

Sends the current time.

### 7.10.2.32 setAwgConfig()

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

Configure AWG.

### 7.10.2.33 setChannelConfig()

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

### 7.10.2.34 setConfig()

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

Should we have this, not generic for different instruments ?

### 7.10.2.35 setMeasurement()

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

Set measurement settings.

### 7.10.2.36 setMeasurementConfig()

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

Set measurement config.

### 7.10.2.37 setMode()

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

Set the current operational mode.

### 7.10.2.38 unit()

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

### 7.10.2.39 unitAdd()

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

### 7.10.2.40 unitDelete()

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

### 7.10.2.41 unitMaster()

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

### 7.10.2.42 unitsConnect()

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

### 7.10.2.43 unitsConnected()

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



#### 7.10.2.44 unitsConnectedNum()

**BUInt** BMeasureApi::BMeasureUnits::unitsConnectedNum ( )

#### 7.10.2.45 unitsDisconnect()

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

#### 7.10.2.46 unitSetEnabled()

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

#### 7.10.2.47 unitSetOrder()

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

#### 7.10.2.48 unitsFind()

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

#### 7.10.2.49 unitsNum()

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

### 7.10.3 Member Data Documentation

### 7.10.3.1 odataBlocksFree

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

### 7.10.3.2 odataBlocksIn

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

### 7.10.3.3 odataBlocksOut

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

### 7.10.3.4 odataBlocksOutCount

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

### 7.10.3.5 odataBlocksProcess

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

### 7.10.3.6 odataBlocksProcessNum

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

### 7.10.3.7 odataStreamNum

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

### 7.10.3.8 ofill

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

### 7.10.3.9 olocalTrigger

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

### 7.10.3.10 olockInput

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

### 7.10.3.11 olockOutput

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

### 7.10.3.12 olockUnits

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

### 7.10.3.13 onumBlocks

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

### 7.10.3.14 onumChannels

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

### 7.10.3.15 onumConnected

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

#### 7.10.3.16 oprocEnable

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

Enable processing.

#### 7.10.3.17 oprocRunning

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

Processing is running.

#### 7.10.3.18 ostartSample

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

#### 7.10.3.19 otriggered

```
Bool BMeasureApi::BMeasureUnits::ottriggered [private]
```

#### 7.10.3.20 ounitMaster

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

#### 7.10.3.21 ounits

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

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

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

## 7.11 BMeasureApi::BMeasureUnitsDataBlock Class Reference

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

## Public Member Functions

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

## Public Attributes

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

## 7.11.1 Constructor & Destructor Documentation

### 7.11.1.1 BMeasureUnitsDataBlock()

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

### 7.11.1.2 ~BMeasureUnitsDataBlock()

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

## 7.11.2 Member Function Documentation

### 7.11.2.1 init()

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

## 7.11.3 Member Data Documentation

### 7.11.3.1 odataBlock

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

### 7.11.3.2 ofill

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

### 7.11.3.3 oinUse

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

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

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

## 7.12 BMeasureApi::BoardConfig Class Reference

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

### Static Public Member Functions

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

### Public Attributes

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

## 7.12.1 Member Function Documentation

### 7.12.1.1 getMembers()

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

## 7.12.2 Member Data Documentation

### 7.12.2.1 buildTime

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

### 7.12.2.2 calibAdcOffsets

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

### 7.12.2.3 calibAdcScales

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

### 7.12.2.4 calibAttenScales

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

### 7.12.2.5 calibDacOffsets

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

#### 7.12.2.6 calibDacScales

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

#### 7.12.2.7 calibFiveVolts

**BFloat64** BMeasureApi::BoardConfig::calibFiveVolts

#### 7.12.2.8 calibTemp

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

#### 7.12.2.9 calibTime

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

#### 7.12.2.10 hardwareVersion

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

#### 7.12.2.11 macAddress

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

#### 7.12.2.12 magic

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

#### 7.12.2.13 serialNumber

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



## 7.12.2.14 spare0

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

## 7.12.2.15 testMode

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

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

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

## 7.13 BMeasureApi::CalibrateInfo Class Reference

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

## Static Public Member Functions

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

## Public Attributes

- **BUInt32** [stage](#)  
*Stage to run.*
- **BFloat64** [calibrateFrequency](#)  
*The Awg frequency for calibration.*
- **BFloat64** [calibrateAmplitude](#)  
*The Awg amplitude for calibration.*
- **BFloat64** [calibrateTime](#)  
*Number of seconds to calibrate over (synced to multiple AWG cycles)*
- **BFloat64** [value](#)  
*Target/Set Value.*

## 7.13.1 Member Function Documentation

## 7.13.1.1 getMembers()

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

## 7.13.2 Member Data Documentation

### 7.13.2.1 calibrateAmplitude

**BFloat64** BMeasureApi::CalibrateInfo::calibrateAmplitude

The Awg amplitude for calibration.

### 7.13.2.2 calibrateFrequency

**BFloat64** BMeasureApi::CalibrateInfo::calibrateFrequency

The Awg frequency for calibration.

### 7.13.2.3 calibrateTime

**BFloat64** BMeasureApi::CalibrateInfo::calibrateTime

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

### 7.13.2.4 stage

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

Stage to run.

### 7.13.2.5 value

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

Target/Set Value.

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

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

## 7.14 BMeasureApi::ChannelConfig Class Reference

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

### Static Public Member Functions

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

### Public Attributes

- **BUInt8** [number](#)  
*The channel number.*
- **BUInt8** [enabled](#)  
*Channel is enabled.*
- **BUInt8** [attenuator](#)  
*Attenuator number in use.*
- [ChannelType](#) [type](#)  
*The channel type.*
- [SampleType](#) [sampleType](#)  
*The sample type.*
- **BUInt8** [spare0](#) [3]
- **BUInt32** [dataChannel](#)  
*Data channel.*
- **BChar** [id](#) [16]
- **BChar** [name](#) [16]
- **BChar** [siUnits](#) [8]
- **BFloat64** [calibOffset](#)  
*The calibration data offset.*
- **BFloat64** [calibScale](#)  
*The calibration data scale factor to volts.*
- **BFloat64** [calibScaleAtten1](#)  
*Attenuator 1 scaling.*
- **BFloat64** [pgaGain](#)  
*The PGA gain.*
- **BFloat64** [scale](#)  
*The user data scale factor.*
- **BFloat64** [offset](#)  
*The user data offset.*
- **BChar** [process](#) [32]

### 7.14.1 Member Function Documentation

#### 7.14.1.1 [getMembers\(\)](#)

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

## 7.14.2 Member Data Documentation

### 7.14.2.1 attenuator

**BUInt8** BMeasureApi::ChannelConfig::attenuator

Attenuator number in use.

### 7.14.2.2 calibOffset

**BFloat64** BMeasureApi::ChannelConfig::calibOffset

The calibration data offset.

### 7.14.2.3 calibScale

**BFloat64** BMeasureApi::ChannelConfig::calibScale

The calibration data scale factor to volts.

### 7.14.2.4 calibScaleAtten1

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

Attenuator 1 scaling.

### 7.14.2.5 dataChannel

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

Data channel.

#### 7.14.2.6 enabled

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

Channel is enabled.

#### 7.14.2.7 id

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

#### 7.14.2.8 name

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

#### 7.14.2.9 number

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

The channel number.

#### 7.14.2.10 offset

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

The user data offset.

#### 7.14.2.11 pgaGain

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

The PGA gain.

#### 7.14.2.12 process

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

#### 7.14.2.13 sampleType

`SampleType` `BMeasureApi::ChannelConfig::sampleType`

The sample type.

#### 7.14.2.14 scale

`BFloat64` `BMeasureApi::ChannelConfig::scale`

The user data scale factor.

#### 7.14.2.15 siUnits

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

#### 7.14.2.16 spare0

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

#### 7.14.2.17 type

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

The channel type.

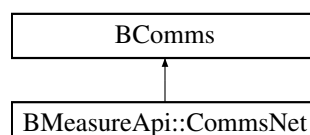
The documentation for this class was generated from the following files:

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

## 7.15 BMeasureApi::CommsNet Class Reference

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

Inheritance diagram for `BMeasureApi::CommsNet`:



## Public Member Functions

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

## Protected Attributes

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

## Additional Inherited Members

### 7.15.1 Constructor & Destructor Documentation

#### 7.15.1.1 CommsNet()

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

#### 7.15.1.2 ~CommsNet()

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

### 7.15.2 Member Function Documentation

#### 7.15.2.1 connect()

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

### 7.15.2.2 disconnect()

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

Reimplemented from **BComms**.

### 7.15.2.3 init()

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

Reimplemented from **BComms**.

### 7.15.2.4 read()

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

Implements **BComms**.

### 7.15.2.5 readAvailable()

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

Reimplemented from **BComms**.

### 7.15.2.6 wait()

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

### 7.15.2.7 write()

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

Implements **BComms**.



### 7.15.2.8 writeAvailable()

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

Reimplemented from **BComms**.

### 7.15.2.9 writeChunks()

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

Reimplemented from **BComms**.

## 7.15.3 Member Data Documentation

### 7.15.3.1 oinWait

```
Bool BMeasureApi::CommsNet::oinWait [protected]
```

### 7.15.3.2 osocket

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

### 7.15.3.3 oterminating

```
Bool BMeasureApi::CommsNet::oterminating [protected]
```

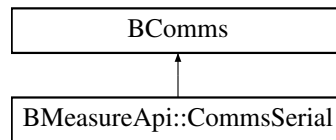
The documentation for this class was generated from the following files:

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

## 7.16 BMeasureApi::CommsSerial Class Reference

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

Inheritance diagram for BMeasureApi::CommsSerial:



### Public Member Functions

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

### Private Attributes

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

### Additional Inherited Members

#### 7.16.1 Constructor & Destructor Documentation

##### 7.16.1.1 CommsSerial()

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

##### 7.16.1.2 ~CommsSerial()

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

## 7.16.2 Member Function Documentation

### 7.16.2.1 connect()

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

### 7.16.2.2 disconnect()

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

Reimplemented from **BComms**.

### 7.16.2.3 read()

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

Implements **BComms**.

### 7.16.2.4 readAvailable()

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

Reimplemented from **BComms**.

### 7.16.2.5 wait()

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

### 7.16.2.6 write()

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

Implements **BComms**.

## 7.16.3 Member Data Documentation

### 7.16.3.1 odevice

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

### 7.16.3.2 oserialPort

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

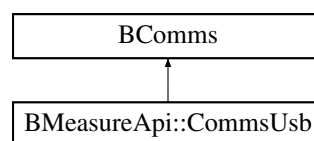
The documentation for this class was generated from the following file:

- [CommsSerial.h](#)

## 7.17 BMeasureApi::CommsUsb Class Reference

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

Inheritance diagram for BMeasureApi::CommsUsb:



### Public Member Functions

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

## Private Member Functions

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

## Private Attributes

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

## Additional Inherited Members

### 7.17.1 Constructor & Destructor Documentation

#### 7.17.1.1 CommsUsb()

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

#### 7.17.1.2 ~CommsUsb()

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

### 7.17.2 Member Function Documentation

#### 7.17.2.1 connect()

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

#### 7.17.2.2 disconnect()

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

Reimplemented from **BComms**.

### 7.17.2.3 read()

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

Implements **BComms**.

### 7.17.2.4 readAvailable()

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

Reimplemented from **BComms**.

### 7.17.2.5 readChunk()

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

### 7.17.2.6 wait()

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

### 7.17.2.7 write()

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

Implements **BComms**.

## 7.17.3 Member Data Documentation

### 7.17.3.1 obuffer

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

### 7.17.3.2 ocontext

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

### 7.17.3.3 odev

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

### 7.17.3.4 odevice

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

### 7.17.3.5 onum

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

### 7.17.3.6 oterminated

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

### 7.17.3.7 oterminating

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

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

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

## 7.18 BMeasureApi::ConfigItem Class Reference

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

### Static Public Member Functions

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

### Public Attributes

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

### 7.18.1 Member Function Documentation

#### 7.18.1.1 [getMembers\(\)](#)

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

### 7.18.2 Member Data Documentation

#### 7.18.2.1 [name](#)

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

#### 7.18.2.2 [spare](#)

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



## 7.18.2.3 type

**BUInt8** BMeasureApi::ConfigItem::type

The type of data.

## 7.18.2.4 value

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

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

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

## 7.19 BMeasureApi::Configuration Class Reference

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

## Static Public Member Functions

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

## Public Attributes

- **BUInt32** [version](#)  
*The configuration version.*
- **BChar** [name](#) [16]
- **BChar** [location](#) [16]
- **Mode** [mode](#)  
*The boot run mode.*
- **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** [spare0](#)
- **DigitalMode** [digitalMode](#)  
*The digital mode.*
- **BUInt8** [digitalPins](#) [8]

- [NetworkMode networkMode](#)  
*The network mode (0 - off, 1 - dhcp, 2 - manual)*
- **BUInt8** [spare1](#) [3]
- **BUInt32** [networkAddress](#)  
*The network IP address.*
- **BUInt32** [networkMask](#)  
*The network netmask.*
- **BUInt32** [networkGateway](#)  
*The network gateway.*
- **BUInt32** [networkNameServer0](#)  
*The network nameserver.*
- **BUInt32** [networkTimeServer](#)  
*The network timeserver.*
- [Rs485Mode rs485Mode](#)  
*The RS485 mode.*
- **BUInt8** [rs485Bits](#)  
*The RS485 number of bits.*
- **BUInt8** [rs485StopBits](#)  
*The RS485 stop bits.*
- **BUInt8** [spare2](#)
- **BUInt32** [rs485BaudRate](#)  
*The RS485 baud rate.*
- [WifiMode wifiMode](#)  
*The wifi mode.*
- **BUInt8** [spare3](#) [3]
- **BChar** [wifiAp0](#) [32]
- **BChar** [wifiAp0Password](#) [32]
- [AlarmConfig alarms](#) [8]
- [EventMode mqttMode](#)  
*MQTT mode.*
- **BUInt8** [spare4](#) [3]
- **BChar** [mqttServer](#) [32]
- **BUInt32** [mqttPort](#)  
*The MQTT port.*
- [EventMode emailMode](#)  
*Email mode.*
- **BUInt8** [spare5](#) [3]
- **BChar** [emailAddress](#) [32]
- **BChar** [program](#) [32]

## 7.19.1 Member Function Documentation

### 7.19.1.1 getMembers()

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

## 7.19.2 Member Data Documentation

### 7.19.2.1 alarms

`AlarmConfig` BMeasureApi::Configuration::alarms[8]

### 7.19.2.2 digitalMode

`DigitalMode` BMeasureApi::Configuration::digitalMode

The digital mode.

### 7.19.2.3 digitalPins

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

### 7.19.2.4 emailAddress

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

### 7.19.2.5 emailMode

`EventMode` BMeasureApi::Configuration::emailMode

Email mode.

### 7.19.2.6 location

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

#### 7.19.2.7 logData

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

Log the data.

#### 7.19.2.8 logDataDevice

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

The device to store data.

#### 7.19.2.9 logDataMode

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

Log data mode.

#### 7.19.2.10 mode

[Mode](#) BMeasureApi::Configuration::mode

The boot run mode.

#### 7.19.2.11 mqttMode

[EventMode](#) BMeasureApi::Configuration::mqttMode

MQTT mode.

#### 7.19.2.12 mqttPort

**BUInt32** BMeasureApi::Configuration::mqttPort

The MQTT port.

#### 7.19.2.13 mqttServer

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

#### 7.19.2.14 name

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

#### 7.19.2.15 networkAddress

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

The network IP address.

#### 7.19.2.16 networkGateway

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

The network gateway.

#### 7.19.2.17 networkMask

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

The network netmask.

#### 7.19.2.18 networkMode

[NetworkMode](#) BMeasureApi::Configuration::networkMode

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

#### 7.19.2.19 networkNameServer0

**BUInt32** BMeasureApi::Configuration::networkNameServer0

The network nameserver.

#### 7.19.2.20 networkTimeServer

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

The network timeserver.

#### 7.19.2.21 program

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

#### 7.19.2.22 rs485BaudRate

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

The RS485 baud rate.

#### 7.19.2.23 rs485Bits

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

The RS485 number of bits.

#### 7.19.2.24 rs485Mode

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

The RS485 mode.

#### 7.19.2.25 rs485StopBits

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

The RS485 stop bits.

#### 7.19.2.26 sampleFrequencyMode

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

The base sample frequency mode.

#### 7.19.2.27 source

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

The source number if multiple units are in use.

#### 7.19.2.28 spare0

**BUInt8** BMeasureApi::Configuration::spare0

#### 7.19.2.29 spare1

**BUInt8** BMeasureApi::Configuration::spare1[3]

#### 7.19.2.30 spare2

**BUInt8** BMeasureApi::Configuration::spare2

#### 7.19.2.31 spare3

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

### 7.19.2.32 spare4

**BUInt8** BMeasureApi::Configuration::spare4[3]

### 7.19.2.33 spare5

**BUInt8** BMeasureApi::Configuration::spare5[3]

### 7.19.2.34 version

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

The configuration version.

### 7.19.2.35 wifiAp0

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

### 7.19.2.36 wifiAp0Password

**BChar** BMeasureApi::Configuration::wifiAp0Password[32]

### 7.19.2.37 wifiMode

[WifiMode](#) BMeasureApi::Configuration::wifiMode

The wifi mode.

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

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

## 7.20 BMeasureApi::DataBlock Class Reference

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



## Static Public Member Functions

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

## Public Attributes

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

### 7.20.1 Member Function Documentation

#### 7.20.1.1 [getMembers\(\)](#)

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

### 7.20.2 Member Data Documentation

#### 7.20.2.1 [data](#)

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

#### 7.20.2.2 [numChannels](#)

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

The number of data channels.

### 7.20.2.3 numSamples

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

The number of samples.

### 7.20.2.4 sequence

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

The sequence number.

### 7.20.2.5 source

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

The source unit.

### 7.20.2.6 spare

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

### 7.20.2.7 status

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

### 7.20.2.8 time

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

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

## 7.20.2.9 type

`DataBlockType` BMeasureApi::DataBlock::type

The type of data block.

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

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

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

## Private Attributes

- **BString** [ofileName](#)
- **BString** [omode](#)
- **BString** [offormat](#)
- **BFile** [ofile](#)
- **BUInt32** [opacketLen](#)
- **BoapMc1Packet** \* [opacket](#)

## 7.21.1 Constructor & Destructor Documentation

### 7.21.1.1 DataFile()

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

### 7.21.1.2 ~DataFile()

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

## 7.21.2 Member Function Documentation

### 7.21.2.1 close()

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

Close the file.

### 7.21.2.2 getFileName()

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

Return the file name.

### 7.21.2.3 init()

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

Initialise.

#### 7.21.2.4 open()

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

Open the file for read or write.

#### 7.21.2.5 readData()

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

Read a block of data.

#### 7.21.2.6 readInfo()

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

#### 7.21.2.7 validateFormat()

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

#### 7.21.2.8 writeData()

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

Write a block of data.

#### 7.21.2.9 writeEnd()

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

#### 7.21.2.10 writeInfo()

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

#### 7.21.2.11 writeInfoBMeas()

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

#### 7.21.2.12 writeInfoCsv()

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

#### 7.21.2.13 writeInfoTdms()

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

### 7.21.3 Member Data Documentation

#### 7.21.3.1 ofile

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

#### 7.21.3.2 ofileName

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

## 7.21.3.3 oformat

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

## 7.21.3.4 omode

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

## 7.21.3.5 opacket

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

## 7.21.3.6 opacketLen

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

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

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

## 7.22 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](#)

### 7.22.1 Detailed Description

The [Dfu](#) access class.

### 7.22.2 Constructor & Destructor Documentation

#### 7.22.2.1 Dfu()

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

#### 7.22.2.2 ~Dfu()

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

### 7.22.3 Member Function Documentation

#### 7.22.3.1 clearStatus()

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

#### 7.22.3.2 connect()

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

Connect to USB DFU device.



### 7.22.3.3 detectDevice()

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

Check if DFU devuce exists.

### 7.22.3.4 disconnect()

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

Disconnect from USB DFU device.

### 7.22.3.5 getStatus()

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

### 7.22.3.6 init()

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

Initialise.

### 7.22.3.7 reset()

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

Reset.

### 7.22.3.8 upload()

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

Upload a file.

### 7.22.3.9 upload\_cmd()

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

### 7.22.3.10 validateFile()

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

Check if file is valid firmware.

## 7.22.4 Member Data Documentation

### 7.22.4.1 oconnected

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

### 7.22.4.2 ocontext

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

### 7.22.4.3 odev

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

### 7.22.4.4 overbose

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

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

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

## 7.23 DfuStatus Struct Reference

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

### Public Attributes

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

### 7.23.1 Member Data Documentation

#### 7.23.1.1 iString

```
BUInt8 DfuStatus::iString
```

#### 7.23.1.2 pollTimeout

```
BUInt DfuStatus::pollTimeout
```

#### 7.23.1.3 state

```
BUInt8 DfuStatus::state
```

#### 7.23.1.4 status

```
BUInt8 DfuStatus::status
```

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

- [Dfu.h](#)

## 7.24 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]

## 7.24.1 Member Function Documentation

### 7.24.1.1 getMembers()

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

## 7.24.2 Member Data Documentation

### 7.24.2.1 data

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

### 7.24.2.2 length

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

The data length.

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

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

## 7.25 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.*

## 7.25.1 Member Function Documentation

### 7.25.1.1 getMembers()

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

## 7.25.2 Member Data Documentation

### 7.25.2.1 fileLength

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

The file length.

### 7.25.2.2 fileType

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

The file type.

### 7.25.2.3 name

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

### 7.25.2.4 spare

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

### 7.25.2.5 time

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

The file date/time.

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

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

## 7.26 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.*

### 7.26.1 Member Function Documentation

### 7.26.1.1 getMembers()

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

## 7.26.2 Member Data Documentation

### 7.26.2.1 free

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

The store free space.

### 7.26.2.2 name

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

### 7.26.2.3 size

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

The store size.

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

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

## 7.27 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.*

## 7.27.1 Member Function Documentation

### 7.27.1.1 [getMembers\(\)](#)

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

## 7.27.2 Member Data Documentation

### 7.27.2.1 [location](#)

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

### 7.27.2.2 [measureConfig](#)

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

The measurement configuration.



### 7.27.2.3 name

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

### 7.27.2.4 nodeInfo

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

[Information](#) on the unit.

### 7.27.2.5 numChannels

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

The number of data channels.

### 7.27.2.6 source

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

The source unit.

### 7.27.2.7 spare0

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

### 7.27.2.8 time

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

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

### 7.27.2.9 version

**BUInt16** BMeasureApi::InfoBlock::version

The info/data version.

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

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

## 7.28 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** [networkNameServer0](#)  
*The network nameserver.*
- **BUInt32** [networkTimeServer](#)  
*The network time server.*
- **BUInt8** [spare1](#) [32]

### 7.28.1 Member Function Documentation

### 7.28.1.1 getMembers()

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

## 7.28.2 Member Data Documentation

### 7.28.2.1 networkAddress

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

The network IP address.

### 7.28.2.2 networkGateway

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

The network gateway.

### 7.28.2.3 networkMask

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

The network netmask.

### 7.28.2.4 networkMode

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

The network Mode.

### 7.28.2.5 networkNameServer0

```
BUInt32 BMeasureApi::Information::networkNameServer0
```

The network nameserver.

#### 7.28.2.6 networkTimeServer

**BUInt32** BMeasureApi::Information::networkTimeServer

The network time server.

#### 7.28.2.7 nodeInfo

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

#### 7.28.2.8 numChannels

**BUInt8** BMeasureApi::Information::numChannels

The number of channels.

#### 7.28.2.9 numConfigItems

**BUInt8** BMeasureApi::Information::numConfigItems

The number of config items.

#### 7.28.2.10 spare0

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

#### 7.28.2.11 spare1

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

## 7.28.2.12 time

```
BTimeUs BMeasureApi::Information::time
```

The system time.

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

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

## 7.29 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]

### 7.29.1 Member Function Documentation

#### 7.29.1.1 getMembers()

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

## 7.29.2 Member Data Documentation

### 7.29.2.1 description

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

### 7.29.2.2 measureMode

**MeasureMode** BMeasureApi::MeasurementConfig::measureMode

### 7.29.2.3 measurePeriod

**BUInt32** BMeasureApi::MeasurementConfig::measurePeriod

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

### 7.29.2.4 numSamples0

**BUInt32** BMeasureApi::MeasurementConfig::numSamples0

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

### 7.29.2.5 numSamples1

**BUInt32** BMeasureApi::MeasurementConfig::numSamples1

The number of samples to capture. 0 is continuous.

### 7.29.2.6 numSamplesBlock

**BUInt32** BMeasureApi::MeasurementConfig::numSamplesBlock

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

### 7.29.2.7 sampleRate

**BFloat64** BMeasureApi::MeasurementConfig::sampleRate

### 7.29.2.8 triggerChannel

**BUInt8** BMeasureApi::MeasurementConfig::triggerChannel

### 7.29.2.9 triggerConfig

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

Trigger config including direction, filters etc.

### 7.29.2.10 triggerDelay

**BInt32** BMeasureApi::MeasurementConfig::triggerDelay

Trigger delay in samples.

### 7.29.2.11 triggerLevel

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

### 7.29.2.12 triggerMode

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

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

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

## 7.30 BMeasureApi::NodeInfo Class Reference

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

## Static Public Member Functions

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

## Public Attributes

- **BUInt32** [apiVersion](#)
- [Version](#) [hardwareVersion](#)
- [Version](#) [fpgaVersion](#)
- [Version](#) [wifiVersion](#)
- [Version](#) [softwareVersion](#)
- **BChar** [serialNumber](#) [12]

## 7.30.1 Member Function Documentation

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

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

## 7.30.2 Member Data Documentation

### 7.30.2.1 [apiVersion](#)

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

### 7.30.2.2 [fpgaVersion](#)

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

### 7.30.2.3 [hardwareVersion](#)

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



#### 7.30.2.4 serialNumber

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

#### 7.30.2.5 softwareVersion

**Version** BMeasureApi::NodeInfo::softwareVersion

#### 7.30.2.6 wifiVersion

**Version** BMeasureApi::NodeInfo::wifiVersion

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

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

## 7.31 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]

### 7.31.1 Member Function Documentation

#### 7.31.1.1 getMembers()

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

## 7.31.2 Member Data Documentation

### 7.31.2.1 error

**BUInt32** BMeasureApi::NodeStatus::error

### 7.31.2.2 errorStr

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

### 7.31.2.3 mode

[Mode](#) BMeasureApi::NodeStatus::mode

### 7.31.2.4 spare

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

### 7.31.2.5 status

**BUInt32** BMeasureApi::NodeStatus::status

### 7.31.2.6 time

**BTimeUs** BMeasureApi::NodeStatus::time

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

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

## 7.32 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](#)

### 7.32.1 Member Function Documentation

#### 7.32.1.1 [getMembers\(\)](#)

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

### 7.32.2 Member Data Documentation

#### 7.32.2.1 [type](#)

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

#### 7.32.2.2 [ver0](#)

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

#### 7.32.2.3 [ver1](#)

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

#### 7.32.2.4 [ver2](#)

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

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

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



## Chapter 8

# File Documentation

### 8.1 BMdns.cpp File Reference

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

#### Macros

- `#define BDEBUGL1 0`

#### Enumerations

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

#### Functions

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

## 8.1.1 Macro Definition Documentation

### 8.1.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

## 8.1.2 Enumeration Type Documentation

### 8.1.2.1 MdnsClass

```
enum MdnsClass
```

Enumerator

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

### 8.1.2.2 MdnsEntryType

```
enum MdnsEntryType
```

Enumerator

MDNS_ENTRYTYPE_ANSWER	
MDNS_ENTRYTYPE_AUTHORITY	
MDNS_ENTRYTYPE_ADDITIONAL	

### 8.1.2.3 MdnsRecordType

```
enum MdnsRecordType
```

Enumerator

MDNS_RECORDTYPE_IGNORE	
MDNS_RECORDTYPE_A	
MDNS_RECORDTYPE_PTR	
MDNS_RECORDTYPE_TXT	
MDNS_RECORDTYPE_AAAA	
MDNS_RECORDTYPE_SRV	

## 8.1.3 Function Documentation

### 8.1.3.1 mdns\_read\_string()

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

### 8.1.3.2 mdns\_read\_strings()

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

### 8.1.3.3 mdns\_write\_string()

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

## 8.2 BMdns.h File Reference

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

### Classes

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

## 8.3 BMeasureB-1.cpp File Reference

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

## Namespaces

- [BMeasureApi](#)

## 8.4 BMeasureB.cpp File Reference

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

## Namespaces

- [BMeasureApi](#)

## 8.5 BMeasureB.h File Reference

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

## Classes

- class [BMeasureApi::BMeasure](#)

## Namespaces

- [BMeasureApi](#)

## Variables

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

## 8.6 BMeasureD.cpp File Reference

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

## Namespaces

- [BMeasureApi](#)



## Macros

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

### 8.6.1 Macro Definition Documentation

#### 8.6.1.1 boffsetof

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

## 8.7 BMeasureD.h File Reference

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

## Classes

- class [BMeasureApi::Version](#)
- class [BMeasureApi::NodeInfo](#)
- class [BMeasureApi::NodeStatus](#)
- class [BMeasureApi::BoardConfig](#)
- class [BMeasureApi::ChannelConfig](#)
- class [BMeasureApi::Information](#)
- class [BMeasureApi::AlarmConfig](#)
- class [BMeasureApi::Configuration](#)
- class [BMeasureApi::ConfigItem](#)
- class [BMeasureApi::MeasurementConfig](#)
- class [BMeasureApi::DataBlock](#)
- class [BMeasureApi::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, `BMeasureApi::ErrorToFast` = 66 }
- 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, `BMeasureApi::StatusAlarm` = 0x0100 }
- 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::File Type` { `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,  
[BMeasureApi::CalibrateStageAttenScaling](#) = 8, [BMeasureApi::CalibrationStageFiveVolts](#) = 9 }
- enum [BMeasureApi::MessageSource](#) {  
[BMeasureApi::MessageSourceGeneral](#) = 0, [BMeasureApi::MessageSourceDebug](#) = 1, [BMeasureApi::MessageSourceTest](#)  
= 2, [BMeasureApi::MessageSourceWifi](#) = 3,  
[BMeasureApi::MessageSourceWifiTest](#) = 4 }
- enum [BMeasureApi::NetworkMode](#) { [BMeasureApi::NetworkModeOff](#) = 0, [BMeasureApi::NetworkModeDhcp](#)  
= 1, [BMeasureApi::NetworkModeManual](#) = 2 }
- enum [BMeasureApi::WifiMode](#) { [BMeasureApi::WifiModeOff](#), [BMeasureApi::WifiModeClient](#), [BMeasureApi::WifiModeAp](#)  
}
- enum [BMeasureApi::AlarmMode](#) { [BMeasureApi::AlarmModeOff](#), [BMeasureApi::AlarmModeHigher](#),  
[BMeasureApi::AlarmModeLower](#) }
- enum [BMeasureApi::AlarmOutput](#) {  
[BMeasureApi::AlarmOutputOff](#), [BMeasureApi::AlarmOutputDioHigh](#), [BMeasureApi::AlarmOutputDioLow](#),  
[BMeasureApi::AlarmOutputRelayOn](#),  
[BMeasureApi::AlarmOutputRelayOff](#) }
- enum [BMeasureApi::EventMode](#) { [BMeasureApi::EventModeOff](#), [BMeasureApi::EventModeAlarm](#),  
[BMeasureApi::EventModeSecond](#) }
- enum [BMeasureApi::Rs485Mode](#) { [BMeasureApi::Rs485ModeOff](#), [BMeasureApi::Rs485ModeBoap](#) }

## 8.8 BMeasureLib.cpp File Reference

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

### Namespaces

- [BMeasureApi](#)

### Macros

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

### 8.8.1 Macro Definition Documentation

#### 8.8.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

### 8.8.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

## 8.9 BMeasureLib.h File Reference

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

### Namespaces

- [BMeasureApi](#)

### Typedefs

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

## 8.10 BMeasureS.cpp File Reference

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

### Namespaces

- [BMeasureApi](#)

## 8.11 BMeasureUnit.cpp File Reference

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

### Namespaces

- [BMeasureApi](#)

## Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0
- #define [BDEBUGL3](#) 0
- #define [CONVERT\\_FLOAT](#) 0

*Convert to floating point.*

## Functions

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

### 8.11.1 Macro Definition Documentation

#### 8.11.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

#### 8.11.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

#### 8.11.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

#### 8.11.1.4 CONVERT\_FLOAT

```
#define CONVERT_FLOAT 0
```

Convert to floating point.

## 8.12 BMeasureUnit.h File Reference

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

### Classes

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

### Namespaces

- [BMeasureApi](#)

### Functions

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

## 8.13 BMeasureUnits.cpp File Reference

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

### Namespaces

- [BMeasureApi](#)

### Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0
- #define [BDEBUGL3](#) 0

### Functions

- static int [BMeasureApi::unitSort](#) (BMeasureUnit1 \*&u1, BMeasureUnit1 \*&u2)

### 8.13.1 Macro Definition Documentation

### 8.13.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

### 8.13.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

### 8.13.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

## 8.14 BMeasureUnits.h File Reference

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

### Classes

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

### Namespaces

- [BMeasureApi](#)

## 8.15 CommsNet.cpp File Reference

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

### Namespaces

- [BMeasureApi](#)

## Macros

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

### 8.15.1 Macro Definition Documentation

#### 8.15.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

#### 8.15.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

#### 8.15.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

## 8.16 CommsNet.h File Reference

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

## Classes

- class [BMeasureApi::CommsNet](#)

## Namespaces

- [BMeasureApi](#)



## 8.17 CommsSerial.cpp File Reference

## 8.18 CommsSerial.h File Reference

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

### Classes

- class [BMeasureApi::CommsSerial](#)

### Namespaces

- [BMeasureApi](#)

## 8.19 CommsUsb.cpp File Reference

```
#include <CommsUsb.h>  
#include <BSys.h>  
#include <libusb-1.0/libusb.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <BDebug.h>
```

### Namespaces

- [BMeasureApi](#)

### Macros

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

### Functions

- static **BUInt32** [BMeasureApi::roundDown512](#) ( **BUInt32** size)

### 8.19.1 Macro Definition Documentation

### 8.19.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

### 8.19.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

## 8.20 CommsUsb.h File Reference

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

### Classes

- class [BMeasureApi::CommsUsb](#)

### Namespaces

- [BMeasureApi](#)

## 8.21 DataFile.cpp File Reference

```
#include <DataFile.h>  
#include <BoapMcl.h>  
#include <BBuffer.h>  
#include <BDebug.h>  
#include <errno.h>
```

### Namespaces

- [BMeasureApi](#)

### Macros

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

## Enumerations

- enum `BMeasureApi::TdsDataType` {  
`BMeasureApi::TdsTypeVoid`, `BMeasureApi::TdsTypeI8`, `BMeasureApi::TdsTypeI16`, `BMeasureApi::TdsTypeI32`,  
`BMeasureApi::TdsTypeI64`, `BMeasureApi::TdsTypeU8`, `BMeasureApi::TdsTypeU16`, `BMeasureApi::TdsTypeU32`,  
`BMeasureApi::TdsTypeU64`, `BMeasureApi::TdsTypeSingleFloat`, `BMeasureApi::TdsTypeDoubleFloat`,  
`BMeasureApi::TdsTypeExtendedFloat`,  
`BMeasureApi::TdsTypeSingleFloatWithUnit =0x19`, `BMeasureApi::TdsTypeDoubleFloatWithUnit`, `BMeasureApi::TdsTypeExtendedFloatWithUnit =0x1A`,  
`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` )

### 8.21.1 Macro Definition Documentation

#### 8.21.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

#### 8.21.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

## 8.22 DataFile.h File Reference

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

## Classes

- class [BMeasureApi::DataFile](#)

## Namespaces

- [BMeasureApi](#)

## 8.23 Dfu.cpp File Reference

```
#include <Dfu.h>
#include <BFile.h>
#include <BDebug.h>
#include <unistd.h>
```

## Classes

- struct [BFirmwareInfo](#)

## Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0
- #define [STATE\\_APP\\_IDLE](#) 0x00
- #define [STATE\\_APP\\_DETACH](#) 0x01
- #define [STATE\\_DFU\\_IDLE](#) 0x02
- #define [STATE\\_DFU\\_DOWNLOAD\\_SYNC](#) 0x03
- #define [STATE\\_DFU\\_DOWNLOAD\\_BUSY](#) 0x04
- #define [STATE\\_DFU\\_DOWNLOAD\\_IDLE](#) 0x05
- #define [STATE\\_DFU\\_MANIFEST\\_SYNC](#) 0x06
- #define [STATE\\_DFU\\_MANIFEST](#) 0x07
- #define [STATE\\_DFU\\_MANIFEST\\_WAIT\\_RESET](#) 0x08
- #define [STATE\\_DFU\\_UPLOAD\\_IDLE](#) 0x09
- #define [STATE\\_DFU\\_ERROR](#) 0x0a
- #define [DFU\\_STATUS\\_OK](#) 0x00
- #define [DFU\\_STATUS\\_ERROR\\_TARGET](#) 0x01
- #define [DFU\\_STATUS\\_ERROR\\_FILE](#) 0x02
- #define [DFU\\_STATUS\\_ERROR\\_WRITE](#) 0x03
- #define [DFU\\_STATUS\\_ERROR\\_ERASE](#) 0x04
- #define [DFU\\_STATUS\\_ERROR\\_CHECK\\_ERASED](#) 0x05
- #define [DFU\\_STATUS\\_ERROR\\_PROG](#) 0x06
- #define [DFU\\_STATUS\\_ERROR\\_VERIFY](#) 0x07
- #define [DFU\\_STATUS\\_ERROR\\_ADDRESS](#) 0x08
- #define [DFU\\_STATUS\\_ERROR\\_NOTDONE](#) 0x09
- #define [DFU\\_STATUS\\_ERROR\\_FIRMWARE](#) 0x0a
- #define [DFU\\_STATUS\\_ERROR\\_VENDOR](#) 0x0b
- #define [DFU\\_STATUS\\_ERROR\\_USBR](#) 0x0c
- #define [DFU\\_STATUS\\_ERROR\\_POR](#) 0x0d
- #define [DFU\\_STATUS\\_ERROR\\_UNKNOWN](#) 0x0e

- `#define DFU_STATUS_ERROR_STALLEDPKT 0x0f`
- `#define DFU_DETACH 0`
- `#define DFU_DNLOAD 1`
- `#define DFU_UPLOAD 2`
- `#define DFU_GETSTATUS 3`
- `#define DFU_CLRSTATUS 4`
- `#define DFU_GETSTATE 5`
- `#define DFU_ABORT 6`
- `#define DFU_IFF_DFU 0x0001 /* DFU Mode, (not Runtime) */`
- `#define DFU_IFF_VENDOR 0x0100`
- `#define DFU_IFF_PRODUCT 0x0200`
- `#define DFU_IFF_CONFIG 0x0400`
- `#define DFU_IFF_IFACE 0x0800`
- `#define DFU_IFF_ALT 0x1000`
- `#define DFU_IFF_DEVNUM 0x2000`
- `#define DFU_IFF_PATH 0x4000`

## Enumerations

- `enum dfuse_command { SET_ADDRESS, ERASE_PAGE, MASS_ERASE, READ_UNPROTECT }`

## Functions

- `static BInt32 pageNumber ( BUInt32 address)`
- `static BUInt32 pageAddress ( BUInt32 page)`

## Variables

- `const BUInt32 BFirmwareInfoMagic = 0xBBEAA00`
- `const BUInt8 BFirmwareInfoEncrypt1 = 0x40`

### 8.23.1 Macro Definition Documentation

#### 8.23.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

#### 8.23.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

**8.23.1.3 DFU\_ABORT**

```
#define DFU_ABORT 6
```

**8.23.1.4 DFU\_CLRSTATUS**

```
#define DFU_CLRSTATUS 4
```

**8.23.1.5 DFU\_DETACH**

```
#define DFU_DETACH 0
```

**8.23.1.6 DFU\_DNLOAD**

```
#define DFU_DNLOAD 1
```

**8.23.1.7 DFU\_GETSTATE**

```
#define DFU_GETSTATE 5
```

**8.23.1.8 DFU\_GETSTATUS**

```
#define DFU_GETSTATUS 3
```

**8.23.1.9 DFU\_IFF\_ALT**

```
#define DFU_IFF_ALT 0x1000
```

**8.23.1.10 DFU\_IFF\_CONFIG**

```
#define DFU_IFF_CONFIG 0x0400
```

**8.23.1.11 DFU\_IFF\_DEVNUM**

```
#define DFU_IFF_DEVNUM 0x2000
```

**8.23.1.12 DFU\_IFF\_DFU**

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

**8.23.1.13 DFU\_IFF\_IFACE**

```
#define DFU_IFF_IFACE 0x0800
```

**8.23.1.14 DFU\_IFF\_PATH**

```
#define DFU_IFF_PATH 0x4000
```

**8.23.1.15 DFU\_IFF\_PRODUCT**

```
#define DFU_IFF_PRODUCT 0x0200
```

**8.23.1.16 DFU\_IFF\_VENDOR**

```
#define DFU_IFF_VENDOR 0x0100
```

**8.23.1.17 DFU\_STATUS\_ERROR\_ADDRESS**

```
#define DFU_STATUS_ERROR_ADDRESS 0x08
```

**8.23.1.18 DFU\_STATUS\_ERROR\_CHECK\_ERASED**

```
#define DFU_STATUS_ERROR_CHECK_ERASED 0x05
```

**8.23.1.19 DFU\_STATUS\_ERROR\_ERASE**

```
#define DFU_STATUS_ERROR_ERASE 0x04
```

**8.23.1.20 DFU\_STATUS\_ERROR\_FILE**

```
#define DFU_STATUS_ERROR_FILE 0x02
```

**8.23.1.21 DFU\_STATUS\_ERROR\_FIRMWARE**

```
#define DFU_STATUS_ERROR_FIRMWARE 0x0a
```

**8.23.1.22 DFU\_STATUS\_ERROR\_NOTDONE**

```
#define DFU_STATUS_ERROR_NOTDONE 0x09
```

**8.23.1.23 DFU\_STATUS\_ERROR\_POR**

```
#define DFU_STATUS_ERROR_POR 0x0d
```

**8.23.1.24 DFU\_STATUS\_ERROR\_PROG**

```
#define DFU_STATUS_ERROR_PROG 0x06
```

**8.23.1.25 DFU\_STATUS\_ERROR\_STALLEDPKT**

```
#define DFU_STATUS_ERROR_STALLEDPKT 0x0f
```

**8.23.1.26 DFU\_STATUS\_ERROR\_TARGET**

```
#define DFU_STATUS_ERROR_TARGET 0x01
```



**8.23.1.27 DFU\_STATUS\_ERROR\_UNKNOWN**

```
#define DFU_STATUS_ERROR_UNKNOWN 0x0e
```

**8.23.1.28 DFU\_STATUS\_ERROR\_USBR**

```
#define DFU_STATUS_ERROR_USBR 0x0c
```

**8.23.1.29 DFU\_STATUS\_ERROR\_VENDOR**

```
#define DFU_STATUS_ERROR_VENDOR 0x0b
```

**8.23.1.30 DFU\_STATUS\_ERROR\_VERIFY**

```
#define DFU_STATUS_ERROR_VERIFY 0x07
```

**8.23.1.31 DFU\_STATUS\_ERROR\_WRITE**

```
#define DFU_STATUS_ERROR_WRITE 0x03
```

**8.23.1.32 DFU\_STATUS\_OK**

```
#define DFU_STATUS_OK 0x00
```

**8.23.1.33 DFU\_UPLOAD**

```
#define DFU_UPLOAD 2
```

**8.23.1.34 STATE\_APP\_DETACH**

```
#define STATE_APP_DETACH 0x01
```

**8.23.1.35 STATE\_APP\_IDLE**

```
#define STATE_APP_IDLE 0x00
```

**8.23.1.36 STATE\_DFU\_DOWNLOAD\_BUSY**

```
#define STATE_DFU_DOWNLOAD_BUSY 0x04
```

**8.23.1.37 STATE\_DFU\_DOWNLOAD\_IDLE**

```
#define STATE_DFU_DOWNLOAD_IDLE 0x05
```

**8.23.1.38 STATE\_DFU\_DOWNLOAD\_SYNC**

```
#define STATE_DFU_DOWNLOAD_SYNC 0x03
```

**8.23.1.39 STATE\_DFU\_ERROR**

```
#define STATE_DFU_ERROR 0x0a
```

**8.23.1.40 STATE\_DFU\_IDLE**

```
#define STATE_DFU_IDLE 0x02
```

**8.23.1.41 STATE\_DFU\_MANIFEST**

```
#define STATE_DFU_MANIFEST 0x07
```

**8.23.1.42 STATE\_DFU\_MANIFEST\_SYNC**

```
#define STATE_DFU_MANIFEST_SYNC 0x06
```

#### 8.23.1.43 STATE\_DFU\_MANIFEST\_WAIT\_RESET

```
#define STATE_DFU_MANIFEST_WAIT_RESET 0x08
```

#### 8.23.1.44 STATE\_DFU\_UPLOAD\_IDLE

```
#define STATE_DFU_UPLOAD_IDLE 0x09
```

### 8.23.2 Enumeration Type Documentation

#### 8.23.2.1 dfuse\_command

```
enum dfuse_command
```

##### Enumerator

SET_ADDRESS	
ERASE_PAGE	
MASS_ERASE	
READ_UNPROTECT	

### 8.23.3 Function Documentation

#### 8.23.3.1 pageAddress()

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

#### 8.23.3.2 pageNumber()

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

### 8.23.4 Variable Documentation

#### 8.23.4.1 BFirmwareInfoEncrypt1

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

#### 8.23.4.2 BFirmwareInfoMagic

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

## 8.24 Dfu.h File Reference

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

### Classes

- struct [DfuStatus](#)
- class [Dfu](#)

*The [Dfu](#) access class.*

## 8.25 overview.dox File Reference

# Index

- ~BMdns
  - BMdns, [35](#)
- ~BMeasureUnit
  - BMeasureApi::BMeasureUnit, [57](#)
- ~BMeasureUnits
  - BMeasureApi::BMeasureUnits, [68](#)
- ~BMeasureUnitsDataBlock
  - BMeasureApi::BMeasureUnitsDataBlock, [79](#)
- ~CommsNet
  - BMeasureApi::CommsNet, [89](#)
- ~CommsSerial
  - BMeasureApi::CommsSerial, [92](#)
- ~CommsUsb
  - BMeasureApi::CommsUsb, [95](#)
- ~DataFile
  - BMeasureApi::DataFile, [110](#)
- ~Dfu
  - Dfu, [114](#)
- address
  - BMdnsService, [36](#)
- AlarmMode
  - BMeasureApi, [17](#)
- AlarmModeHigher
  - BMeasureApi, [18](#)
- AlarmModeLower
  - BMeasureApi, [18](#)
- AlarmModeOff
  - BMeasureApi, [18](#)
- AlarmOutput
  - BMeasureApi, [18](#)
- AlarmOutputDioHigh
  - BMeasureApi, [18](#)
- AlarmOutputDioLow
  - BMeasureApi, [18](#)
- AlarmOutputOff
  - BMeasureApi, [18](#)
- AlarmOutputRelayOff
  - BMeasureApi, [18](#)
- AlarmOutputRelayOn
  - BMeasureApi, [18](#)
- alarms
  - BMeasureApi::Configuration, [101](#)
- amplitude
  - BMeasureApi::AwgConfig, [31](#)
- apiVersion
  - BMeasureApi, [28](#)
  - BMeasureApi::NodeInfo, [130](#)
- attenuator
  - BMeasureApi::ChannelConfig, [86](#)
- AwgOutput
  - BMeasureApi, [18](#)
- AwgOutputAO0
  - BMeasureApi, [18](#)
- AwgOutputAO01
  - BMeasureApi, [18](#)
- AwgOutputAO1
  - BMeasureApi, [18](#)
- AwgOutputNone
  - BMeasureApi, [18](#)
- BDEBUGL1
  - BMdns.cpp, [136](#)
  - BMeasureLib.cpp, [141](#)
  - BMeasureUnit.cpp, [143](#)
  - BMeasureUnits.cpp, [144](#)
  - CommsNet.cpp, [146](#)
  - CommsUsb.cpp, [147](#)
  - DataFile.cpp, [149](#)
  - Dfu.cpp, [151](#)
- BDEBUGL2
  - BMeasureLib.cpp, [141](#)
  - BMeasureUnit.cpp, [143](#)
  - BMeasureUnits.cpp, [145](#)
  - CommsNet.cpp, [146](#)
  - CommsUsb.cpp, [148](#)
  - DataFile.cpp, [149](#)
  - Dfu.cpp, [151](#)
- BDEBUGL3
  - BMeasureUnit.cpp, [143](#)
  - BMeasureUnits.cpp, [145](#)
  - CommsNet.cpp, [146](#)
- BFirmwareInfo, [33](#)
  - checksum, [33](#)
  - length, [33](#)
  - magic, [33](#)
  - type, [33](#)
  - ver0, [34](#)
  - ver1, [34](#)
  - ver2, [34](#)
- BFirmwareInfoEncrypt1
  - Dfu.cpp, [157](#)
- BFirmwareInfoMagic
  - Dfu.cpp, [158](#)
- blockNumChannels
  - BMeasureApi::BMeasureUnit, [60](#)
- blockNumSamples
  - BMeasureApi::BMeasureUnit, [60](#)
- BlockTypeData
  - BMeasureApi, [18](#)

- BlockTypeInfo
  - BMeasureApi, 18
- BlockTypes
  - BMeasureApi, 18
- BMdns, 34
  - ~BMdns, 35
  - BMdns, 34
  - findServices, 35
  - init, 35
  - osocket, 35
  - otransactionId, 35
- BMdns.cpp, 135
  - BDEBUGL1, 136
  - MDNS\_CLASS\_IN, 136
  - MDNS\_ENTRYTYPE\_ADDITIONAL, 136
  - MDNS\_ENTRYTYPE\_ANSWER, 136
  - MDNS\_ENTRYTYPE\_AUTHORITY, 136
  - mdns\_read\_string, 137
  - mdns\_read\_strings, 137
  - MDNS\_RECORDTYPE\_A, 136
  - MDNS\_RECORDTYPE\_AAAA, 136
  - MDNS\_RECORDTYPE\_IGNORE, 136
  - MDNS\_RECORDTYPE\_PTR, 136
  - MDNS\_RECORDTYPE\_SRV, 136
  - MDNS\_RECORDTYPE\_TXT, 136
  - mdns\_write\_string, 137
  - MdnsClass, 136
  - MdnsEntryType, 136
  - MdnsRecordType, 136
- BMdns.h, 137
- BMdnsService, 36
  - address, 36
  - extra, 36
  - hostname, 36
  - name, 36
- BMeasure
  - BMeasureApi::BMeasure, 39
- BMeasureApi, 15
  - AlarmMode, 17
  - AlarmModeHigher, 18
  - AlarmModeLower, 18
  - AlarmModeOff, 18
  - AlarmOutput, 18
  - AlarmOutputDioHigh, 18
  - AlarmOutputDioLow, 18
  - AlarmOutputOff, 18
  - AlarmOutputRelayOff, 18
  - AlarmOutputRelayOn, 18
  - apiVersion, 28
  - AwgOutput, 18
  - AwgOutputAO0, 18
  - AwgOutputAO01, 18
  - AwgOutputAO1, 18
  - AwgOutputNone, 18
  - BlockTypeData, 18
  - BlockTypeInfo, 18
  - BlockTypes, 18
  - CalibrateStage, 18
  - CalibrateStageAdcOffsets, 19
  - CalibrateStageAdcScaling, 19
  - CalibrateStageAttenScaling, 19
  - CalibrateStageClear, 19
  - CalibrateStageDacOffsets, 19
  - CalibrateStageDacScaling0, 19
  - CalibrateStageDacScaling1, 19
  - CalibrateStageNone, 19
  - CalibrateStageSettle, 19
  - CalibrationStageFiveVolts, 19
  - ChannelConfigs, 17
  - ChannelType, 19
  - ChannelTypeAnalogueIn, 19
  - ChannelTypeAnalogueOut, 19
  - ChannelTypeDigitalIn, 19
  - ChannelTypeDigitalOut, 19
  - ChannelTypeNone, 19
  - channelTypeString, 26
  - DataBlockType, 19
  - DataBlockType125i, 19
  - DataBlockTypeFloat32, 19
  - DataSend, 19
  - DataSendOff, 20
  - DataSendOn, 20
  - DigitalInOut, 20
  - DigitalMode, 20
  - DigitalModeInput, 20
  - DigitalModeOutput, 20
  - DigitalModeSyncMaster, 20
  - DigitalModeSyncSlave, 20
  - ErrorDataOverrun, 20
  - ErrorNum, 20
  - ErrorSystem, 20
  - ErrorToFast, 20
  - EventMode, 20
  - EventModeAlarm, 20
  - EventModeOff, 20
  - EventModeSecond, 20
  - FilesysDeleteType, 20
  - FilesysDeleteTypeData, 21
  - FilesysDeleteTypeFormat, 21
  - FilesysDeleteTypeNone, 21
  - FileType, 21
  - FileTypeDir, 21
  - FileTypeFile, 21
  - FileTypeNone, 21
  - LogDataMode, 21
  - LogDataModeDeleteOld, 21
  - LogDataModeNormal, 21
  - MeasureMode, 21
  - MeasureModeContinuous, 21
  - MeasureModeOff, 21
  - MeasureModeOneShot, 21
  - MeasureModeRepeat, 21
  - MessageSource, 22
  - MessageSourceDebug, 22
  - MessageSourceGeneral, 22
  - MessageSourceTest, 22

- MessageSourceWifi, [22](#)
- MessageSourceWifiTest, [22](#)
- Mode, [22](#)
- ModeDemo1, [22](#)
- ModeIdle, [22](#)
- ModeInternal, [22](#)
- ModeRun, [22](#)
- ModeRunProgram, [22](#)
- ModeSleep, [22](#)
- NetworkMode, [22](#)
- NetworkModeDhcp, [22](#)
- NetworkModeManual, [22](#)
- NetworkModeOff, [22](#)
- NodeType, [22](#)
- NodeTypeBMeasure1, [23](#)
- NodeTypeNone, [23](#)
- round512, [26](#)
- roundDown512, [26](#)
- Rs485Mode, [23](#)
- Rs485ModeBoap, [23](#)
- Rs485ModeOff, [23](#)
- SampleType, [23](#)
- SampleTypeBool, [23](#)
- SampleTypeFloat32, [23](#)
- SampleTypeFloat64, [23](#)
- SampleTypeInt16, [23](#)
- SampleTypeInt32, [23](#)
- SampleTypeInt8, [23](#)
- SampleTypeNone, [23](#)
- sampleTypeString, [26](#)
- SecureMode, [23](#)
- SecureModeFull, [23](#)
- SecureModeOpen, [23](#)
- SecureMoteRemote, [23](#)
- Status, [23](#)
- StatusAlarm, [24](#)
- StatusDataOverrun, [24](#)
- StatusEnd0, [24](#)
- StatusEnd1, [24](#)
- StatusError, [24](#)
- StatusFpgaOverrun, [24](#)
- StatusNone, [24](#)
- StatusRun, [24](#)
- StatusTriggerWait, [24](#)
- StatusWarning, [24](#)
- SyncMode, [24](#)
- SyncModeMaster, [24](#)
- SyncModeOff, [24](#)
- SyncModeSlave, [24](#)
- TdsDataType, [24](#)
- TdsTypeBoolean, [25](#)
- TdsTypeComplexDoubleFloat, [25](#)
- TdsTypeComplexSingleFloat, [25](#)
- TdsTypeDAQmxRawData, [25](#)
- TdsTypeDoubleFloat, [24](#)
- TdsTypeDoubleFloatWithUnit, [25](#)
- TdsTypeExtendedFloat, [25](#)
- TdsTypeExtendedFloatWithUnit, [25](#)
- TdsTypeFixedPoint, [25](#)
- TdsTypeI16, [24](#)
- TdsTypeI32, [24](#)
- TdsTypeI64, [24](#)
- TdsTypeI8, [24](#)
- TdsTypeSingleFloat, [24](#)
- TdsTypeSingleFloatWithUnit, [25](#)
- TdsTypeString, [25](#)
- TdsTypeTimeStamp, [25](#)
- TdsTypeU16, [24](#)
- TdsTypeU32, [24](#)
- TdsTypeU64, [24](#)
- TdsTypeU8, [24](#)
- TdsTypeVoid, [24](#)
- TocBigEndian, [27](#)
- TocDaqRawData, [27](#)
- TocInterleavedData, [27](#)
- TocMetaData, [27](#)
- TocNewObjList, [27](#)
- TocRawData, [27](#)
- toFloat, [27](#)
- TriggerConfig, [25](#)
- TriggerConfigNone, [25](#)
- TriggerMode, [25](#)
- TriggerModeNegative, [25](#)
- TriggerModeOff, [25](#)
- TriggerModePositive, [25](#)
- unitSort, [28](#)
- Waveform, [25](#)
- WaveformArbitrary, [26](#)
- WaveformDc, [25](#)
- WaveformNoise, [26](#)
- WaveformNone, [25](#)
- WaveformSine, [26](#)
- WaveformSquare, [26](#)
- WaveformTriangle, [26](#)
- WifiMode, [26](#)
- WifiModeAp, [26](#)
- WifiModeClient, [26](#)
- WifiModeOff, [26](#)
- BMeasureApi::AlarmConfig, [29](#)
  - channel, [30](#)
  - getMembers, [29](#)
  - level, [30](#)
  - mode, [30](#)
  - output, [30](#)
  - outputChannel, [30](#)
- BMeasureApi::AwgConfig, [31](#)
  - amplitude, [31](#)
  - duty, [31](#)
  - frequency, [32](#)
  - getMembers, [31](#)
  - numSamples, [32](#)
  - offset, [32](#)
  - output, [32](#)
  - spare, [32](#)
  - waveform, [32](#)
- BMeasureApi::BMeasure, [37](#)

BMeasure, 39  
 calibrate, 40  
 calibrateServe, 40  
 factoryReset, 40  
 factoryResetServe, 40  
 fileClose, 40  
 fileCloseServe, 40  
 fileDelete, 41  
 fileDeleteServe, 41  
 fileList, 41  
 fileListServe, 41  
 fileOpen, 41  
 fileOpenServe, 41  
 fileRead, 42  
 fileReadServe, 42  
 fileSysDelete, 42  
 fileSysDeleteServe, 42  
 fileSysInfo, 42  
 fileSysInfoServe, 42  
 fileWrite, 43  
 fileWriteServe, 43  
 functionUnLock, 43  
 functionUnLockServe, 43  
 getAwgConfig, 43  
 getAwgConfigServe, 43  
 getBoardConfig, 44  
 getBoardConfigServe, 44  
 getChannelConfig, 44  
 getChannelConfigServe, 44  
 getConfig, 44  
 getConfigServe, 44  
 getDigital, 45  
 getDigitalServe, 45  
 getInfoBlock, 45  
 getInfoBlockServe, 45  
 getInformation, 45  
 getInformationServe, 45  
 getMeasurement, 46  
 getMeasurementConfig, 46  
 getMeasurementConfigServe, 46  
 getMeasurementServe, 46  
 getNodeInfo, 46  
 getNodeInfoServe, 46  
 getStatus, 47  
 getStatusServe, 47  
 getSwitch, 47  
 getSwitchServe, 47  
 login, 47  
 loginServe, 47  
 measure, 48  
 measureServe, 48  
 processRequest, 48  
 runBoardTest, 48  
 runBoardTestServe, 48  
 sendData, 48  
 sendDataEnable, 49  
 sendDataEnableServe, 49  
 sendDataServe, 49  
 sendInfo, 49  
 sendInfoServe, 49  
 sendMessage, 49  
 sendMessageServe, 50  
 sendStatus, 50  
 sendStatusServe, 50  
 sendTime, 50  
 sendTimeServe, 50  
 setAnalogueOut, 50  
 setAnalogueOutServe, 51  
 setAwgConfig, 51  
 setAwgConfigServe, 51  
 setAwgWaveform, 51  
 setAwgWaveformServe, 51  
 setBoardConfig, 51  
 setBoardConfigServe, 52  
 setChannelConfig, 52  
 setChannelConfigFull, 52  
 setChannelConfigFullServe, 52  
 setChannelConfigServe, 52  
 setConfig, 52  
 setConfigServe, 53  
 setDigital, 53  
 setDigitalServe, 53  
 setMeasurement, 53  
 setMeasurementConfig, 53  
 setMeasurementConfigServe, 53  
 setMeasurementServe, 54  
 setMode, 54  
 setModeServe, 54  
 setRelay, 54  
 setRelayServe, 54  
 setSecureKey, 54  
 setSecureKeyServe, 55  
 setSecureMode, 55  
 setSecureModeServe, 55  
 BMeasureApi::BMeasureUnit, 55  
 ~BMeasureUnit, 57  
 blockNumChannels, 60  
 blockNumSamples, 60  
 BMeasureUnit, 57  
 connect, 57  
 device, 57  
 disconnect, 57  
 disconnected, 57  
 findDevices, 58  
 findDevicesNetwork, 58  
 findDevicesUsb, 58  
 info, 58  
 numChannels, 58  
 oblockCount, 60  
 ochannels, 60  
 oconfigMeasurement, 60  
 odataBlock, 60  
 odevice, 60  
 odisconnecting, 61  
 oinfo, 61  
 onodeInfo, 61



- osampleCount, 61
- osequenceNext, 61
- processdataBlock, 58
- run, 59
- sendDataServe, 59
- sendDataServe1, 59
- serialNumber, 59
- setChannelConfig, 59
- setMeasurement, 59
- BMeasureApi::BMeasureUnit1, 62
  - BMeasureUnit1, 62
  - disconnected, 63
  - oconnected, 63
  - oenabled, 64
  - omeasureUnits, 64
  - oorder, 64
  - oserialNumber, 64
  - osource, 64
  - sendDataServe1, 63
  - sendMessageServe, 63
  - serialNumber, 63
  - setSerialNumber, 63
- BMeasureApi::BMeasureUnitDevice, 64
  - BMeasureUnitDevice, 65
  - device, 65
  - serialNumber, 65
- BMeasureApi::BMeasureUnits, 65
  - ~BMeasureUnits, 68
  - BMeasureUnits, 67
  - clear, 68
  - dataAvailable, 68
  - dataClear, 68
  - dataDone, 68
  - dataEvent, 68
  - dataProcessEnable, 68
  - dataRead, 69
  - dataSetNumStreams, 69
  - dataWait, 69
  - debugPrint, 69
  - disconnected, 69
  - getAwgConfig, 69
  - getChannelConfig, 70
  - getConfig, 70
  - getFreeBlock, 70
  - getInfoBlock, 70
  - getInformation, 70
  - getMeasurement, 70
  - getMeasurementConfig, 71
  - getStatus, 71
  - numChannels, 71
  - odataBlocksFree, 75
  - odataBlocksIn, 76
  - odataBlocksOut, 76
  - odataBlocksOutCount, 76
  - odataBlocksProcess, 76
  - odataBlocksProcessNum, 76
  - odataStreamNum, 76
  - ofill, 76
  - olocalTrigger, 76
  - oLockInput, 77
  - oLockOutput, 77
  - oLockUnits, 77
  - onumBlocks, 77
  - onumChannels, 77
  - onumConnected, 77
  - oprocEnable, 77
  - oprocRunning, 78
  - ostartSample, 78
  - otriggered, 78
  - ounitMaster, 78
  - ounits, 78
  - outputBlock, 71
  - run, 71
  - sendDataEnable, 71
  - sendDataProcess, 72
  - sendDataProcessTrigger, 72
  - sendDataQueue, 72
  - sendDataServe1, 72
  - sendMessage, 72
  - sendMessageServe, 72
  - sendTime, 72
  - setAwgConfig, 73
  - setChannelConfig, 73
  - setConfig, 73
  - setMeasurement, 73
  - setMeasurementConfig, 73
  - setMode, 73
  - unit, 74
  - unitAdd, 74
  - unitDelete, 74
  - unitMaster, 74
  - unitsConnect, 74
  - unitsConnected, 74
  - unitsConnectedNum, 74
  - unitsDisconnect, 75
  - unitSetEnabled, 75
  - unitSetOrder, 75
  - unitsFind, 75
  - unitsNum, 75
- BMeasureApi::BMeasureUnitsDataBlock, 78
  - ~BMeasureUnitsDataBlock, 79
  - BMeasureUnitsDataBlock, 79
  - init, 79
  - odataBlock, 79
  - ofill, 80
  - oinUse, 80
- BMeasureApi::BoardConfig, 80
  - buildTime, 81
  - calibAdcOffsets, 81
  - calibAdcScales, 81
  - calibAttenScales, 81
  - calibDacOffsets, 81
  - calibDacScales, 81
  - calibFiveVolts, 82
  - calibTemp, 82
  - calibTime, 82

- getMembers, 81
- hardwareVersion, 82
- macAddress, 82
- magic, 82
- serialNumber, 82
- spare0, 82
- testMode, 83
- BMeasureApi::CalibrateInfo, 83
  - calibrateAmplitude, 84
  - calibrateFrequency, 84
  - calibrateTime, 84
  - getMembers, 83
  - stage, 84
  - value, 84
- BMeasureApi::ChannelConfig, 85
  - attenuator, 86
  - calibOffset, 86
  - calibScale, 86
  - calibScaleAtten1, 86
  - dataChannel, 86
  - enabled, 86
  - getMembers, 85
  - id, 87
  - name, 87
  - number, 87
  - offset, 87
  - pgaGain, 87
  - process, 87
  - sampleType, 87
  - scale, 88
  - siUnits, 88
  - spare0, 88
  - type, 88
- BMeasureApi::CommsNet, 88
  - ~CommsNet, 89
  - CommsNet, 89
  - connect, 89
  - disconnect, 89
  - init, 90
  - oinWait, 91
  - osocket, 91
  - oterminating, 91
  - read, 90
  - readAvailable, 90
  - wait, 90
  - write, 90
  - writeAvailable, 90
  - writeChunks, 91
- BMeasureApi::CommsSerial, 92
  - ~CommsSerial, 92
  - CommsSerial, 92
  - connect, 93
  - disconnect, 93
  - odevice, 94
  - oserialPort, 94
  - read, 93
  - readAvailable, 93
  - wait, 93
  - write, 93
- BMeasureApi::CommsUsb, 94
  - ~CommsUsb, 95
  - CommsUsb, 95
  - connect, 95
  - disconnect, 95
  - obuffer, 96
  - ocontext, 97
  - odev, 97
  - odevice, 97
  - onum, 97
  - oterminated, 97
  - oterminating, 97
  - read, 95
  - readAvailable, 96
  - readChunk, 96
  - wait, 96
  - write, 96
- BMeasureApi::ConfigItem, 98
  - getMembers, 98
  - name, 98
  - spare, 98
  - type, 98
  - value, 99
- BMeasureApi::Configuration, 99
  - alarms, 101
  - digitalMode, 101
  - digitalPins, 101
  - emailAddress, 101
  - emailMode, 101
  - getMembers, 100
  - location, 101
  - logData, 101
  - logDataDevice, 102
  - logDataMode, 102
  - mode, 102
  - mqttMode, 102
  - mqttPort, 102
  - mqttServer, 102
  - name, 103
  - networkAddress, 103
  - networkGateway, 103
  - networkMask, 103
  - networkMode, 103
  - networkNameServer0, 103
  - networkTimeServer, 104
  - program, 104
  - rs485BaudRate, 104
  - rs485Bits, 104
  - rs485Mode, 104
  - rs485StopBits, 104
  - sampleFrequencyMode, 105
  - source, 105
  - spare0, 105
  - spare1, 105
  - spare2, 105
  - spare3, 105
  - spare4, 105

- spare5, 106
- version, 106
- wifiAp0, 106
- wifiAp0Password, 106
- wifiMode, 106
- BMeasureApi::DataBlock, 106
  - data, 107
  - getMembers, 107
  - numChannels, 107
  - numSamples, 107
  - sequence, 108
  - source, 108
  - spare, 108
  - status, 108
  - time, 108
  - type, 108
- BMeasureApi::DataFile, 109
  - ~DataFile, 110
  - close, 110
  - DataFile, 110
  - getFileName, 110
  - init, 110
  - ofile, 112
  - ofilename, 112
  - offormat, 112
  - omode, 113
  - opacket, 113
  - opacketLen, 113
  - open, 110
  - readData, 111
  - readInfo, 111
  - validateFormat, 111
  - writeData, 111
  - writeEnd, 111
  - writelnInfo, 111
  - writelnInfoBMeas, 112
  - writelnInfoCsv, 112
  - writelnInfoTdms, 112
- BMeasureApi::FileData, 117
  - data, 118
  - getMembers, 118
  - length, 118
- BMeasureApi::FileInfo, 118
  - fileLength, 119
  - fileType, 119
  - getMembers, 119
  - name, 119
  - spare, 120
  - time, 120
- BMeasureApi::FilesysInfo, 120
  - free, 121
  - getMembers, 120
  - name, 121
  - size, 121
- BMeasureApi::InfoBlock, 121
  - getMembers, 122
  - location, 122
  - measureConfig, 122
  - name, 122
  - nodeInfo, 123
  - numChannels, 123
  - source, 123
  - spare0, 123
  - time, 123
  - version, 123
- BMeasureApi::Information, 124
  - getMembers, 124
  - networkAddress, 125
  - networkGateway, 125
  - networkMask, 125
  - networkMode, 125
  - networkNameServer0, 125
  - networkTimeServer, 125
  - nodeInfo, 126
  - numChannels, 126
  - numConfigItems, 126
  - spare0, 126
  - spare1, 126
  - time, 126
- BMeasureApi::MeasurementConfig, 127
  - description, 128
  - getMembers, 127
  - measureMode, 128
  - measurePeriod, 128
  - numSamples0, 128
  - numSamples1, 128
  - numSamplesBlock, 128
  - sampleRate, 128
  - triggerChannel, 129
  - triggerConfig, 129
  - triggerDelay, 129
  - triggerLevel, 129
  - triggerMode, 129
- BMeasureApi::NodeInfo, 129
  - apiVersion, 130
  - fpgaVersion, 130
  - getMembers, 130
  - hardwareVersion, 130
  - serialNumber, 130
  - softwareVersion, 131
  - wifiVersion, 131
- BMeasureApi::NodeStatus, 131
  - error, 132
  - errorStr, 132
  - getMembers, 131
  - mode, 132
  - spare, 132
  - status, 132
  - time, 132
- BMeasureApi::Version, 133
  - getMembers, 133
  - type, 133
  - ver0, 133
  - ver1, 133
  - ver2, 133
- BMeasureB-1.cpp, 137

- BMeasureB.cpp, 138
- BMeasureB.h, 138
- BMeasureD.cpp, 138
  - boffsetof, 139
- BMeasureD.h, 139
- BMeasureLib.cpp, 141
  - BDEBUGL1, 141
  - BDEBUGL2, 141
- BMeasureLib.h, 142
- BMeasureS.cpp, 142
- BMeasureUnit
  - BMeasureApi::BMeasureUnit, 57
- BMeasureUnit.cpp, 142
  - BDEBUGL1, 143
  - BDEBUGL2, 143
  - BDEBUGL3, 143
  - CONVERT\_FLOAT, 143
- BMeasureUnit.h, 144
- BMeasureUnit1
  - BMeasureApi::BMeasureUnit1, 62
- BMeasureUnitDevice
  - BMeasureApi::BMeasureUnitDevice, 65
- BMeasureUnits
  - BMeasureApi::BMeasureUnits, 67
- BMeasureUnits.cpp, 144
  - BDEBUGL1, 144
  - BDEBUGL2, 145
  - BDEBUGL3, 145
- BMeasureUnits.h, 145
- BMeasureUnitsDataBlock
  - BMeasureApi::BMeasureUnitsDataBlock, 79
- boffsetof
  - BMeasureD.cpp, 139
- buildTime
  - BMeasureApi::BoardConfig, 81
- calibAdcOffsets
  - BMeasureApi::BoardConfig, 81
- calibAdcScales
  - BMeasureApi::BoardConfig, 81
- calibAttenScales
  - BMeasureApi::BoardConfig, 81
- calibDacOffsets
  - BMeasureApi::BoardConfig, 81
- calibDacScales
  - BMeasureApi::BoardConfig, 81
- calibFiveVolts
  - BMeasureApi::BoardConfig, 82
- calibOffset
  - BMeasureApi::ChannelConfig, 86
- calibrate
  - BMeasureApi::BMeasure, 40
- calibrateAmplitude
  - BMeasureApi::CalibrateInfo, 84
- calibrateFrequency
  - BMeasureApi::CalibrateInfo, 84
- calibrateServe
  - BMeasureApi::BMeasure, 40
- CalibrateStage
  - BMeasureApi, 18
  - CalibrateStageAdcOffsets
    - BMeasureApi, 19
  - CalibrateStageAdcScaling
    - BMeasureApi, 19
  - CalibrateStageAttenScaling
    - BMeasureApi, 19
  - CalibrateStageClear
    - BMeasureApi, 19
  - CalibrateStageDacOffsets
    - BMeasureApi, 19
  - CalibrateStageDacScaling0
    - BMeasureApi, 19
  - CalibrateStageDacScaling1
    - BMeasureApi, 19
  - CalibrateStageNone
    - BMeasureApi, 19
  - CalibrateStageSettle
    - BMeasureApi, 19
  - calibrateTime
    - BMeasureApi::CalibrateInfo, 84
  - CalibrationStageFiveVolts
    - BMeasureApi, 19
  - calibScale
    - BMeasureApi::ChannelConfig, 86
  - calibScaleAtten1
    - BMeasureApi::ChannelConfig, 86
  - calibTemp
    - BMeasureApi::BoardConfig, 82
  - calibTime
    - BMeasureApi::BoardConfig, 82
  - channel
    - BMeasureApi::AlarmConfig, 30
  - ChannelConfigs
    - BMeasureApi, 17
  - ChannelType
    - BMeasureApi, 19
  - ChannelTypeAnalogueIn
    - BMeasureApi, 19
  - ChannelTypeAnalogueOut
    - BMeasureApi, 19
  - ChannelTypeDigitalIn
    - BMeasureApi, 19
  - ChannelTypeDigitalOut
    - BMeasureApi, 19
  - ChannelTypeNone
    - BMeasureApi, 19
  - channelTypeString
    - BMeasureApi, 26
  - checksum
    - BFirmwareInfo, 33
  - clear
    - BMeasureApi::BMeasureUnits, 68
  - clearStatus
    - Dfu, 114
  - close
    - BMeasureApi::DataFile, 110
  - CommsNet

- BMeasureApi::CommsNet, [89](#)
- CommsNet.cpp, [145](#)
  - BDEBUGL1, [146](#)
  - BDEBUGL2, [146](#)
  - BDEBUGL3, [146](#)
- CommsNet.h, [146](#)
- CommsSerial
  - BMeasureApi::CommsSerial, [92](#)
- CommsSerial.cpp, [147](#)
- CommsSerial.h, [147](#)
- CommsUsb
  - BMeasureApi::CommsUsb, [95](#)
- CommsUsb.cpp, [147](#)
  - BDEBUGL1, [147](#)
  - BDEBUGL2, [148](#)
- CommsUsb.h, [148](#)
- connect
  - BMeasureApi::BMeasureUnit, [57](#)
  - BMeasureApi::CommsNet, [89](#)
  - BMeasureApi::CommsSerial, [93](#)
  - BMeasureApi::CommsUsb, [95](#)
  - Dfu, [114](#)
- CONVERT\_FLOAT
  - BMeasureUnit.cpp, [143](#)
- data
  - BMeasureApi::DataBlock, [107](#)
  - BMeasureApi::FileData, [118](#)
- dataAvailable
  - BMeasureApi::BMeasureUnits, [68](#)
- DataBlockType
  - BMeasureApi, [19](#)
- DataBlockType125i
  - BMeasureApi, [19](#)
- DataBlockTypeFloat32
  - BMeasureApi, [19](#)
- dataChannel
  - BMeasureApi::ChannelConfig, [86](#)
- dataClear
  - BMeasureApi::BMeasureUnits, [68](#)
- dataDone
  - BMeasureApi::BMeasureUnits, [68](#)
- dataEvent
  - BMeasureApi::BMeasureUnits, [68](#)
- DataFile
  - BMeasureApi::DataFile, [110](#)
- DataFile.cpp, [148](#)
  - BDEBUGL1, [149](#)
  - BDEBUGL2, [149](#)
- DataFile.h, [149](#)
- dataProcessEnable
  - BMeasureApi::BMeasureUnits, [68](#)
- dataRead
  - BMeasureApi::BMeasureUnits, [69](#)
- DataSend
  - BMeasureApi, [19](#)
- DataSendOff
  - BMeasureApi, [20](#)
- DataSendOn
  - BMeasureApi, [20](#)
- dataSetNumStreams
  - BMeasureApi::BMeasureUnits, [69](#)
- dataWait
  - BMeasureApi::BMeasureUnits, [69](#)
- debugPrint
  - BMeasureApi::BMeasureUnits, [69](#)
- description
  - BMeasureApi::MeasurementConfig, [128](#)
- detectDevice
  - Dfu, [114](#)
- device
  - BMeasureApi::BMeasureUnit, [57](#)
  - BMeasureApi::BMeasureUnitDevice, [65](#)
- Dfu, [113](#)
  - ~Dfu, [114](#)
  - clearStatus, [114](#)
  - connect, [114](#)
  - detectDevice, [114](#)
  - Dfu, [114](#)
  - disconnect, [115](#)
  - getStatus, [115](#)
  - init, [115](#)
  - oconnected, [116](#)
  - ocontext, [116](#)
  - odev, [116](#)
  - overbose, [116](#)
  - reset, [115](#)
  - upload, [115](#)
  - upload\_cmd, [115](#)
  - validateFile, [116](#)
- Dfu.cpp, [150](#)
  - BDEBUGL1, [151](#)
  - BDEBUGL2, [151](#)
  - BFirmwareInfoEncrypt1, [157](#)
  - BFirmwareInfoMagic, [158](#)
  - DFU\_ABORT, [151](#)
  - DFU\_CLRSTATUS, [152](#)
  - DFU\_DETACH, [152](#)
  - DFU\_DNLOAD, [152](#)
  - DFU\_GETSTATE, [152](#)
  - DFU\_GETSTATUS, [152](#)
  - DFU\_IFF\_ALT, [152](#)
  - DFU\_IFF\_CONFIG, [152](#)
  - DFU\_IFF\_DEVNUM, [152](#)
  - DFU\_IFF\_DFU, [153](#)
  - DFU\_IFF\_IFACE, [153](#)
  - DFU\_IFF\_PATH, [153](#)
  - DFU\_IFF\_PRODUCT, [153](#)
  - DFU\_IFF\_VENDOR, [153](#)
  - DFU\_STATUS\_ERROR\_ADDRESS, [153](#)
  - DFU\_STATUS\_ERROR\_CHECK\_ERASED, [153](#)
  - DFU\_STATUS\_ERROR\_ERASE, [153](#)
  - DFU\_STATUS\_ERROR\_FILE, [154](#)
  - DFU\_STATUS\_ERROR\_FIRMWARE, [154](#)
  - DFU\_STATUS\_ERROR\_NOTDONE, [154](#)
  - DFU\_STATUS\_ERROR\_POR, [154](#)
  - DFU\_STATUS\_ERROR\_PROG, [154](#)

- DFU\_STATUS\_ERROR\_STALLEDPKT, [154](#)
- DFU\_STATUS\_ERROR\_TARGET, [154](#)
- DFU\_STATUS\_ERROR\_UNKNOWN, [154](#)
- DFU\_STATUS\_ERROR\_USBR, [155](#)
- DFU\_STATUS\_ERROR\_VENDOR, [155](#)
- DFU\_STATUS\_ERROR\_VERIFY, [155](#)
- DFU\_STATUS\_ERROR\_WRITE, [155](#)
- DFU\_STATUS\_OK, [155](#)
- DFU\_UPLOAD, [155](#)
- dfuse\_command, [157](#)
- ERASE\_PAGE, [157](#)
- MASS\_ERASE, [157](#)
- pageAddress, [157](#)
- pageNumber, [157](#)
- READ\_UNPROTECT, [157](#)
- SET\_ADDRESS, [157](#)
- STATE\_APP\_DETACH, [155](#)
- STATE\_APP\_IDLE, [155](#)
- STATE\_DFU\_DOWNLOAD\_BUSY, [156](#)
- STATE\_DFU\_DOWNLOAD\_IDLE, [156](#)
- STATE\_DFU\_DOWNLOAD\_SYNC, [156](#)
- STATE\_DFU\_ERROR, [156](#)
- STATE\_DFU\_IDLE, [156](#)
- STATE\_DFU\_MANIFEST, [156](#)
- STATE\_DFU\_MANIFEST\_SYNC, [156](#)
- STATE\_DFU\_MANIFEST\_WAIT\_RESET, [156](#)
- STATE\_DFU\_UPLOAD\_IDLE, [157](#)
- Dfu.h, [158](#)
- DFU\_ABORT
  - Dfu.cpp, [151](#)
- DFU\_CLRSTATUS
  - Dfu.cpp, [152](#)
- DFU\_DETACH
  - Dfu.cpp, [152](#)
- DFU\_DNLOAD
  - Dfu.cpp, [152](#)
- DFU\_GETSTATE
  - Dfu.cpp, [152](#)
- DFU\_GETSTATUS
  - Dfu.cpp, [152](#)
- DFU\_IFF\_ALT
  - Dfu.cpp, [152](#)
- DFU\_IFF\_CONFIG
  - Dfu.cpp, [152](#)
- DFU\_IFF\_DEVNUM
  - Dfu.cpp, [152](#)
- DFU\_IFF\_DFU
  - Dfu.cpp, [153](#)
- DFU\_IFF\_IFACE
  - Dfu.cpp, [153](#)
- DFU\_IFF\_PATH
  - Dfu.cpp, [153](#)
- DFU\_IFF\_PRODUCT
  - Dfu.cpp, [153](#)
- DFU\_IFF\_VENDOR
  - Dfu.cpp, [153](#)
- DFU\_STATUS\_ERROR\_ADDRESS
  - Dfu.cpp, [153](#)
- DFU\_STATUS\_ERROR\_CHECK\_ERASED
  - Dfu.cpp, [153](#)
- DFU\_STATUS\_ERROR\_ERASE
  - Dfu.cpp, [153](#)
- DFU\_STATUS\_ERROR\_FILE
  - Dfu.cpp, [154](#)
- DFU\_STATUS\_ERROR\_FIRMWARE
  - Dfu.cpp, [154](#)
- DFU\_STATUS\_ERROR\_NOTDONE
  - Dfu.cpp, [154](#)
- DFU\_STATUS\_ERROR\_POR
  - Dfu.cpp, [154](#)
- DFU\_STATUS\_ERROR\_PROG
  - Dfu.cpp, [154](#)
- DFU\_STATUS\_ERROR\_STALLEDPKT
  - Dfu.cpp, [154](#)
- DFU\_STATUS\_ERROR\_TARGET
  - Dfu.cpp, [154](#)
- DFU\_STATUS\_ERROR\_UNKNOWN
  - Dfu.cpp, [154](#)
- DFU\_STATUS\_ERROR\_USBR
  - Dfu.cpp, [155](#)
- DFU\_STATUS\_ERROR\_VENDOR
  - Dfu.cpp, [155](#)
- DFU\_STATUS\_ERROR\_VERIFY
  - Dfu.cpp, [155](#)
- DFU\_STATUS\_ERROR\_WRITE
  - Dfu.cpp, [155](#)
- DFU\_STATUS\_OK
  - Dfu.cpp, [155](#)
- DFU\_UPLOAD
  - Dfu.cpp, [155](#)
- dfuse\_command
  - Dfu.cpp, [157](#)
- DfuStatus, [117](#)
  - iString, [117](#)
  - pollTimeout, [117](#)
  - state, [117](#)
  - status, [117](#)
- DigitalInOut
  - BMeasureApi, [20](#)
- DigitalMode
  - BMeasureApi, [20](#)
- digitalMode
  - BMeasureApi::Configuration, [101](#)
- DigitalModeInput
  - BMeasureApi, [20](#)
- DigitalModeOutput
  - BMeasureApi, [20](#)
- DigitalModeSyncMaster
  - BMeasureApi, [20](#)
- DigitalModeSyncSlave
  - BMeasureApi, [20](#)
- digitalPins
  - BMeasureApi::Configuration, [101](#)
- disconnect
  - BMeasureApi::BMeasureUnit, [57](#)
  - BMeasureApi::CommsNet, [89](#)

- BMeasureApi::CommsSerial, 93
- BMeasureApi::CommsUsb, 95
- Dfu, 115
- disconnected
  - BMeasureApi::BMeasureUnit, 57
  - BMeasureApi::BMeasureUnit1, 63
  - BMeasureApi::BMeasureUnits, 69
- duty
  - BMeasureApi::AwgConfig, 31
- emailAddress
  - BMeasureApi::Configuration, 101
- emailMode
  - BMeasureApi::Configuration, 101
- enabled
  - BMeasureApi::ChannelConfig, 86
- ERASE\_PAGE
  - Dfu.cpp, 157
- error
  - BMeasureApi::NodeStatus, 132
- ErrorDataOverrun
  - BMeasureApi, 20
- ErrorNum
  - BMeasureApi, 20
- errorStr
  - BMeasureApi::NodeStatus, 132
- ErrorSystem
  - BMeasureApi, 20
- ErrorToFast
  - BMeasureApi, 20
- EventMode
  - BMeasureApi, 20
- EventModeAlarm
  - BMeasureApi, 20
- EventModeOff
  - BMeasureApi, 20
- EventModeSecond
  - BMeasureApi, 20
- extra
  - BMdnsService, 36
- factoryReset
  - BMeasureApi::BMeasure, 40
- factoryResetServe
  - BMeasureApi::BMeasure, 40
- fileClose
  - BMeasureApi::BMeasure, 40
- fileCloseServe
  - BMeasureApi::BMeasure, 40
- fileDelete
  - BMeasureApi::BMeasure, 41
- fileDeleteServe
  - BMeasureApi::BMeasure, 41
- fileLength
  - BMeasureApi::FileInfo, 119
- fileList
  - BMeasureApi::BMeasure, 41
- fileListServe
  - BMeasureApi::BMeasure, 41
- fileOpen
  - BMeasureApi::BMeasure, 41
- fileOpenServe
  - BMeasureApi::BMeasure, 41
- fileRead
  - BMeasureApi::BMeasure, 42
- fileReadServe
  - BMeasureApi::BMeasure, 42
- filesysDelete
  - BMeasureApi::BMeasure, 42
- filesysDeleteServe
  - BMeasureApi::BMeasure, 42
- FilesysDeleteType
  - BMeasureApi, 20
- FilesysDeleteTypeData
  - BMeasureApi, 21
- FilesysDeleteTypeFormat
  - BMeasureApi, 21
- FilesysDeleteTypeNone
  - BMeasureApi, 21
- filesysInfo
  - BMeasureApi::BMeasure, 42
- filesysInfoServe
  - BMeasureApi::BMeasure, 42
- FileType
  - BMeasureApi, 21
- fileType
  - BMeasureApi::FileInfo, 119
- FileTypeDir
  - BMeasureApi, 21
- FileTypeFile
  - BMeasureApi, 21
- FileTypeNone
  - BMeasureApi, 21
- fileWrite
  - BMeasureApi::BMeasure, 43
- fileWriteServe
  - BMeasureApi::BMeasure, 43
- findDevices
  - BMeasureApi::BMeasureUnit, 58
- findDevicesNetwork
  - BMeasureApi::BMeasureUnit, 58
- findDevicesUsb
  - BMeasureApi::BMeasureUnit, 58
- findServices
  - BMdns, 35
- fpgaVersion
  - BMeasureApi::NodeInfo, 130
- free
  - BMeasureApi::FilesysInfo, 121
- frequency
  - BMeasureApi::AwgConfig, 32
- functionUnLock
  - BMeasureApi::BMeasure, 43
- functionUnLockServe
  - BMeasureApi::BMeasure, 43
- getAwgConfig
  - BMeasureApi::BMeasure, 43

- BMeasureApi::BMeasureUnits, 69
- getAwgConfigServe
  - BMeasureApi::BMeasure, 43
- getBoardConfig
  - BMeasureApi::BMeasure, 44
- getBoardConfigServe
  - BMeasureApi::BMeasure, 44
- getChannelConfig
  - BMeasureApi::BMeasure, 44
  - BMeasureApi::BMeasureUnits, 70
- getChannelConfigServe
  - BMeasureApi::BMeasure, 44
- getConfig
  - BMeasureApi::BMeasure, 44
  - BMeasureApi::BMeasureUnits, 70
- getConfigServe
  - BMeasureApi::BMeasure, 44
- getDigital
  - BMeasureApi::BMeasure, 45
- getDigitalServe
  - BMeasureApi::BMeasure, 45
- getFileName
  - BMeasureApi::DataFile, 110
- getFreeBlock
  - BMeasureApi::BMeasureUnits, 70
- getInfoBlock
  - BMeasureApi::BMeasure, 45
  - BMeasureApi::BMeasureUnits, 70
- getInfoBlockServe
  - BMeasureApi::BMeasure, 45
- getInformation
  - BMeasureApi::BMeasure, 45
  - BMeasureApi::BMeasureUnits, 70
- getInformationServe
  - BMeasureApi::BMeasure, 45
- getMeasurement
  - BMeasureApi::BMeasure, 46
  - BMeasureApi::BMeasureUnits, 70
- getMeasurementConfig
  - BMeasureApi::BMeasure, 46
  - BMeasureApi::BMeasureUnits, 71
- getMeasurementConfigServe
  - BMeasureApi::BMeasure, 46
- getMeasurementServe
  - BMeasureApi::BMeasure, 46
- getMembers
  - BMeasureApi::AlarmConfig, 29
  - BMeasureApi::AwgConfig, 31
  - BMeasureApi::BoardConfig, 81
  - BMeasureApi::CalibrateInfo, 83
  - BMeasureApi::ChannelConfig, 85
  - BMeasureApi::ConfigItem, 98
  - BMeasureApi::Configuration, 100
  - BMeasureApi::DataBlock, 107
  - BMeasureApi::FileData, 118
  - BMeasureApi::FileInfo, 119
  - BMeasureApi::FilesysInfo, 120
  - BMeasureApi::InfoBlock, 122
  - BMeasureApi::Information, 124
  - BMeasureApi::MeasurementConfig, 127
  - BMeasureApi::NodeInfo, 130
  - BMeasureApi::NodeStatus, 131
  - BMeasureApi::Version, 133
- getNodeInfo
  - BMeasureApi::BMeasure, 46
- getNodeInfoServe
  - BMeasureApi::BMeasure, 46
- getStatus
  - BMeasureApi::BMeasure, 47
  - BMeasureApi::BMeasureUnits, 71
  - Dfu, 115
- getStatusServe
  - BMeasureApi::BMeasure, 47
- getSwitch
  - BMeasureApi::BMeasure, 47
- getSwitchServe
  - BMeasureApi::BMeasure, 47
- hardwareVersion
  - BMeasureApi::BoardConfig, 82
  - BMeasureApi::NodeInfo, 130
- hostname
  - BMdnsService, 36
- id
  - BMeasureApi::ChannelConfig, 87
- info
  - BMeasureApi::BMeasureUnit, 58
- init
  - BMdns, 35
  - BMeasureApi::BMeasureUnitsDataBlock, 79
  - BMeasureApi::CommsNet, 90
  - BMeasureApi::DataFile, 110
  - Dfu, 115
- iString
  - DfuStatus, 117
- length
  - BFirmwareInfo, 33
  - BMeasureApi::FileData, 118
- level
  - BMeasureApi::AlarmConfig, 30
- location
  - BMeasureApi::Configuration, 101
  - BMeasureApi::InfoBlock, 122
- logData
  - BMeasureApi::Configuration, 101
- logDataDevice
  - BMeasureApi::Configuration, 102
- LogDataMode
  - BMeasureApi, 21
- logDataMode
  - BMeasureApi::Configuration, 102
- LogDataModeDeleteOld
  - BMeasureApi, 21
- LogDataModeNormal
  - BMeasureApi, 21



- login
  - BMeasureApi::BMeasure, 47
- loginServe
  - BMeasureApi::BMeasure, 47
- macAddress
  - BMeasureApi::BoardConfig, 82
- magic
  - BFirmwareInfo, 33
  - BMeasureApi::BoardConfig, 82
- MASS\_ERASE
  - Dfu.cpp, 157
- MDNS\_CLASS\_IN
  - BMdns.cpp, 136
- MDNS\_ENTRYTYPE\_ADDITIONAL
  - BMdns.cpp, 136
- MDNS\_ENTRYTYPE\_ANSWER
  - BMdns.cpp, 136
- MDNS\_ENTRYTYPE\_AUTHORITY
  - BMdns.cpp, 136
- mdns\_read\_string
  - BMdns.cpp, 137
- mdns\_read\_strings
  - BMdns.cpp, 137
- MDNS\_RECORDTYPE\_A
  - BMdns.cpp, 136
- MDNS\_RECORDTYPE\_AAAA
  - BMdns.cpp, 136
- MDNS\_RECORDTYPE\_IGNORE
  - BMdns.cpp, 136
- MDNS\_RECORDTYPE\_PTR
  - BMdns.cpp, 136
- MDNS\_RECORDTYPE\_SRV
  - BMdns.cpp, 136
- MDNS\_RECORDTYPE\_TXT
  - BMdns.cpp, 136
- mdns\_write\_string
  - BMdns.cpp, 137
- MdnsClass
  - BMdns.cpp, 136
- MdnsEntryType
  - BMdns.cpp, 136
- MdnsRecordType
  - BMdns.cpp, 136
- measure
  - BMeasureApi::BMeasure, 48
- measureConfig
  - BMeasureApi::InfoBlock, 122
- MeasureMode
  - BMeasureApi, 21
- measureMode
  - BMeasureApi::MeasurementConfig, 128
- MeasureModeContinuous
  - BMeasureApi, 21
- MeasureModeOff
  - BMeasureApi, 21
- MeasureModeOneShot
  - BMeasureApi, 21
- MeasureModeRepeat
  - BMeasureApi, 21
- measurePeriod
  - BMeasureApi::MeasurementConfig, 128
- measureServe
  - BMeasureApi::BMeasure, 48
- MessageSource
  - BMeasureApi, 22
- MessageSourceDebug
  - BMeasureApi, 22
- MessageSourceGeneral
  - BMeasureApi, 22
- MessageSourceTest
  - BMeasureApi, 22
- MessageSourceWifi
  - BMeasureApi, 22
- MessageSourceWifiTest
  - BMeasureApi, 22
- Mode
  - BMeasureApi, 22
- mode
  - BMeasureApi::AlarmConfig, 30
  - BMeasureApi::Configuration, 102
  - BMeasureApi::NodeStatus, 132
- ModeDemo1
  - BMeasureApi, 22
- ModelIdle
  - BMeasureApi, 22
- ModelInternal
  - BMeasureApi, 22
- ModeRun
  - BMeasureApi, 22
- ModeRunProgram
  - BMeasureApi, 22
- ModeSleep
  - BMeasureApi, 22
- mqttMode
  - BMeasureApi::Configuration, 102
- mqttPort
  - BMeasureApi::Configuration, 102
- mqttServer
  - BMeasureApi::Configuration, 102
- name
  - BMdnsService, 36
  - BMeasureApi::ChannelConfig, 87
  - BMeasureApi::ConfigItem, 98
  - BMeasureApi::Configuration, 103
  - BMeasureApi::FileInfo, 119
  - BMeasureApi::FilesysInfo, 121
  - BMeasureApi::InfoBlock, 122
- networkAddress
  - BMeasureApi::Configuration, 103
  - BMeasureApi::Information, 125
- networkGateway
  - BMeasureApi::Configuration, 103
  - BMeasureApi::Information, 125
- networkMask
  - BMeasureApi::Configuration, 103
  - BMeasureApi::Information, 125

- NetworkMode
  - BMeasureApi, 22
- networkMode
  - BMeasureApi::Configuration, 103
  - BMeasureApi::Information, 125
- NetworkModeDhcp
  - BMeasureApi, 22
- NetworkModeManual
  - BMeasureApi, 22
- NetworkModeOff
  - BMeasureApi, 22
- networkNameServer0
  - BMeasureApi::Configuration, 103
  - BMeasureApi::Information, 125
- networkTimeServer
  - BMeasureApi::Configuration, 104
  - BMeasureApi::Information, 125
- nodeInfo
  - BMeasureApi::InfoBlock, 123
  - BMeasureApi::Information, 126
- NodeType
  - BMeasureApi, 22
- NodeTypeBMeasure1
  - BMeasureApi, 23
- NodeTypeNone
  - BMeasureApi, 23
- number
  - BMeasureApi::ChannelConfig, 87
- numChannels
  - BMeasureApi::BMeasureUnit, 58
  - BMeasureApi::BMeasureUnits, 71
  - BMeasureApi::DataBlock, 107
  - BMeasureApi::InfoBlock, 123
  - BMeasureApi::Information, 126
- numConfigItems
  - BMeasureApi::Information, 126
- numSamples
  - BMeasureApi::AwgConfig, 32
  - BMeasureApi::DataBlock, 107
- numSamples0
  - BMeasureApi::MeasurementConfig, 128
- numSamples1
  - BMeasureApi::MeasurementConfig, 128
- numSamplesBlock
  - BMeasureApi::MeasurementConfig, 128
- oblockCount
  - BMeasureApi::BMeasureUnit, 60
- obuffer
  - BMeasureApi::CommsUsb, 96
- ochannels
  - BMeasureApi::BMeasureUnit, 60
- oconfigMeasurement
  - BMeasureApi::BMeasureUnit, 60
- oconnected
  - BMeasureApi::BMeasureUnit1, 63
  - Dfu, 116
- ocontext
  - BMeasureApi::CommsUsb, 97
- Dfu, 116
- odataBlock
  - BMeasureApi::BMeasureUnit, 60
  - BMeasureApi::BMeasureUnitsDataBlock, 79
- odataBlocksFree
  - BMeasureApi::BMeasureUnits, 75
- odataBlocksIn
  - BMeasureApi::BMeasureUnits, 76
- odataBlocksOut
  - BMeasureApi::BMeasureUnits, 76
- odataBlocksOutCount
  - BMeasureApi::BMeasureUnits, 76
- odataBlocksProcess
  - BMeasureApi::BMeasureUnits, 76
- odataBlocksProcessNum
  - BMeasureApi::BMeasureUnits, 76
- odataStreamNum
  - BMeasureApi::BMeasureUnits, 76
- odev
  - BMeasureApi::CommsUsb, 97
  - Dfu, 116
- odevice
  - BMeasureApi::BMeasureUnit, 60
  - BMeasureApi::CommsSerial, 94
  - BMeasureApi::CommsUsb, 97
- odisconnecting
  - BMeasureApi::BMeasureUnit, 61
- oenabled
  - BMeasureApi::BMeasureUnit1, 64
- offset
  - BMeasureApi::AwgConfig, 32
  - BMeasureApi::ChannelConfig, 87
- ofile
  - BMeasureApi::DataFile, 112
- ofileName
  - BMeasureApi::DataFile, 112
- ofill
  - BMeasureApi::BMeasureUnits, 76
  - BMeasureApi::BMeasureUnitsDataBlock, 80
- oformat
  - BMeasureApi::DataFile, 112
- oinfo
  - BMeasureApi::BMeasureUnit, 61
- oinUse
  - BMeasureApi::BMeasureUnitsDataBlock, 80
- oinWait
  - BMeasureApi::CommsNet, 91
- olocalTrigger
  - BMeasureApi::BMeasureUnits, 76
- olockInput
  - BMeasureApi::BMeasureUnits, 77
- olockOutput
  - BMeasureApi::BMeasureUnits, 77
- olockUnits
  - BMeasureApi::BMeasureUnits, 77
- omeasureUnits
  - BMeasureApi::BMeasureUnit1, 64
- omode

- BMeasureApi::DataFile, 113
- onodeInfo
  - BMeasureApi::BMeasureUnit, 61
- onum
  - BMeasureApi::CommsUsb, 97
- onumBlocks
  - BMeasureApi::BMeasureUnits, 77
- onumChannels
  - BMeasureApi::BMeasureUnits, 77
- onumConnected
  - BMeasureApi::BMeasureUnits, 77
- oorder
  - BMeasureApi::BMeasureUnit1, 64
- opacket
  - BMeasureApi::DataFile, 113
- opacketLen
  - BMeasureApi::DataFile, 113
- open
  - BMeasureApi::DataFile, 110
- oprocEnable
  - BMeasureApi::BMeasureUnits, 77
- oprocRunning
  - BMeasureApi::BMeasureUnits, 78
- osampleCount
  - BMeasureApi::BMeasureUnit, 61
- osequenceNext
  - BMeasureApi::BMeasureUnit, 61
- oserialNumber
  - BMeasureApi::BMeasureUnit1, 64
- oserialPort
  - BMeasureApi::CommsSerial, 94
- osocket
  - Bmdns, 35
  - BMeasureApi::CommsNet, 91
- osource
  - BMeasureApi::BMeasureUnit1, 64
- ostartSample
  - BMeasureApi::BMeasureUnits, 78
- oterminated
  - BMeasureApi::CommsUsb, 97
- oterminating
  - BMeasureApi::CommsNet, 91
  - BMeasureApi::CommsUsb, 97
- otransactionId
  - Bmdns, 35
- otriggered
  - BMeasureApi::BMeasureUnits, 78
- ounitMaster
  - BMeasureApi::BMeasureUnits, 78
- ounits
  - BMeasureApi::BMeasureUnits, 78
- output
  - BMeasureApi::AlarmConfig, 30
  - BMeasureApi::AwgConfig, 32
- outputBlock
  - BMeasureApi::BMeasureUnits, 71
- outputChannel
  - BMeasureApi::AlarmConfig, 30
- overbose
  - Dfu, 116
- overview.dox, 158
- pageAddress
  - Dfu.cpp, 157
- pageNumber
  - Dfu.cpp, 157
- pgaGain
  - BMeasureApi::ChannelConfig, 87
- pollTimeout
  - DfuStatus, 117
- process
  - BMeasureApi::ChannelConfig, 87
- processdataBlock
  - BMeasureApi::BMeasureUnit, 58
- processRequest
  - BMeasureApi::BMeasure, 48
- program
  - BMeasureApi::Configuration, 104
- read
  - BMeasureApi::CommsNet, 90
  - BMeasureApi::CommsSerial, 93
  - BMeasureApi::CommsUsb, 95
- READ\_UNPROTECT
  - Dfu.cpp, 157
- readAvailable
  - BMeasureApi::CommsNet, 90
  - BMeasureApi::CommsSerial, 93
  - BMeasureApi::CommsUsb, 96
- readChunk
  - BMeasureApi::CommsUsb, 96
- readData
  - BMeasureApi::DataFile, 111
- readInfo
  - BMeasureApi::DataFile, 111
- reset
  - Dfu, 115
- round512
  - BMeasureApi, 26
- roundDown512
  - BMeasureApi, 26
- rs485BaudRate
  - BMeasureApi::Configuration, 104
- rs485Bits
  - BMeasureApi::Configuration, 104
- Rs485Mode
  - BMeasureApi, 23
- rs485Mode
  - BMeasureApi::Configuration, 104
- Rs485ModeBoap
  - BMeasureApi, 23
- Rs485ModeOff
  - BMeasureApi, 23
- rs485StopBits
  - BMeasureApi::Configuration, 104
- run
  - BMeasureApi::BMeasureUnit, 59

- BMeasureApi::BMeasureUnits, 71
- runBoardTest
  - BMeasureApi::BMeasure, 48
- runBoardTestServe
  - BMeasureApi::BMeasure, 48
- sampleFrequencyMode
  - BMeasureApi::Configuration, 105
- sampleRate
  - BMeasureApi::MeasurementConfig, 128
- SampleType
  - BMeasureApi, 23
- sampleType
  - BMeasureApi::ChannelConfig, 87
- SampleTypeBool
  - BMeasureApi, 23
- SampleTypeFloat32
  - BMeasureApi, 23
- SampleTypeFloat64
  - BMeasureApi, 23
- SampleTypeInt16
  - BMeasureApi, 23
- SampleTypeInt32
  - BMeasureApi, 23
- SampleTypeInt8
  - BMeasureApi, 23
- SampleTypeNone
  - BMeasureApi, 23
- sampleTypeString
  - BMeasureApi, 26
- scale
  - BMeasureApi::ChannelConfig, 88
- SecureMode
  - BMeasureApi, 23
- SecureModeFull
  - BMeasureApi, 23
- SecureModeOpen
  - BMeasureApi, 23
- SecureMoteRemote
  - BMeasureApi, 23
- sendData
  - BMeasureApi::BMeasure, 48
- sendDataEnable
  - BMeasureApi::BMeasure, 49
  - BMeasureApi::BMeasureUnits, 71
- sendDataEnableServe
  - BMeasureApi::BMeasure, 49
- sendDataProcess
  - BMeasureApi::BMeasureUnits, 72
- sendDataProcessTrigger
  - BMeasureApi::BMeasureUnits, 72
- sendDataQueue
  - BMeasureApi::BMeasureUnits, 72
- sendDataServe
  - BMeasureApi::BMeasure, 49
  - BMeasureApi::BMeasureUnit, 59
- sendDataServe1
  - BMeasureApi::BMeasureUnit, 59
  - BMeasureApi::BMeasureUnit1, 63
- BMeasureApi::BMeasureUnits, 72
- sendInfo
  - BMeasureApi::BMeasure, 49
- sendInfoServe
  - BMeasureApi::BMeasure, 49
- sendMessage
  - BMeasureApi::BMeasure, 49
  - BMeasureApi::BMeasureUnits, 72
- sendMessageServe
  - BMeasureApi::BMeasure, 50
  - BMeasureApi::BMeasureUnit1, 63
  - BMeasureApi::BMeasureUnits, 72
- sendStatus
  - BMeasureApi::BMeasure, 50
- sendStatusServe
  - BMeasureApi::BMeasure, 50
- sendTime
  - BMeasureApi::BMeasure, 50
  - BMeasureApi::BMeasureUnits, 72
- sendTimeServe
  - BMeasureApi::BMeasure, 50
- sequence
  - BMeasureApi::DataBlock, 108
- serialNumber
  - BMeasureApi::BMeasureUnit, 59
  - BMeasureApi::BMeasureUnit1, 63
  - BMeasureApi::BMeasureUnitDevice, 65
  - BMeasureApi::BoardConfig, 82
  - BMeasureApi::NodeInfo, 130
- SET\_ADDRESS
  - Dfu.cpp, 157
- setAnalogueOut
  - BMeasureApi::BMeasure, 50
- setAnalogueOutServe
  - BMeasureApi::BMeasure, 51
- setAwgConfig
  - BMeasureApi::BMeasure, 51
  - BMeasureApi::BMeasureUnits, 73
- setAwgConfigServe
  - BMeasureApi::BMeasure, 51
- setAwgWaveform
  - BMeasureApi::BMeasure, 51
- setAwgWaveformServe
  - BMeasureApi::BMeasure, 51
- setBoardConfig
  - BMeasureApi::BMeasure, 51
- setBoardConfigServe
  - BMeasureApi::BMeasure, 52
- setChannelConfig
  - BMeasureApi::BMeasure, 52
  - BMeasureApi::BMeasureUnit, 59
  - BMeasureApi::BMeasureUnits, 73
- setChannelConfigFull
  - BMeasureApi::BMeasure, 52
- setChannelConfigFullServe
  - BMeasureApi::BMeasure, 52
- setChannelConfigServe
  - BMeasureApi::BMeasure, 52

- setConfig
  - BMeasureApi::BMeasure, [52](#)
  - BMeasureApi::BMeasureUnits, [73](#)
- setConfigServe
  - BMeasureApi::BMeasure, [53](#)
- setDigital
  - BMeasureApi::BMeasure, [53](#)
- setDigitalServe
  - BMeasureApi::BMeasure, [53](#)
- setMeasurement
  - BMeasureApi::BMeasure, [53](#)
  - BMeasureApi::BMeasureUnit, [59](#)
  - BMeasureApi::BMeasureUnits, [73](#)
- setMeasurementConfig
  - BMeasureApi::BMeasure, [53](#)
  - BMeasureApi::BMeasureUnits, [73](#)
- setMeasurementConfigServe
  - BMeasureApi::BMeasure, [53](#)
- setMeasurementServe
  - BMeasureApi::BMeasure, [54](#)
- setMode
  - BMeasureApi::BMeasure, [54](#)
  - BMeasureApi::BMeasureUnits, [73](#)
- setModeServe
  - BMeasureApi::BMeasure, [54](#)
- setRelay
  - BMeasureApi::BMeasure, [54](#)
- setRelayServe
  - BMeasureApi::BMeasure, [54](#)
- setSecureKey
  - BMeasureApi::BMeasure, [54](#)
- setSecureKeyServe
  - BMeasureApi::BMeasure, [55](#)
- setSecureMode
  - BMeasureApi::BMeasure, [55](#)
- setSecureModeServe
  - BMeasureApi::BMeasure, [55](#)
- setSerialNumber
  - BMeasureApi::BMeasureUnit1, [63](#)
- siUnits
  - BMeasureApi::ChannelConfig, [88](#)
- size
  - BMeasureApi::FilesysInfo, [121](#)
- softwareVersion
  - BMeasureApi::NodeInfo, [131](#)
- source
  - BMeasureApi::Configuration, [105](#)
  - BMeasureApi::DataBlock, [108](#)
  - BMeasureApi::InfoBlock, [123](#)
- spare
  - BMeasureApi::AwgConfig, [32](#)
  - BMeasureApi::ConfigItem, [98](#)
  - BMeasureApi::DataBlock, [108](#)
  - BMeasureApi::FileInfo, [120](#)
  - BMeasureApi::NodeStatus, [132](#)
- spare0
  - BMeasureApi::BoardConfig, [82](#)
  - BMeasureApi::ChannelConfig, [88](#)
  - BMeasureApi::Configuration, [105](#)
  - BMeasureApi::InfoBlock, [123](#)
  - BMeasureApi::Information, [126](#)
- spare1
  - BMeasureApi::Configuration, [105](#)
  - BMeasureApi::Information, [126](#)
- spare2
  - BMeasureApi::Configuration, [105](#)
- spare3
  - BMeasureApi::Configuration, [105](#)
- spare4
  - BMeasureApi::Configuration, [105](#)
- spare5
  - BMeasureApi::Configuration, [106](#)
- stage
  - BMeasureApi::CalibrateInfo, [84](#)
- state
  - DfuStatus, [117](#)
- STATE\_APP\_DETACH
  - Dfu.cpp, [155](#)
- STATE\_APP\_IDLE
  - Dfu.cpp, [155](#)
- STATE\_DFU\_DOWNLOAD\_BUSY
  - Dfu.cpp, [156](#)
- STATE\_DFU\_DOWNLOAD\_IDLE
  - Dfu.cpp, [156](#)
- STATE\_DFU\_DOWNLOAD\_SYNC
  - Dfu.cpp, [156](#)
- STATE\_DFU\_ERROR
  - Dfu.cpp, [156](#)
- STATE\_DFU\_IDLE
  - Dfu.cpp, [156](#)
- STATE\_DFU\_MANIFEST
  - Dfu.cpp, [156](#)
- STATE\_DFU\_MANIFEST\_SYNC
  - Dfu.cpp, [156](#)
- STATE\_DFU\_MANIFEST\_WAIT\_RESET
  - Dfu.cpp, [156](#)
- STATE\_DFU\_UPLOAD\_IDLE
  - Dfu.cpp, [157](#)
- Status
  - BMeasureApi, [23](#)
- status
  - BMeasureApi::DataBlock, [108](#)
  - BMeasureApi::NodeStatus, [132](#)
  - DfuStatus, [117](#)
- StatusAlarm
  - BMeasureApi, [24](#)
- StatusDataOverrun
  - BMeasureApi, [24](#)
- StatusEnd0
  - BMeasureApi, [24](#)
- StatusEnd1
  - BMeasureApi, [24](#)
- StatusError
  - BMeasureApi, [24](#)
- StatusFpgaOverrun
  - BMeasureApi, [24](#)

- StatusNone
  - BMeasureApi, [24](#)
- StatusRun
  - BMeasureApi, [24](#)
- StatusTriggerWait
  - BMeasureApi, [24](#)
- StatusWarning
  - BMeasureApi, [24](#)
- SyncMode
  - BMeasureApi, [24](#)
- SyncModeMaster
  - BMeasureApi, [24](#)
- SyncModeOff
  - BMeasureApi, [24](#)
- SyncModeSlave
  - BMeasureApi, [24](#)
- TdsDataType
  - BMeasureApi, [24](#)
- TdsTypeBoolean
  - BMeasureApi, [25](#)
- TdsTypeComplexDoubleFloat
  - BMeasureApi, [25](#)
- TdsTypeComplexSingleFloat
  - BMeasureApi, [25](#)
- TdsTypeDAQmxRawData
  - BMeasureApi, [25](#)
- TdsTypeDoubleFloat
  - BMeasureApi, [24](#)
- TdsTypeDoubleFloatWithUnit
  - BMeasureApi, [25](#)
- TdsTypeExtendedFloat
  - BMeasureApi, [25](#)
- TdsTypeExtendedFloatWithUnit
  - BMeasureApi, [25](#)
- TdsTypeFixedPoint
  - BMeasureApi, [25](#)
- TdsTypeI16
  - BMeasureApi, [24](#)
- TdsTypeI32
  - BMeasureApi, [24](#)
- TdsTypeI64
  - BMeasureApi, [24](#)
- TdsTypeI8
  - BMeasureApi, [24](#)
- TdsTypeSingleFloat
  - BMeasureApi, [24](#)
- TdsTypeSingleFloatWithUnit
  - BMeasureApi, [25](#)
- TdsTypeString
  - BMeasureApi, [25](#)
- TdsTypeTimeStamp
  - BMeasureApi, [25](#)
- TdsTypeU16
  - BMeasureApi, [24](#)
- TdsTypeU32
  - BMeasureApi, [24](#)
- TdsTypeU64
  - BMeasureApi, [24](#)
- TdsTypeU8
  - BMeasureApi, [24](#)
- TdsTypeVoid
  - BMeasureApi, [24](#)
- testMode
  - BMeasureApi::BoardConfig, [83](#)
- time
  - BMeasureApi::DataBlock, [108](#)
  - BMeasureApi::FileInfo, [120](#)
  - BMeasureApi::InfoBlock, [123](#)
  - BMeasureApi::Information, [126](#)
  - BMeasureApi::NodeStatus, [132](#)
- TocBigEndian
  - BMeasureApi, [27](#)
- TocDaqRawData
  - BMeasureApi, [27](#)
- TocInterleavedData
  - BMeasureApi, [27](#)
- TocMetaData
  - BMeasureApi, [27](#)
- TocNewObjList
  - BMeasureApi, [27](#)
- TocRawData
  - BMeasureApi, [27](#)
- toFloat
  - BMeasureApi, [27](#)
- triggerChannel
  - BMeasureApi::MeasurementConfig, [129](#)
- TriggerConfig
  - BMeasureApi, [25](#)
- triggerConfig
  - BMeasureApi::MeasurementConfig, [129](#)
- TriggerConfigNone
  - BMeasureApi, [25](#)
- triggerDelay
  - BMeasureApi::MeasurementConfig, [129](#)
- triggerLevel
  - BMeasureApi::MeasurementConfig, [129](#)
- TriggerMode
  - BMeasureApi, [25](#)
- triggerMode
  - BMeasureApi::MeasurementConfig, [129](#)
- TriggerModeNegative
  - BMeasureApi, [25](#)
- TriggerModeOff
  - BMeasureApi, [25](#)
- TriggerModePositive
  - BMeasureApi, [25](#)
- type
  - BFirmwareInfo, [33](#)
  - BMeasureApi::ChannelConfig, [88](#)
  - BMeasureApi::ConfigItem, [98](#)
  - BMeasureApi::DataBlock, [108](#)
  - BMeasureApi::Version, [133](#)
- unit
  - BMeasureApi::BMeasureUnits, [74](#)
- unitAdd
  - BMeasureApi::BMeasureUnits, [74](#)

- unitDelete
  - BMeasureApi::BMeasureUnits, 74
- unitMaster
  - BMeasureApi::BMeasureUnits, 74
- unitsConnect
  - BMeasureApi::BMeasureUnits, 74
- unitsConnected
  - BMeasureApi::BMeasureUnits, 74
- unitsConnectedNum
  - BMeasureApi::BMeasureUnits, 74
- unitsDisconnect
  - BMeasureApi::BMeasureUnits, 75
- unitSetEnabled
  - BMeasureApi::BMeasureUnits, 75
- unitSetOrder
  - BMeasureApi::BMeasureUnits, 75
- unitsFind
  - BMeasureApi::BMeasureUnits, 75
- unitsNum
  - BMeasureApi::BMeasureUnits, 75
- unitSort
  - BMeasureApi, 28
- upload
  - Dfu, 115
- upload\_cmd
  - Dfu, 115
- validateFile
  - Dfu, 116
- validateFormat
  - BMeasureApi::DataFile, 111
- value
  - BMeasureApi::CalibrateInfo, 84
  - BMeasureApi::ConfigItem, 99
- ver0
  - BFirmwareInfo, 34
  - BMeasureApi::Version, 133
- ver1
  - BFirmwareInfo, 34
  - BMeasureApi::Version, 133
- ver2
  - BFirmwareInfo, 34
  - BMeasureApi::Version, 133
- version
  - BMeasureApi::Configuration, 106
  - BMeasureApi::InfoBlock, 123
- wait
  - BMeasureApi::CommsNet, 90
  - BMeasureApi::CommsSerial, 93
  - BMeasureApi::CommsUsb, 96
- Waveform
  - BMeasureApi, 25
- waveform
  - BMeasureApi::AwgConfig, 32
- WaveformArbitrary
  - BMeasureApi, 26
- WaveformDc
  - BMeasureApi, 25
- WaveformNoise
  - BMeasureApi, 26
- WaveformNone
  - BMeasureApi, 25
- WaveformSine
  - BMeasureApi, 26
- WaveformSquare
  - BMeasureApi, 26
- WaveformTriangle
  - BMeasureApi, 26
- wifiAp0
  - BMeasureApi::Configuration, 106
- wifiAp0Password
  - BMeasureApi::Configuration, 106
- WifiMode
  - BMeasureApi, 26
- wifiMode
  - BMeasureApi::Configuration, 106
- WifiModeAp
  - BMeasureApi, 26
- WifiModeClient
  - BMeasureApi, 26
- WifiModeOff
  - BMeasureApi, 26
- wifiVersion
  - BMeasureApi::NodeInfo, 131
- write
  - BMeasureApi::CommsNet, 90
  - BMeasureApi::CommsSerial, 93
  - BMeasureApi::CommsUsb, 96
- writeAvailable
  - BMeasureApi::CommsNet, 90
- writeChunks
  - BMeasureApi::CommsNet, 91
- writeData
  - BMeasureApi::DataFile, 111
- writeEnd
  - BMeasureApi::DataFile, 111
- writeInfo
  - BMeasureApi::DataFile, 111
- writeInfoBMeas
  - BMeasureApi::DataFile, 112
- writeInfoCsv
  - BMeasureApi::DataFile, 112
- writeInfoTdms
  - BMeasureApi::DataFile, 112