

BMeasure-lib

1.3.0

Generated by Doxygen 1.12.0

1 BMeasure-lib	1
1.1 Introduction	1
1.2 Overview	2
1.3 API Usage	2
1.4 API Usage	3
2 Namespace Index	7
2.1 Namespace List	7
3 Hierarchical Index	9
3.1 Class Hierarchy	9
4 Class Index	11
4.1 Class List	11
5 File Index	13
5.1 File List	13
6 Namespace Documentation	15
6.1 BMeasureApi Namespace Reference	15
6.1.1 Typedef Documentation	16
6.1.1.1 ChannelConfigs	16
6.1.2 Enumeration Type Documentation	16
6.1.2.1 TdsDataType	16
6.1.3 Function Documentation	17
6.1.3.1 channelTypeString()	17
6.1.3.2 round512()	17
6.1.3.3 roundDown512()	17
6.1.3.4 sampleTypeString()	17
6.1.3.5 TocBigEndian()	17
6.1.3.6 TocDaqRawData()	17
6.1.3.7 TocInterleavedData()	17
6.1.3.8 TocMetaData()	17
6.1.3.9 TocNewObjList()	18
6.1.3.10 TocRawData()	18
6.1.3.11 toFloat()	18
6.1.3.12 unitSort()	18
7 Class Documentation	19
7.1 BMdns Class Reference	19
7.1.1 Constructor & Destructor Documentation	19
7.1.1.1 BMdns()	19
7.1.1.2 ~BMdns()	19
7.1.2 Member Function Documentation	20

7.1.2.1 findServices()	20
7.1.2.2 init()	20
7.1.3 Member Data Documentation	20
7.1.3.1 osocket	20
7.1.3.2 otransactionId	20
7.2 BMdnsService Class Reference	20
7.2.1 Member Data Documentation	20
7.2.1.1 address	20
7.2.1.2 extra	21
7.2.1.3 hostname	21
7.2.1.4 name	21
7.3 BMeasureApi::BMeasureUnit Class Reference	21
7.3.1 Constructor & Destructor Documentation	23
7.3.1.1 BMeasureUnit()	23
7.3.1.2 ~BMeasureUnit()	23
7.3.2 Member Function Documentation	23
7.3.2.1 connect()	23
7.3.2.2 device()	23
7.3.2.3 disconnect()	23
7.3.2.4 disconnected()	23
7.3.2.5 findDevices()	24
7.3.2.6 findDevicesNetwork()	24
7.3.2.7 findDevicesUsb()	24
7.3.2.8 getNodeInfo()	24
7.3.2.9 info()	24
7.3.2.10 numChannels()	24
7.3.2.11 processDataBlock()	24
7.3.2.12 run()	25
7.3.2.13 sendDataFloatServe()	25
7.3.2.14 sendDataProcServe()	25
7.3.2.15 sendDataServe()	25
7.3.2.16 sendStreamDataServe()	25
7.3.2.17 serialNumber()	25
7.3.2.18 setChannelConfig()	25
7.3.2.19 setMeasurementConfig()	25
7.3.3 Member Data Documentation	26
7.3.3.1 blockNumChannels	26
7.3.3.2 blockNumSamples	26
7.3.3.3 oblockCount	26
7.3.3.4 ochannels	26
7.3.3.5 oconfigMeasurement	26
7.3.3.6 odataBlockFloat	26

7.3.3.7 odevice	26
7.3.3.8 odisconnecting	26
7.3.3.9 oinfo	26
7.3.3.10 onodeInfo	27
7.3.3.11 oprocEnable	27
7.3.3.12 oprocRunning	27
7.3.3.13 osampleCount	27
7.3.3.14 osequenceNext	27
7.4 BMeasureApi::BMeasureUnit1 Class Reference	27
7.4.1 Constructor & Destructor Documentation	30
7.4.1.1 BMeasureUnit1()	30
7.4.2 Member Function Documentation	30
7.4.2.1 disconnected()	30
7.4.2.2 sendDataFloatServe()	30
7.4.2.3 sendDataProcServe()	30
7.4.2.4 sendMessageServe()	30
7.4.2.5 sendStatusServe()	30
7.4.2.6 serialNumber()	30
7.4.2.7 setSerialNumber()	31
7.4.3 Member Data Documentation	31
7.4.3.1 oconnected	31
7.4.3.2 oenabled	31
7.4.3.3 omeasureUnits	31
7.4.3.4 oorder	31
7.4.3.5 oserialNumber	31
7.4.3.6 osource	31
7.5 BMeasureApi::BMeasureUnitDevice Class Reference	31
7.5.1 Constructor & Destructor Documentation	32
7.5.1.1 BMeasureUnitDevice()	32
7.5.2 Member Data Documentation	32
7.5.2.1 device	32
7.5.2.2 serialNumber	32
7.6 BMeasureApi::BMeasureUnits Class Reference	32
7.6.1 Constructor & Destructor Documentation	35
7.6.1.1 BMeasureUnits()	35
7.6.1.2 ~BMeasureUnits()	35
7.6.2 Member Function Documentation	36
7.6.2.1 alarmsClear()	36
7.6.2.2 changePassword()	36
7.6.2.3 clear()	36
7.6.2.4 dataAvailable()	36
7.6.2.5 dataClear()	36

7.6.2.6 dataDone()	36
7.6.2.7 dataEvent()	36
7.6.2.8 dataProcDone()	36
7.6.2.9 dataProcEvent()	37
7.6.2.10 dataProcRead()	37
7.6.2.11 dataRead()	37
7.6.2.12 dataSetNumStreams()	37
7.6.2.13 dataStreamEnable()	37
7.6.2.14 dataWait()	37
7.6.2.15 debugPrint()	37
7.6.2.16 disconnected()	37
7.6.2.17 getAwgConfig()	38
7.6.2.18 getChannelConfig()	38
7.6.2.19 getConfig()	38
7.6.2.20 getFreeBlock()	38
7.6.2.21 getInfoBlock()	38
7.6.2.22 getInformation()	38
7.6.2.23 getMeasurementConfig()	38
7.6.2.24 getNodeInfo()	39
7.6.2.25 getStatus()	39
7.6.2.26 login()	39
7.6.2.27 logout()	39
7.6.2.28 numChannels()	39
7.6.2.29 outputBlock()	39
7.6.2.30 run()	39
7.6.2.31 sendDataEnable()	39
7.6.2.32 sendDataFloatQueue()	40
7.6.2.33 sendDataFloatServe()	40
7.6.2.34 sendDataProcess()	40
7.6.2.35 sendDataProcessTrigger()	40
7.6.2.36 sendDataProcQueue()	40
7.6.2.37 sendDataProcServe()	40
7.6.2.38 sendMessage()	40
7.6.2.39 sendMessageServe()	40
7.6.2.40 sendStatusServe()	40
7.6.2.41 sendTime()	41
7.6.2.42 setAwgConfig()	41
7.6.2.43 setChannelConfig()	41
7.6.2.44 setConfig()	41
7.6.2.45 setMeasurementConfig()	41
7.6.2.46 setMode()	41
7.6.2.47 setMulti()	41

7.6.2.48	unit()	42
7.6.2.49	unitAdd()	42
7.6.2.50	unitDelete()	42
7.6.2.51	unitMaster()	42
7.6.2.52	unitsConnect()	42
7.6.2.53	unitsConnected()	42
7.6.2.54	unitsConnectedNum()	42
7.6.2.55	unitsDisconnect()	42
7.6.2.56	unitSetEnabled()	42
7.6.2.57	unitSetOrder()	43
7.6.2.58	unitsFind()	43
7.6.2.59	unitsNum()	43
7.6.3	Member Data Documentation	43
7.6.3.1	odataBlocksFree	43
7.6.3.2	odataBlocksIn	43
7.6.3.3	odataBlocksOut	43
7.6.3.4	odataBlocksOutCount	43
7.6.3.5	odataBlocksProcess	43
7.6.3.6	odataBlocksProcessNum	43
7.6.3.7	odataProcBlocks	44
7.6.3.8	odataStreamNum	44
7.6.3.9	ofill	44
7.6.3.10	olocalTrigger	44
7.6.3.11	olockInput	44
7.6.3.12	olockOutput	44
7.6.3.13	olockProclnput	44
7.6.3.14	olockUnits	44
7.6.3.15	omulti	44
7.6.3.16	onumBlocks	44
7.6.3.17	onumChannels	45
7.6.3.18	onumConnected	45
7.6.3.19	oprocEnable	45
7.6.3.20	oprocRunning	45
7.6.3.21	ostartSample	45
7.6.3.22	otriggered	45
7.6.3.23	ounitMaster	45
7.6.3.24	ounits	45
7.7	BMeasureApi::BMeasureUnitsDataBlock Class Reference	46
7.7.1	Constructor & Destructor Documentation	46
7.7.1.1	BMeasureUnitsDataBlock()	46
7.7.1.2	~BMeasureUnitsDataBlock()	46
7.7.2	Member Function Documentation	46

7.7.2.1 init()	46
7.7.3 Member Data Documentation	46
7.7.3.1 odataBlock	46
7.7.3.2 ofill	46
7.7.3.3 oinUse	47
7.8 BMeasureApi::CommsNet Class Reference	47
7.8.1 Constructor & Destructor Documentation	48
7.8.1.1 CommsNet()	48
7.8.1.2 ~CommsNet()	48
7.8.2 Member Function Documentation	48
7.8.2.1 connect()	48
7.8.2.2 disconnect()	49
7.8.2.3 init()	49
7.8.2.4 read()	49
7.8.2.5 readAvailable()	49
7.8.2.6 wait()	49
7.8.2.7 write()	49
7.8.2.8 writeAvailable()	50
7.8.2.9 writeChunks()	50
7.8.3 Member Data Documentation	50
7.8.3.1 oinWait	50
7.8.3.2 osocket	50
7.8.3.3 oterminating	50
7.9 BMeasureApi::CommsSerial Class Reference	50
7.9.1 Constructor & Destructor Documentation	52
7.9.1.1 CommsSerial()	52
7.9.1.2 ~CommsSerial()	52
7.9.2 Member Function Documentation	52
7.9.2.1 connect()	52
7.9.2.2 disconnect()	52
7.9.2.3 read()	52
7.9.2.4 readAvailable()	52
7.9.2.5 wait()	53
7.9.2.6 write()	53
7.9.3 Member Data Documentation	53
7.9.3.1 odevice	53
7.9.3.2 oserialPort	53
7.10 BMeasureApi::CommsUsb Class Reference	53
7.10.1 Constructor & Destructor Documentation	55
7.10.1.1 CommsUsb()	55
7.10.1.2 ~CommsUsb()	55
7.10.2 Member Function Documentation	55

7.10.2.1 connect()	55
7.10.2.2 disconnect()	55
7.10.2.3 read()	55
7.10.2.4 readAvailable()	56
7.10.2.5 readChunk()	56
7.10.2.6 wait()	56
7.10.2.7 write()	56
7.10.3 Member Data Documentation	56
7.10.3.1 obuffer	56
7.10.3.2 ocontext	56
7.10.3.3 odev	56
7.10.3.4 odevice	56
7.10.3.5 onum	57
7.10.3.6 oterminated	57
7.10.3.7 oterminating	57
7.10.3.8 ousbDisconnected	57
7.11 BMeasureApi::DataFile Class Reference	57
7.11.1 Constructor & Destructor Documentation	58
7.11.1.1 DataFile()	58
7.11.1.2 ~DataFile()	58
7.11.2 Member Function Documentation	58
7.11.2.1 close()	58
7.11.2.2 getFileName()	58
7.11.2.3 init()	58
7.11.2.4 open()	59
7.11.2.5 readData()	59
7.11.2.6 readInfo()	59
7.11.2.7 validateFormat()	59
7.11.2.8 writeData() [1/3]	59
7.11.2.9 writeData() [2/3]	59
7.11.2.10 writeData() [3/3]	59
7.11.2.11 writeEnd()	60
7.11.2.12 writeInfo()	60
7.11.2.13 writeInfoBMeas()	60
7.11.2.14 writeInfoCsv()	60
7.11.2.15 writeInfoTdms()	60
7.11.3 Member Data Documentation	60
7.11.3.1 ofile	60
7.11.3.2 ofileName	60
7.11.3.3 ofileType	60
7.11.3.4 oformat	61
7.11.3.5 omode	61

7.11.3.6 opacket	61
7.11.3.7 opacketLen	61
7.12 Dfu Class Reference	61
7.12.1 Detailed Description	62
7.12.2 Constructor & Destructor Documentation	62
7.12.2.1 Dfu()	62
7.12.2.2 ~Dfu()	62
7.12.3 Member Function Documentation	62
7.12.3.1 clearStatus()	62
7.12.3.2 connect()	62
7.12.3.3 detectDevice()	62
7.12.3.4 disconnect()	62
7.12.3.5 getStatus()	63
7.12.3.6 init()	63
7.12.3.7 reset()	63
7.12.3.8 upload()	63
7.12.3.9 upload_cmd()	63
7.12.3.10 validateFile()	63
7.12.4 Member Data Documentation	63
7.12.4.1 oconnected	63
7.12.4.2 ocontext	64
7.12.4.3 odev	64
7.12.4.4 overbose	64
7.13 DfuStatus Struct Reference	64
7.13.1 Member Data Documentation	64
7.13.1.1 iString	64
7.13.1.2 pollTimeout	64
7.13.1.3 state	64
7.13.1.4 status	64
8 File Documentation	65
8.1 BMdns.cpp File Reference	65
8.1.1 Macro Definition Documentation	66
8.1.1.1 BDEBUGL1	66
8.1.2 Enumeration Type Documentation	66
8.1.2.1 MdnsClass	66
8.1.2.2 MdnsEntryType	66
8.1.2.3 MdnsRecordType	66
8.1.3 Function Documentation	66
8.1.3.1 mdns_read_string()	66
8.1.3.2 mdns_read_strings()	67
8.1.3.3 mdns_write_string()	67

8.2 BMDns.h File Reference	67
8.3 BMeasureLib.cpp File Reference	67
8.3.1 Macro Definition Documentation	68
8.3.1.1 BDEBUGL1	68
8.3.1.2 BDEBUGL2	68
8.3.2 Function Documentation	68
8.3.2.1 toBStringJson() [1/3]	68
8.3.2.2 toBStringJson() [2/3]	68
8.3.2.3 toBStringJson() [3/3]	68
8.4 BMeasureLib.h File Reference	68
8.4.1 Function Documentation	69
8.4.1.1 toBStringJson() [1/3]	69
8.4.1.2 toBStringJson() [2/3]	69
8.4.1.3 toBStringJson() [3/3]	69
8.5 BMeasureUnit.cpp File Reference	69
8.5.1 Macro Definition Documentation	70
8.5.1.1 BDEBUGL1	70
8.5.1.2 BDEBUGL2	70
8.5.1.3 BDEBUGL3	70
8.5.1.4 CONVERT_FLOAT	70
8.6 BMeasureUnit.h File Reference	70
8.7 BMeasureUnits.cpp File Reference	71
8.7.1 Macro Definition Documentation	71
8.7.1.1 BDEBUGL1	71
8.7.1.2 BDEBUGL2	71
8.7.1.3 BDEBUGL3	71
8.8 BMeasureUnits.h File Reference	72
8.9 CommsNet.cpp File Reference	72
8.9.1 Macro Definition Documentation	72
8.9.1.1 BDEBUGL1	72
8.9.1.2 BDEBUGL2	72
8.9.1.3 BDEBUGL3	73
8.10 CommsNet.h File Reference	73
8.11 CommsSerial.cpp File Reference	73
8.12 CommsSerial.h File Reference	73
8.13 CommsUsb.cpp File Reference	73
8.13.1 Macro Definition Documentation	74
8.13.1.1 BDEBUGL1	74
8.13.1.2 BDEBUGL2	74
8.14 CommsUsb.h File Reference	74
8.15 DataFile.cpp File Reference	74
8.15.1 Macro Definition Documentation	75

8.15.1.1 BDEBUGL1	75
8.15.1.2 BDEBUGL2	75
8.16 DataFile.h File Reference	76
8.17 Dfu.cpp File Reference	76
8.17.1 Macro Definition Documentation	77
8.17.1.1 BDEBUGL1	77
8.17.1.2 BDEBUGL2	77
8.17.1.3 DFU_ABORT	78
8.17.1.4 DFU_CLRSTATUS	78
8.17.1.5 DFU_DETACH	78
8.17.1.6 DFU_DNLOAD	78
8.17.1.7 DFU_GETSTATE	78
8.17.1.8 DFU_GETSTATUS	78
8.17.1.9 DFU_IFF_ALT	78
8.17.1.10 DFU_IFF_CONFIG	78
8.17.1.11 DFU_IFF_DEVNUM	78
8.17.1.12 DFU_IFF_DFU	78
8.17.1.13 DFU_IFF_IFACE	79
8.17.1.14 DFU_IFF_PATH	79
8.17.1.15 DFU_IFF_PRODUCT	79
8.17.1.16 DFU_IFF_VENDOR	79
8.17.1.17 DFU_STATUS_ERROR_ADDRESS	79
8.17.1.18 DFU_STATUS_ERROR_CHECK_ERASED	79
8.17.1.19 DFU_STATUS_ERROR_ERASE	79
8.17.1.20 DFU_STATUS_ERROR_FILE	79
8.17.1.21 DFU_STATUS_ERROR_FIRMWARE	79
8.17.1.22 DFU_STATUS_ERROR_NOTDONE	79
8.17.1.23 DFU_STATUS_ERROR_POR	80
8.17.1.24 DFU_STATUS_ERROR_PROG	80
8.17.1.25 DFU_STATUS_ERROR_STALLEDPKT	80
8.17.1.26 DFU_STATUS_ERROR_TARGET	80
8.17.1.27 DFU_STATUS_ERROR_UNKNOWN	80
8.17.1.28 DFU_STATUS_ERROR_USBR	80
8.17.1.29 DFU_STATUS_ERROR_VENDOR	80
8.17.1.30 DFU_STATUS_ERROR_VERIFY	80
8.17.1.31 DFU_STATUS_ERROR_WRITE	80
8.17.1.32 DFU_STATUS_OK	80
8.17.1.33 DFU_UPLOAD	81
8.17.1.34 STATE_APP_DETACH	81
8.17.1.35 STATE_APP_IDLE	81
8.17.1.36 STATE_DFU_DOWNLOAD_BUSY	81
8.17.1.37 STATE_DFU_DOWNLOAD_IDLE	81

8.17.1.38 STATE_DFU_DOWNLOAD_SYNC	81
8.17.1.39 STATE_DFU_ERROR	81
8.17.1.40 STATE_DFU_IDLE	81
8.17.1.41 STATE_DFU_MANIFEST	81
8.17.1.42 STATE_DFU_MANIFEST_SYNC	81
8.17.1.43 STATE_DFU_MANIFEST_WAIT_RESET	82
8.17.1.44 STATE_DFU_UPLOAD_IDLE	82
8.17.2 Enumeration Type Documentation	82
8.17.2.1 dfuse_command	82
8.17.3 Function Documentation	82
8.17.3.1 pageAddress()	82
8.17.3.2 pageNumber()	82
8.17.4 Variable Documentation	82
8.17.4.1 BFirmwareInfoEncrypt1	82
8.17.4.2 BFirmwareInfoMagic	82
8.18 Dfu.h File Reference	83
8.19 overview.dox File Reference	83
Index	85

Chapter 1

BMeasure-lib

Author

Dr Terry Barnaby

Version

1.3.0

Date

2024-12-19

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 BMeasure 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");
    }
    device = devices[0].device;

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

    //printf("Exit\n"); return err;

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

    printf("NumChannels: %d\n", info.numChannels);

    //printf("Exit\n"); return err;

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

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

    printf("DataBlock: from: %d numChannels: %d numSamples: %d\n", data.source, data.numChannels,
    data.numSamples);
    for(c = 0; c < data.numChannels; c++){
        printf("%f ", data.data[c]);
    }
    printf("\n");

    return err;
}

int main(){

```

```

    BError    err;

    if(err = test1()){
        printf("Error: %d %s\n", err.getErrorNo(), err.str());
        return 1;
    }

    printf("Complete\n");

    return 0;
}

```

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

```
#!/usr/bin/python3
```

```

import sys
import time
import getopt
from threading import Thread
from bmeasure import *

# Function to read some data
def test1():
    bmeasure = BMeasureUnit(True);

    print("Find BMeasure units");
    (err, devices) = BMeasureUnit.findDevicesUsb();
    if(err):
        return err;

    if(devices.number() == 0):
        return err.set(1, "No USB BMeasure units found\n");

    print("Found", len(devices));
    device = devices[0].device;

    print("Start Processing Task");
    bmeasure.start();

    print("Connect to BMeasure");
    err = bmeasure.connect(device);
    if(err):
        return err;

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

    print("NumChannels: ", info.numChannels);

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

    print("Run single measurement");
    (err, data) = bmeasure.measure();
    if(err):
        return err;

    print("DataBlock: from: %d numChannels: %d numSamples: %d" % (data.source, data.numChannels,
data.numSamples));
    for c in range(0, data.numChannels):
        print("Chan:", c, data.data[c]);

    return err;

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

    print("Complete");

```

```

        return 0;

if __name__ == "__main__":
    main();

```

Simple example to show operating the relays in Python

```

#!/usr/bin/python3

import sys
import time
import getopt
from threading import Thread
from bmeasure import *

# Function to set the relays on/off
def test1():
    bmeasure = BMeasureUnit(True);

    print("Find BMeasure units");
    (err, devices) = BMeasureUnit.findDevicesUsb();
    if(err):
        return err;

    if(devices.number() == 0):
        return err.set(1, "No USB BMeasure units found\n");

    print("Found", len(devices));
    device = devices[0].device;

    print("Start Communications Task");
    bmeasure.start();

    print("Connect");
    err = bmeasure.connect(device);
    if(err):
        return err;

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

    print("NumChannels: ", info.numChannels);

    # Toggle relay1
    state = 0;
    for i in range(0, 6):
        if(state):
            state = False;
        else:
            state = True;

        print("Set relay 0: %d" % (state));
        err = bmeasure.setRelay(0, state);
        if(err):
            return err;

        time.sleep(1);

    return err;

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

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

    print("Complete");

    return 0;

if __name__ == "__main__":
    main();

```


Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

[BMeasureApi](#) 15

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

BComms [external]	
BMeasureApi::CommsNet	47
BMeasureApi::CommsSerial	50
BMeasureApi::CommsUsb	53
BMdns	19
BMdnsService	20
BMeasure	
BMeasureApi::BMeasureUnit	21
BMeasureApi::BMeasureUnit1	27
BMeasureApi::BMeasureUnitDevice	31
BMeasureApi::BMeasureUnitsDataBlock	46
BTask [external]	
BMeasureApi::BMeasureUnit	21
BMeasureApi::BMeasureUnits	32
BMeasureApi::DataFile	57
Dfu	61
DfuStatus	64

Chapter 4

Class Index

4.1 Class List

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

BMdns	19
BMdnsService	20
BMeasureApi::BMeasureUnit	21
BMeasureApi::BMeasureUnit1	27
BMeasureApi::BMeasureUnitDevice	31
BMeasureApi::BMeasureUnits	32
BMeasureApi::BMeasureUnitsDataBlock	46
BMeasureApi::CommsNet	47
BMeasureApi::CommsSerial	50
BMeasureApi::CommsUsb	53
BMeasureApi::DataFile	57
Dfu	
The Dfu access class	61
DfuStatus	64

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

BMdns.cpp	65
BMdns.h	67
BMeasureLib.cpp	67
BMeasureLib.h	68
BMeasureUnit.cpp	69
BMeasureUnit.h	70
BMeasureUnits.cpp	71
BMeasureUnits.h	72
CommsNet.cpp	72
CommsNet.h	73
CommsSerial.cpp	73
CommsSerial.h	73
CommsUsb.cpp	73
CommsUsb.h	74
DataFile.cpp	74
DataFile.h	76
Dfu.cpp	76
Dfu.h	83

Chapter 6

Namespace Documentation

6.1 BMeasureApi Namespace Reference

Classes

- class [BMeasureUnit](#)
- class [BMeasureUnit1](#)
- class [BMeasureUnitDevice](#)
- class [BMeasureUnits](#)
- class [BMeasureUnitsDataBlock](#)
- class [CommsNet](#)
- class [CommsSerial](#)
- class [CommsUsb](#)
- class [DataFile](#)

Typedefs

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

Enumerations

- enum [TdsDataType](#) {
 [TdsTypeVoid](#) , [TdsType18](#) , [TdsType16](#) , [TdsType32](#) ,
 [TdsType64](#) , [TdsTypeU8](#) , [TdsTypeU16](#) , [TdsTypeU32](#) ,
 [TdsTypeU64](#) , [TdsTypeSingleFloat](#) , [TdsTypeDoubleFloat](#) , [TdsTypeExtendedFloat](#) ,
 [TdsTypeSingleFloatWithUnit](#) =0x19 , [TdsTypeDoubleFloatWithUnit](#) , [TdsTypeExtendedFloatWithUnit](#) ,
 [TdsTypeString](#) =0x20 ,
 [TdsTypeBoolean](#) =0x21 , [TdsTypeTimeStamp](#) =0x44 , [TdsTypeFixedPoint](#) =0x4F , [TdsTypeComplexSingleFloat](#)
 =0x08000c ,
 [TdsTypeComplexDoubleFloat](#) =0x10000d , [TdsTypeDAQmxRawData](#) =0xFFFFFFFF }

Functions

- 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)

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 TdsDataType

```
enum BMeasureApi::TdsDataType
```

Enumerator

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

6.1.3 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]
```


Chapter 7

Class Documentation

7.1 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](#)

7.1.1 Constructor & Destructor Documentation

7.1.1.1 BMdns()

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

7.1.1.2 ~BMdns()

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

7.1.2 Member Function Documentation

7.1.2.1 findServices()

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

Unicast response, class IN

7.1.2.2 init()

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

7.1.3 Member Data Documentation

7.1.3.1 osocket

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

7.1.3.2 otransactionId

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

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

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

7.2 BMdnsService Class Reference

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

Public Attributes

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

7.2.1 Member Data Documentation

7.2.1.1 address

```
BSocketAddressINET BMdnsService::address
```

7.2.1.2 extra

```
BStringList BMdnsService::extra
```

7.2.1.3 hostname

```
BString BMdnsService::hostname
```

7.2.1.4 name

```
BString BMdnsService::name
```

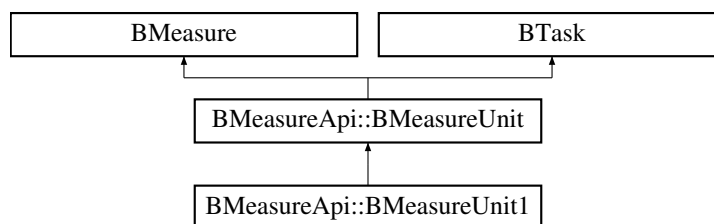
The documentation for this class was generated from the following file:

- [BMdns.h](#)

7.3 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 [sendDataFloatServe](#) (const DataBlockFloat &dataBlockFloat)
- virtual void [sendDataProcServe](#) (const DataBlockProc &dataBlockProc)
- virtual void [sendStreamDataServe](#) (const StreamData &streamData)
- virtual **BError** [setMeasurementConfig](#) (const **Bool** &save, const MeasurementConfig &configMeasurement)
- virtual **BError** [setChannelConfig](#) (const **BUInt8** &channelNumber, const ChannelConfig &channelConfig)
- **BError** [getNodeInfo](#) (NodeInfo &nodeInfo)
 - Override getNodeInfo.*

Public Member Functions inherited from BTask

- **BTask** (const char *name="", **BUInt** stackSize=0, **BUInt** priority=1)
- virtual **~BTask** ()
- void **init** (const char *name, **BUInt** stackSize=0, **BUInt** priority=1)
- **BError start** ()
- void **stop** ()
- void **waitForCompletion** ()
- int **setPriority** (**BUInt** priority)

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, **DataBlockFloat** *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**
- **DataBlockFloat** * **odataBlockFloat**
- **BUInt32** **osequenceNext**
- **BUInt32** **osampleCount**
- **BUInt32** **oblockCount**
- **Bool** **odisconnecting**
- **BSemaphoreBool** **oprocEnable**
Enable processing.
- **BSemaphoreBool** **oprocRunning**
Processing is running.

Protected Attributes inherited from BTask

- const char * **oname**
- **BUInt** **ostackSize**
- **BUInt** **opolicy**
- **BUInt** **opriority**
- **pthread_t** **othread**
- **Bool** **orunning**

Additional Inherited Members

Static Protected Member Functions inherited from BTask

- static void * **taskFunc** (void *)

7.3.1 Constructor & Destructor Documentation

7.3.1.1 BMeasureUnit()

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

7.3.1.2 ~BMeasureUnit()

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

7.3.2 Member Function Documentation

7.3.2.1 connect()

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

Connect to a device.

7.3.2.2 device()

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

7.3.2.3 disconnect()

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

7.3.2.4 disconnected()

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

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

7.3.2.5 findDevices()

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

Find available devices.

7.3.2.6 findDevicesNetwork()

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

Find available devices on Network.

7.3.2.7 findDevicesUsb()

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

Find available devices on USB bus.

7.3.2.8 getNodeInfo()

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

Override getNodeInfo.

7.3.2.9 info()

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

7.3.2.10 numChannels()

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

The number of channels of data.

7.3.2.11 processDataBlock()

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

7.3.2.12 run()

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

Threaded run mode.

Reimplemented from **BTask**.

7.3.2.13 sendDataFloatServe()

```
void BMeasureApi::BMeasureUnit::sendDataFloatServe (  
    const DataBlockFloat & dataBlockFloat) [virtual]
```

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

7.3.2.14 sendDataProcServe()

```
void BMeasureApi::BMeasureUnit::sendDataProcServe (  
    const DataBlockProc & dataBlockProc) [virtual]
```

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

7.3.2.15 sendDataServe()

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

7.3.2.16 sendStreamDataServe()

```
void BMeasureApi::BMeasureUnit::sendStreamDataServe (  
    const StreamData & streamData) [virtual]
```

7.3.2.17 serialNumber()

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

7.3.2.18 setChannelConfig()

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

7.3.2.19 setMeasurementConfig()

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

7.3.3 Member Data Documentation

7.3.3.1 blockNumChannels

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

7.3.3.2 blockNumSamples

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

7.3.3.3 oblockCount

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

7.3.3.4 ochannels

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

7.3.3.5 oconfigMeasurement

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

7.3.3.6 odataBlockFloat

```
DataBlockFloat* BMeasureApi::BMeasureUnit::odataBlockFloat [protected]
```

7.3.3.7 odevice

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

7.3.3.8 odisconnecting

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

7.3.3.9 oinfo

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

Instrument info.

7.3.3.10 onodeInfo

NodeInfo BMeasureApi::BMeasureUnit::onodeInfo [protected]

7.3.3.11 oprocEnable

BSemaphoreBool BMeasureApi::BMeasureUnit::oprocEnable [protected]

Enable processing.

7.3.3.12 oprocRunning

BSemaphoreBool BMeasureApi::BMeasureUnit::oprocRunning [protected]

Processing is running.

7.3.3.13 osampleCount

BUInt32 BMeasureApi::BMeasureUnit::osampleCount [protected]

7.3.3.14 osequenceNext

BUInt32 BMeasureApi::BMeasureUnit::osequenceNext [protected]

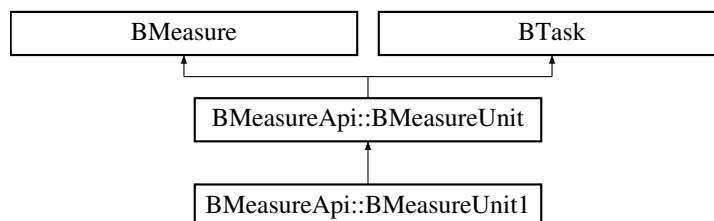
The documentation for this class was generated from the following files:

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

7.4 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 [sendStatusServe](#) (const NodeStatus &nodeStatus)
- void [sendMessageServe](#) (const **BUInt32** &source, const **BString** &message)
- void [sendDataFloatServe](#) (const DataBlockFloat &dataBlockFloat)
- void [sendDataProcServe](#) (const DataBlockProc &dataBlockProc)

Public Member Functions inherited from [BMeasureApi::BMeasureUnit](#)

- [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 [sendDataServe](#) (const DataBlock &dataBlock)
- virtual void [sendStreamDataServe](#) (const StreamData &streamData)
- virtual **BError** [setMeasurementConfig](#) (const **Bool** &save, const MeasurementConfig &configMeasurement)
- virtual **BError** [setChannelConfig](#) (const **BUInt8** &channelNumber, const ChannelConfig &channelConfig)
- **BError** [getNodeInfo](#) (NodeInfo &nodeInfo)
 - *Override getNodeInfo.*

Public Member Functions inherited from [BTask](#)

- [BTask](#) (const char *name="", **BUInt** stackSize=0, **BUInt** priority=1)
- virtual [~BTask](#) ()
- void [init](#) (const char *name, **BUInt** stackSize=0, **BUInt** priority=1)
- **BError** [start](#) ()
- void [stop](#) ()
- void [waitForCompletion](#) ()
- int [setPriority](#) (**BUInt** priority)

Public Attributes

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

Additional Inherited Members

Static Public Member Functions inherited from [BMeasureApi::BMeasureUnit](#)

- 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, DataBlockFloat *dataBlockOut)

Static Public Attributes inherited from [BMeasureApi::BMeasureUnit](#)

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

Static Protected Member Functions inherited from [BTask](#)

- static void * [taskFunc](#) (void *)

Protected Attributes inherited from [BMeasureApi::BMeasureUnit](#)

- **BString** [odevice](#)
- NodeInfo [onodeInfo](#)
- Information [oinfo](#)
Instrument info.
- MeasurementConfig [oconfigMeasurement](#)
- **BArray**< ChannelConfig > [ochannels](#)
- DataBlockFloat * [odataBlockFloat](#)
- **BUInt32** [osequenceNext](#)
- **BUInt32** [osampleCount](#)
- **BUInt32** [oblockCount](#)
- **Bool** [odisconnecting](#)
- **BSemaphoreBool** [oprocEnable](#)
Enable processing.
- **BSemaphoreBool** [oprocRunning](#)
Processing is running.

Protected Attributes inherited from [BTask](#)

- const char * [oname](#)
- **BUInt** [ostackSize](#)
- **BUInt** [opolicy](#)
- **BUInt** [opriority](#)
- pthread_t [othread](#)
- **Bool** [orunning](#)

7.4.1 Constructor & Destructor Documentation

7.4.1.1 BMeasureUnit1()

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

7.4.2 Member Function Documentation

7.4.2.1 disconnected()

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

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

7.4.2.2 sendDataFloatServe()

```
void BMeasureApi::BMeasureUnit1::sendDataFloatServe (  
    const DataBlockFloat & dataBlockFloat) [virtual]
```

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

7.4.2.3 sendDataProcServe()

```
void BMeasureApi::BMeasureUnit1::sendDataProcServe (  
    const DataBlockProc & dataBlockProc) [virtual]
```

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

7.4.2.4 sendMessageServe()

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

7.4.2.5 sendStatusServe()

```
void BMeasureApi::BMeasureUnit1::sendStatusServe (  
    const NodeStatus & nodeStatus)
```

7.4.2.6 serialNumber()

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

7.4.2.7 setSerialNumber()

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

7.4.3 Member Data Documentation

7.4.3.1 oconnected

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

7.4.3.2 oenabled

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

7.4.3.3 omeasureUnits

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

7.4.3.4 oorder

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

7.4.3.5 oserialNumber

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

7.4.3.6 osource

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

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

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

7.5 BMeasureApi::BMeasureUnitDevice Class Reference

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

Public Member Functions

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

Public Attributes

- `BString serialNumber`
- `BString device`

7.5.1 Constructor & Destructor Documentation

7.5.1.1 BMeasureUnitDevice()

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

7.5.2 Member Data Documentation

7.5.2.1 device

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

7.5.2.2 serialNumber

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

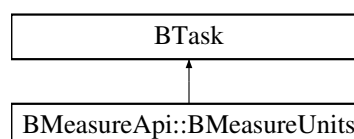
The documentation for this class was generated from the following file:

- [BMeasureUnit.h](#)

7.6 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)
- void [setMulti](#) (**Bool** on)
- **BUInt32** [unitsNum](#) ()
- **BUInt32** [unitsConnectedNum](#) ()
- [BMeasureUnit1](#) & [unit](#) (**BUInt** u)
- [BMeasureUnit1](#) & [unitMaster](#) ()
- **BError** [unitsConnect](#) ()
- **Bool** [unitsConnected](#) ()
- **BError** [unitsDisconnect](#) ()
- virtual void [disconnected](#) ()
- **BError** [unitSetOrder](#) (**BUInt** u, **BUInt** order, **Bool** move)
- **BError** [unitSetEnabled](#) (**BUInt** u, **Bool** enable)
- **BError** [dataSetNumStreams](#) (**BUInt** num)
Set the number of data output channels.
- void [dataStreamEnable](#) (**Bool** on)
Enable the streaming of data.
- void [dataClear](#) ()
- **BUInt** [dataAvailable](#) (**BUInt** stream)
- **BError** [dataWait](#) (**BUInt** stream, **BTimeout** timeoutUs= **BTimeoutForever**)
- virtual void [dataEvent](#) (**BUInt** stream)
- [DataBlockFloat](#) * [dataRead](#) (**BUInt** stream)
- void [dataDone](#) (**BUInt** stream)
- virtual void [dataProcEvent](#) (**BUInt** stream)
- [DataBlockProc](#) * [dataProcRead](#) (**BUInt** stream)
- void [dataProcDone](#) (**BUInt** stream)
- void [run](#) ()
Threaded run mode.
- void [sendDataFloatQueue](#) (const [DataBlockFloat](#) &dataBlock)
- void [sendDataProcQueue](#) (const [DataBlockProc](#) &dataBlock)
- void [sendDataProcess](#) ()
- void [sendDataProcessTrigger](#) ()
- void [outputBlock](#) ([BMeasureUnitsDataBlock](#) *block)
- virtual **BError** [getNodeInfo](#) ([NodeInfo](#) &nodeInfo)
- virtual **BError** [login](#) (const **BString** &userid, const **BString** &password)
- virtual **BError** [logout](#) ()
- virtual **BError** [changePassword](#) (const **BString** &userid, const **BString** &oldPassword, const **BString** &newPassword)
- virtual **BUInt** [numChannels](#) ()
The number of channels of data.
- virtual **BError** [setMode](#) (const **Mode** &mode)
Set the current operational mode.
- virtual **BError** [getStatus](#) ([NodeStatus](#) &nodeStatus)
- virtual void [sendTime](#) (const **BTimeUs** &time)
Sends the current time.
- virtual **BError** [getInformation](#) ([Information](#) &info)
- virtual **BError** [getInfoBlock](#) ([InfoBlock](#) &infoBlock)
- virtual **BError** [getChannelConfig](#) (const **BUInt8** &channelNumber, [ChannelConfig](#) &channelConfig)

- virtual **BError** [setChannelConfig](#) (const **BUInt8** &channelNumber, const ChannelConfig &channelConfig)
- virtual **BError** [getConfig](#) (Configuration &config)
 - *Should we have this, not generic for different instruments ?*
- virtual **BError** [setConfig](#) (const Configuration &config)
 - *Should we have this, not generic for different instruments ?*
- virtual **BError** [getMeasurementConfig](#) (const **Bool** &saved, MeasurementConfig &measurement)
 - *Get measurement config.*
- virtual **BError** [setMeasurementConfig](#) (const **Bool** &save, const MeasurementConfig &measurement)
 - *Set measurement config.*
- virtual **BError** [sendDataEnable](#) (const DataSend &dataSend)
 - *Enables the sending of data.*
- virtual **BError** [getAwgConfig](#) (const **Bool** &saved, AwgConfig &awgConfig)
 - *Get AWG Configuration.*
- virtual **BError** [setAwgConfig](#) (const **Bool** &save, const AwgConfig &awgConfig)
 - *Configure AWG.*
- virtual **BError** [alarmsClear](#) (const **BUInt32** &bits)
 - *Clear all alarms.*
- virtual void [sendStatusServe](#) (const NodeStatus &nodeStatus)
- virtual void [sendDataFloatServe](#) (const DataBlockFloat &dataBlockFloat)
- virtual void [sendDataProcServe](#) (const DataBlockProc &dataBlockProc)
- virtual void [sendMessage](#) (**BUInt32** &source, **BString** &message)
- virtual void [sendMessageServe](#) (const **BUInt32** &source, const **BString** &message)
- void [debugPrint](#) ()

Public Member Functions inherited from BTask

- **BTask** (const char *name="", **BUInt** stackSize=0, **BUInt** priority=1)
- virtual **~BTask** ()
- void **init** (const char *name, **BUInt** stackSize=0, **BUInt** priority=1)
- **BError** **start** ()
- void **stop** ()
- void **waitForCompletion** ()
- int **setPriority** (**BUInt** priority)

Private Member Functions

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

Private Attributes

- **BSemaphoreBool** [oprocEnable](#)
 - *Enable processing.*
- **BSemaphoreBool** [oprocRunning](#)
 - *Processing is running.*
- **Bool** [omulti](#)
 - *Multiple BMeasures connected together.*
- **BMutex** [olockUnits](#)
- **BList**< **BMeasureUnit1** * > [ounits](#)
- **BInt** [ounitMaster](#)
- **BUInt** [onumConnected](#)

- **BUInt** `onumChannels`
- **BUInt** `odataStreamNum`
- **BUInt32** `ofill`
- **BUInt** `onumBlocks`
- **BMutex** `olockInput`
- **BList**< **BMeasureUnitsDataBlock** * > `odataBlocksFree`
- **BList**< **BMeasureUnitsDataBlock** * > `odataBlocksIn`
- **BList**< **BMeasureUnitsDataBlock** * > `odataBlocksProcess`
- **BCondInt** `odataBlocksProcessNum`
- **BMutex** `olockOutput`
- **BList**< **BMeasureUnitsDataBlock** * > `odataBlocksOut` [2]
- **BCondInt** `odataBlocksOutCount` [2]
- **BMutex** `olockProcInput`
- **BList**< **DataBlockProc** > `odataProcBlocks`
- **MeasurementConfig** `olocalTrigger`
- **Bool** `otriggered`
- **BUInt** `ostartSample`

Additional Inherited Members

Static Protected Member Functions inherited from **BTask**

- static void * `taskFunc` (void *)

Protected Attributes inherited from **BTask**

- const char * `oname`
- **BUInt** `ostackSize`
- **BUInt** `opolicy`
- **BUInt** `opriority`
- pthread_t `othread`
- **Bool** `orunning`

7.6.1 Constructor & Destructor Documentation

7.6.1.1 **BMeasureUnits**()

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

7.6.1.2 **~BMeasureUnits**()

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

7.6.2 Member Function Documentation

7.6.2.1 alarmsClear()

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

Clear all alarms.

7.6.2.2 changePassword()

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

7.6.2.3 clear()

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

7.6.2.4 dataAvailable()

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

7.6.2.5 dataClear()

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

7.6.2.6 dataDone()

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

7.6.2.7 dataEvent()

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

7.6.2.8 dataProcDone()

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

7.6.2.9 dataProcEvent()

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

7.6.2.10 dataProcRead()

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

7.6.2.11 dataRead()

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

7.6.2.12 dataSetNumStreams()

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

Set the number of data output channels.

7.6.2.13 dataStreamEnable()

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

Enable the streaming of data.

7.6.2.14 dataWait()

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

7.6.2.15 debugPrint()

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

7.6.2.16 disconnected()

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

7.6.2.17 getAwgConfig()

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

Get AWG Configuration.

7.6.2.18 getChannelConfig()

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

7.6.2.19 getConfig()

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

Should we have this, not generic for different instruments ?

7.6.2.20 getFreeBlock()

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

7.6.2.21 getInfoBlock()

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

7.6.2.22 getInformation()

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

7.6.2.23 getMeasurementConfig()

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

Get measurement config.

7.6.2.24 getNodeInfo()

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

7.6.2.25 getStatus()

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

7.6.2.26 login()

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

7.6.2.27 logout()

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

7.6.2.28 numChannels()

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

The number of channels of data.

7.6.2.29 outputBlock()

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

7.6.2.30 run()

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

Threaded run mode.

Reimplemented from **BTask**.

7.6.2.31 sendDataEnable()

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

Enables the sending of data.

7.6.2.32 sendDataFloatQueue()

```
void BMeasureApi::BMeasureUnits::sendDataFloatQueue (
    const DataBlockFloat & dataBlock)
```

7.6.2.33 sendDataFloatServe()

```
void BMeasureApi::BMeasureUnits::sendDataFloatServe (
    const DataBlockFloat & dataBlockFloat) [virtual]
```

7.6.2.34 sendDataProcess()

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

7.6.2.35 sendDataProcessTrigger()

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

7.6.2.36 sendDataProcQueue()

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

7.6.2.37 sendDataProcServe()

```
void BMeasureApi::BMeasureUnits::sendDataProcServe (
    const DataBlockProc & dataBlockProc) [virtual]
```

7.6.2.38 sendMessage()

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

7.6.2.39 sendMessageServe()

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

7.6.2.40 sendStatusServe()

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

7.6.2.41 sendTime()

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

Sends the current time.

7.6.2.42 setAwgConfig()

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

Configure AWG.

7.6.2.43 setChannelConfig()

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

7.6.2.44 setConfig()

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

Should we have this, not generic for different instruments ?

7.6.2.45 setMeasurementConfig()

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

Set measurement config.

7.6.2.46 setMode()

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

Set the current operational mode.

7.6.2.47 setMulti()

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

7.6.2.48 unit()

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

7.6.2.49 unitAdd()

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

7.6.2.50 unitDelete()

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

7.6.2.51 unitMaster()

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

7.6.2.52 unitsConnect()

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

7.6.2.53 unitsConnected()

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

7.6.2.54 unitsConnectedNum()

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

7.6.2.55 unitsDisconnect()

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

7.6.2.56 unitSetEnabled()

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


7.6.2.57 unitSetOrder()

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

7.6.2.58 unitsFind()

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

7.6.2.59 unitsNum()

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

7.6.3 Member Data Documentation

7.6.3.1 odataBlocksFree

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

7.6.3.2 odataBlocksIn

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

7.6.3.3 odataBlocksOut

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

7.6.3.4 odataBlocksOutCount

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

7.6.3.5 odataBlocksProcess

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

7.6.3.6 odataBlocksProcessNum

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

7.6.3.7 odataProcBlocks

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

7.6.3.8 odataStreamNum

BUInt BMeasureApi::BMeasureUnits::odataStreamNum [private]

7.6.3.9 ofill

BUInt32 BMeasureApi::BMeasureUnits::ofill [private]

7.6.3.10 olocalTrigger

MeasurementConfig BMeasureApi::BMeasureUnits::olocalTrigger [private]

7.6.3.11 olockInput

BMutex BMeasureApi::BMeasureUnits::olockInput [private]

7.6.3.12 olockOutput

BMutex BMeasureApi::BMeasureUnits::olockOutput [private]

7.6.3.13 olockProInput

BMutex BMeasureApi::BMeasureUnits::olockProcInput [private]

7.6.3.14 olockUnits

BMutex BMeasureApi::BMeasureUnits::olockUnits [private]

7.6.3.15 omulti

Bool BMeasureApi::BMeasureUnits::omulti [private]

Multiple BMeasures connected together.

7.6.3.16 onumBlocks

BUInt BMeasureApi::BMeasureUnits::onumBlocks [private]

7.6.3.17 onumChannels

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

7.6.3.18 onumConnected

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

7.6.3.19 oprocEnable

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

Enable processing.

7.6.3.20 oprocRunning

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

Processing is running.

7.6.3.21 ostartSample

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

7.6.3.22 otriggered

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

7.6.3.23 ounitMaster

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

7.6.3.24 ounits

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

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

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

7.7 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](#)

7.7.1 Constructor & Destructor Documentation

7.7.1.1 BMeasureUnitsDataBlock()

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

7.7.1.2 ~BMeasureUnitsDataBlock()

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

7.7.2 Member Function Documentation

7.7.2.1 init()

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

7.7.3 Member Data Documentation

7.7.3.1 odataBlock

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

7.7.3.2 ofill

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

7.7.3.3 oinUse

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

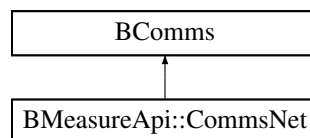
The documentation for this class was generated from the following files:

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

7.8 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)

Public Member Functions inherited from BComms

- **BComms** ()
- virtual **~BComms** ()
- virtual void **close** ()
- virtual const char * **name** ()
- virtual **BUInt32** **byteRate** ()
- virtual **BError** **setPacketMode** (**Bool** packetMode)
- virtual **Bool** **packetMode** ()
- virtual **BError** **setTimeout** (**BTimeout** timeoutUs)
- virtual **BError** **connect** (const char *resource)
- virtual **Bool** **isConnected** ()
- virtual void **flush** (**Flush** flush)
- virtual void **eventQueue** (**BEventQueue** *eventQueue, **BUInt32** event, **BUInt32** eventSet, **BUInt** num=1)
- virtual void **eventEnable** (**Bool** on)

Protected Attributes

- **B**Socket `osocket`
- **Bool** `oinWait`
- **Bool** `oterminating`

Protected Attributes inherited from **BComms**

- **Bool** `oconnected`
- **Bool** `opacketMode`
- **B**Timeout `otimeout`
- **B**EventQueue * `oeventQueue`
- **Bool** `oeventEnabled`
- **BU**Int32 `oevent`
- **BU**Int32 `oeventSet`
- **BU**Int `oeventNum`

Additional Inherited Members

Public Types inherited from **BComms**

- enum **Flush**

Public Attributes inherited from **BComms**

- **Flush**Read
- **Flush**Write
- **Flush**ReadWrite

7.8.1 Constructor & Destructor Documentation

7.8.1.1 **CommsNet**()

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

7.8.1.2 **~CommsNet**()

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

7.8.2 Member Function Documentation

7.8.2.1 **connect**()

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

7.8.2.2 disconnect()

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

Reimplemented from **BComms**.

7.8.2.3 init()

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

Reimplemented from **BComms**.

7.8.2.4 read()

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

Implements **BComms**.

7.8.2.5 readAvailable()

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

Reimplemented from **BComms**.

7.8.2.6 wait()

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

Reimplemented from **BComms**.

7.8.2.7 write()

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

Implements **BComms**.

7.8.2.8 writeAvailable()

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

Reimplemented from **BComms**.

7.8.2.9 writeChunks()

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

Reimplemented from **BComms**.

7.8.3 Member Data Documentation

7.8.3.1 oinWait

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

7.8.3.2 osocket

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

7.8.3.3 oterminating

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

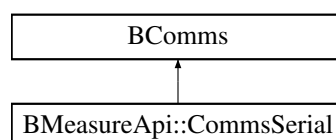
The documentation for this class was generated from the following files:

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

7.9 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)

Public Member Functions inherited from BComms

- **BComms** ()
- virtual [~BComms](#) ()
- virtual **BError** [init](#) ()
- virtual void [close](#) ()
- virtual const char * [name](#) ()
- virtual **BUInt32** [byteRate](#) ()
- virtual **BError** [setPacketMode](#) (**Bool** packetMode)
- virtual **Bool** [packetMode](#) ()
- virtual **BError** [setTimeout](#) (**BTimeout** timeoutUs)
- virtual **BError** [connect](#) (const char *resource)
- virtual **Bool** [isConnected](#) ()
- virtual void [flush](#) (**Flush** flush)
- virtual **BUInt** [writeAvailable](#) ()
- virtual **BError** [writeChunks](#) (const **BDataChunk** *chunks, **BUInt** nChunks, **BUInt32** &nTrans)
- virtual void [eventQueue](#) (**BEventQueue** *eventQueue, **BUInt32** event, **BUInt32** eventSet, **BUInt** num=1)
- virtual void [eventEnable](#) (**Bool** on)

Private Attributes

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

Additional Inherited Members**Public Types inherited from BComms**

- enum **Flush**

Public Attributes inherited from BComms

- **FlushRead**
- **FlushWrite**
- **FlushReadWrite**

Protected Attributes inherited from BComms

- Bool `oconnected`
- Bool `opacketMode`
- BTimeout `otimeout`
- BEventQueue * `oeventQueue`
- Bool `oeventEnabled`
- BUInt32 `oevent`
- BUInt32 `oeventSet`
- BUInt `oeventNum`

7.9.1 Constructor & Destructor Documentation

7.9.1.1 CommsSerial()

`BMeasureApi::CommsSerial::CommsSerial ()`

7.9.1.2 ~CommsSerial()

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

7.9.2 Member Function Documentation

7.9.2.1 connect()

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

7.9.2.2 disconnect()

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

Reimplemented from **BComms**.

7.9.2.3 read()

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

Implements **BComms**.

7.9.2.4 readAvailable()

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

Reimplemented from **BComms**.

7.9.2.5 wait()

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

Reimplemented from **BComms**.

7.9.2.6 write()

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

Implements **BComms**.

7.9.3 Member Data Documentation

7.9.3.1 odevice

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

7.9.3.2 oserialPort

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

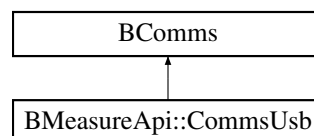
The documentation for this class was generated from the following file:

- [CommsSerial.h](#)

7.10 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)

Public Member Functions inherited from BComms

- **BComms** ()
- virtual [~BComms](#) ()
- virtual **BError** [init](#) ()
- virtual void [close](#) ()
- virtual const char * [name](#) ()
- virtual **BUInt32** [byteRate](#) ()
- virtual **BError** [setPacketMode](#) (**Bool** packetMode)
- virtual **Bool** [packetMode](#) ()
- virtual **BError** [setTimeout](#) (**BTimeout** timeoutUs)
- virtual **BError** [connect](#) (const char *resource)
- virtual **Bool** [isConnected](#) ()
- virtual void [flush](#) (**Flush** flush)
- virtual **BUInt** [writeAvailable](#) ()
- virtual **BError** [writeChunks](#) (const **BDataChunk** *chunks, **BUInt** nChunks, **BUInt32** &nTrans)
- virtual void [eventQueue](#) (**BEventQueue** *eventQueue, **BUInt32** event, **BUInt32** eventSet, **BUInt** num=1)
- virtual void [eventEnable](#) (**Bool** on)

Private Member Functions

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

Private Attributes

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

Additional Inherited Members

Public Types inherited from BComms

- enum **Flush**

Public Attributes inherited from BComms

- FlushRead
- FlushWrite
- FlushReadWrite

Protected Attributes inherited from BComms

- Bool oconnected
- Bool opacketMode
- BTimeout otimeout
- BEventQueue * oeventQueue
- Bool oeventEnabled
- BUInt32 oevent
- BUInt32 oeventSet
- BUInt oeventNum

7.10.1 Constructor & Destructor Documentation

7.10.1.1 CommsUsb()

BMeasureApi::CommsUsb::CommsUsb ()

7.10.1.2 ~CommsUsb()

BMeasureApi::CommsUsb::~~CommsUsb ()

7.10.2 Member Function Documentation

7.10.2.1 connect()

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

7.10.2.2 disconnect()

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

Reimplemented from **BComms**.

7.10.2.3 read()

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

Implements **BComms**.

7.10.2.4 readAvailable()

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

Reimplemented from **BComms**.

7.10.2.5 readChunk()

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

7.10.2.6 wait()

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

Reimplemented from **BComms**.

7.10.2.7 write()

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

Implements **BComms**.

7.10.3 Member Data Documentation

7.10.3.1 obuffer

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

7.10.3.2 ocontext

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

7.10.3.3 odev

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

7.10.3.4 odevice

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

7.10.3.5 onum

BUInt BMeasureApi::CommsUsb::onum [private]

7.10.3.6 oterminated

Bool BMeasureApi::CommsUsb::oterminated [private]

7.10.3.7 oterminating

Bool BMeasureApi::CommsUsb::oterminating [private]

7.10.3.8 ousbDisconnected

Bool BMeasureApi::CommsUsb::ousbDisconnected [private]

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

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

7.11 BMeasureApi::DataFile Class Reference

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

Public Member Functions

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

Private Member Functions

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

Private Attributes

- **BString** [ofileName](#)
- **BString** [omode](#)
- **BString** [oformat](#)
- **BUInt** [ofileType](#)
- **BFile** [ofile](#)
- **BUInt32** [opacketLen](#)
- **BoapMc1Packet** * [opacket](#)

7.11.1 Constructor & Destructor Documentation

7.11.1.1 DataFile()

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

7.11.1.2 ~DataFile()

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

7.11.2 Member Function Documentation

7.11.2.1 close()

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

Close the file.

7.11.2.2 getFileName()

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

Return the file name.

7.11.2.3 init()

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

Initialise.

7.11.2.4 open()

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

Open the file for read or write.

7.11.2.5 readData()

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

Read a block of data.

7.11.2.6 readInfo()

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

7.11.2.7 validateFormat()

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

7.11.2.8 writeData() [1/3]

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

Write a block of data.

7.11.2.9 writeData() [2/3]

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

Write a block of processed data.

7.11.2.10 writeData() [3/3]

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

Write a block of processed data.

7.11.2.11 writeEnd()

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

7.11.2.12 writeInfo()

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

7.11.2.13 writeInfoBMeas()

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

7.11.2.14 writeInfoCsv()

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

7.11.2.15 writeInfoTdms()

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

7.11.3 Member Data Documentation

7.11.3.1 ofile

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

7.11.3.2 ofileName

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

7.11.3.3 ofileType

```
BUInt BMeasureApi::DataFile::ofileType [private]
```

7.11.3.4 oformat

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

7.11.3.5 omode

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

7.11.3.6 opacket

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

7.11.3.7 opacketLen

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

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

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

7.12 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](#)

7.12.1 Detailed Description

The [Dfu](#) access class.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 Dfu()

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

7.12.2.2 ~Dfu()

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

7.12.3 Member Function Documentation

7.12.3.1 clearStatus()

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

7.12.3.2 connect()

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

Connect to USB DFU device.

7.12.3.3 detectDevice()

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

Check if DFU devuce exists.

7.12.3.4 disconnect()

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

Disconnect from USB DFU device.

7.12.3.5 getStatus()

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

7.12.3.6 init()

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

Initialise.

7.12.3.7 reset()

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

Reset.

7.12.3.8 upload()

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

Upload a file.

7.12.3.9 upload_cmd()

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

7.12.3.10 validateFile()

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

Check if file is valid firmware.

7.12.4 Member Data Documentation

7.12.4.1 oconnected

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

7.12.4.2 ocontext

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

7.12.4.3 odev

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

7.12.4.4 overbose

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

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

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

7.13 DfuStatus Struct Reference

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

Public Attributes

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

7.13.1 Member Data Documentation

7.13.1.1 iString

```
BUInt8 DfuStatus::iString
```

7.13.1.2 pollTimeout

```
BUInt DfuStatus::pollTimeout
```

7.13.1.3 state

```
BUInt8 DfuStatus::state
```

7.13.1.4 status

```
BUInt8 DfuStatus::status
```

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

- [Dfu.h](#)

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 BMeasureLib.cpp File Reference

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

Namespaces

- namespace [BMeasureApi](#)

Macros

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

Functions

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

8.3.1 Macro Definition Documentation

8.3.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.3.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.3.2 Function Documentation

8.3.2.1 toBStringJson() [1/3]

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

8.3.2.2 toBStringJson() [2/3]

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

8.3.2.3 toBStringJson() [3/3]

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

8.4 BMeasureLib.h File Reference

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

Namespaces

- namespace [BMeasureApi](#)

Typedefs

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

Functions

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

8.4.1 Function Documentation

8.4.1.1 toBStringJson() [1/3]

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

8.4.1.2 toBStringJson() [2/3]

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

8.4.1.3 toBStringJson() [3/3]

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

8.5 BMeasureUnit.cpp File Reference

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

Namespaces

- namespace [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)`

8.5.1 Macro Definition Documentation

8.5.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.5.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.5.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

8.5.1.4 CONVERT_FLOAT

```
#define CONVERT_FLOAT 0
```

Convert to floating point.

8.6 BMeasureUnit.h File Reference

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

Classes

- class `BMeasureApi::BMeasureUnitDevice`
- class `BMeasureApi::BMeasureUnit`

Namespaces

- namespace [BMeasureApi](#)

Functions

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

8.7 BMeasureUnits.cpp File Reference

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

Namespaces

- namespace [BMeasureApi](#)

Macros

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

Functions

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

8.7.1 Macro Definition Documentation

8.7.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.7.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.7.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

8.8 BMeasureUnits.h File Reference

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

Classes

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

Namespaces

- namespace [BMeasureApi](#)

8.9 CommsNet.cpp File Reference

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

Namespaces

- namespace [BMeasureApi](#)

Macros

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

8.9.1 Macro Definition Documentation

8.9.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.9.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.9.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

8.10 CommsNet.h File Reference

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

Classes

- class [BMeasureApi::CommsNet](#)

Namespaces

- namespace [BMeasureApi](#)

8.11 CommsSerial.cpp File Reference

8.12 CommsSerial.h File Reference

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

Classes

- class [BMeasureApi::CommsSerial](#)

Namespaces

- namespace [BMeasureApi](#)

8.13 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

- namespace [BMeasureApi](#)

Macros

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

Functions

- static `BUInt32 BMeasureApi::roundDown512 (BUInt32 size)`

8.13.1 Macro Definition Documentation

8.13.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.13.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.14 CommsUsb.h File Reference

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

Classes

- class [BMeasureApi::CommsUsb](#)

Namespaces

- namespace [BMeasureApi](#)

8.15 DataFile.cpp File Reference

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


Namespaces

- namespace [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 =0x40000e](#) }
 FFFFFFFF }

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)

8.15.1 Macro Definition Documentation

8.15.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.15.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.16 DataFile.h File Reference

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

Classes

- class [BMeasureApi::DataFile](#)

Namespaces

- namespace [BMeasureApi](#)

8.17 Dfu.cpp File Reference

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

Classes

- struct [BFirmwareInfo](#)

Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0
- #define [STATE_APP_IDLE](#) 0x00
- #define [STATE_APP_DETACH](#) 0x01
- #define [STATE_DFU_IDLE](#) 0x02
- #define [STATE_DFU_DOWNLOAD_SYNC](#) 0x03
- #define [STATE_DFU_DOWNLOAD_BUSY](#) 0x04
- #define [STATE_DFU_DOWNLOAD_IDLE](#) 0x05
- #define [STATE_DFU_MANIFEST_SYNC](#) 0x06
- #define [STATE_DFU_MANIFEST](#) 0x07
- #define [STATE_DFU_MANIFEST_WAIT_RESET](#) 0x08
- #define [STATE_DFU_UPLOAD_IDLE](#) 0x09
- #define [STATE_DFU_ERROR](#) 0x0a
- #define [DFU_STATUS_OK](#) 0x00
- #define [DFU_STATUS_ERROR_TARGET](#) 0x01
- #define [DFU_STATUS_ERROR_FILE](#) 0x02
- #define [DFU_STATUS_ERROR_WRITE](#) 0x03
- #define [DFU_STATUS_ERROR_ERASE](#) 0x04
- #define [DFU_STATUS_ERROR_CHECK_ERASED](#) 0x05

- `#define DFU_STATUS_ERROR_PROG 0x06`
- `#define DFU_STATUS_ERROR_VERIFY 0x07`
- `#define DFU_STATUS_ERROR_ADDRESS 0x08`
- `#define DFU_STATUS_ERROR_NOTDONE 0x09`
- `#define DFU_STATUS_ERROR_FIRMWARE 0x0a`
- `#define DFU_STATUS_ERROR_VENDOR 0x0b`
- `#define DFU_STATUS_ERROR_USBR 0x0c`
- `#define DFU_STATUS_ERROR_POR 0x0d`
- `#define DFU_STATUS_ERROR_UNKNOWN 0x0e`
- `#define DFU_STATUS_ERROR_STALLEDPKT 0x0f`
- `#define DFU_DETACH 0`
- `#define DFU_DNLOAD 1`
- `#define DFU_UPLOAD 2`
- `#define DFU_GETSTATUS 3`
- `#define DFU_CLRSTATUS 4`
- `#define DFU_GETSTATE 5`
- `#define DFU_ABORT 6`
- `#define DFU_IFF_DFU 0x0001 /* DFU Mode, (not Runtime) */`
- `#define DFU_IFF_VENDOR 0x0100`
- `#define DFU_IFF_PRODUCT 0x0200`
- `#define DFU_IFF_CONFIG 0x0400`
- `#define DFU_IFF_IFACE 0x0800`
- `#define DFU_IFF_ALT 0x1000`
- `#define DFU_IFF_DEVNUM 0x2000`
- `#define DFU_IFF_PATH 0x4000`

Enumerations

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

Functions

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

Variables

- const `BUInt32 BFirmwareInfoMagic = 0xBBEEAA00`
- const `BUInt8 BFirmwareInfoEncrypt1 = 0x40`

8.17.1 Macro Definition Documentation

8.17.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.17.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.17.1.3 DFU_ABORT

```
#define DFU_ABORT 6
```

8.17.1.4 DFU_CLRSTATUS

```
#define DFU_CLRSTATUS 4
```

8.17.1.5 DFU_DETACH

```
#define DFU_DETACH 0
```

8.17.1.6 DFU_DNLOAD

```
#define DFU_DNLOAD 1
```

8.17.1.7 DFU_GETSTATE

```
#define DFU_GETSTATE 5
```

8.17.1.8 DFU_GETSTATUS

```
#define DFU_GETSTATUS 3
```

8.17.1.9 DFU_IFF_ALT

```
#define DFU_IFF_ALT 0x1000
```

8.17.1.10 DFU_IFF_CONFIG

```
#define DFU_IFF_CONFIG 0x0400
```

8.17.1.11 DFU_IFF_DEVNUM

```
#define DFU_IFF_DEVNUM 0x2000
```

8.17.1.12 DFU_IFF_DFU

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

8.17.1.13 DFU_IFF_IFACE

```
#define DFU_IFF_IFACE 0x0800
```

8.17.1.14 DFU_IFF_PATH

```
#define DFU_IFF_PATH 0x4000
```

8.17.1.15 DFU_IFF_PRODUCT

```
#define DFU_IFF_PRODUCT 0x0200
```

8.17.1.16 DFU_IFF_VENDOR

```
#define DFU_IFF_VENDOR 0x0100
```

8.17.1.17 DFU_STATUS_ERROR_ADDRESS

```
#define DFU_STATUS_ERROR_ADDRESS 0x08
```

8.17.1.18 DFU_STATUS_ERROR_CHECK_ERASED

```
#define DFU_STATUS_ERROR_CHECK_ERASED 0x05
```

8.17.1.19 DFU_STATUS_ERROR_ERASE

```
#define DFU_STATUS_ERROR_ERASE 0x04
```

8.17.1.20 DFU_STATUS_ERROR_FILE

```
#define DFU_STATUS_ERROR_FILE 0x02
```

8.17.1.21 DFU_STATUS_ERROR_FIRMWARE

```
#define DFU_STATUS_ERROR_FIRMWARE 0x0a
```

8.17.1.22 DFU_STATUS_ERROR_NOTDONE

```
#define DFU_STATUS_ERROR_NOTDONE 0x09
```

8.17.1.23 DFU_STATUS_ERROR_POR

```
#define DFU_STATUS_ERROR_POR 0x0d
```

8.17.1.24 DFU_STATUS_ERROR_PROG

```
#define DFU_STATUS_ERROR_PROG 0x06
```

8.17.1.25 DFU_STATUS_ERROR_STALLEDPKT

```
#define DFU_STATUS_ERROR_STALLEDPKT 0x0f
```

8.17.1.26 DFU_STATUS_ERROR_TARGET

```
#define DFU_STATUS_ERROR_TARGET 0x01
```

8.17.1.27 DFU_STATUS_ERROR_UNKNOWN

```
#define DFU_STATUS_ERROR_UNKNOWN 0x0e
```

8.17.1.28 DFU_STATUS_ERROR_USBR

```
#define DFU_STATUS_ERROR_USBR 0x0c
```

8.17.1.29 DFU_STATUS_ERROR_VENDOR

```
#define DFU_STATUS_ERROR_VENDOR 0x0b
```

8.17.1.30 DFU_STATUS_ERROR_VERIFY

```
#define DFU_STATUS_ERROR_VERIFY 0x07
```

8.17.1.31 DFU_STATUS_ERROR_WRITE

```
#define DFU_STATUS_ERROR_WRITE 0x03
```

8.17.1.32 DFU_STATUS_OK

```
#define DFU_STATUS_OK 0x00
```

8.17.1.33 DFU_UPLOAD

```
#define DFU_UPLOAD 2
```

8.17.1.34 STATE_APP_DETACH

```
#define STATE_APP_DETACH 0x01
```

8.17.1.35 STATE_APP_IDLE

```
#define STATE_APP_IDLE 0x00
```

8.17.1.36 STATE_DFU_DOWNLOAD_BUSY

```
#define STATE_DFU_DOWNLOAD_BUSY 0x04
```

8.17.1.37 STATE_DFU_DOWNLOAD_IDLE

```
#define STATE_DFU_DOWNLOAD_IDLE 0x05
```

8.17.1.38 STATE_DFU_DOWNLOAD_SYNC

```
#define STATE_DFU_DOWNLOAD_SYNC 0x03
```

8.17.1.39 STATE_DFU_ERROR

```
#define STATE_DFU_ERROR 0x0a
```

8.17.1.40 STATE_DFU_IDLE

```
#define STATE_DFU_IDLE 0x02
```

8.17.1.41 STATE_DFU_MANIFEST

```
#define STATE_DFU_MANIFEST 0x07
```

8.17.1.42 STATE_DFU_MANIFEST_SYNC

```
#define STATE_DFU_MANIFEST_SYNC 0x06
```

8.17.1.43 STATE_DFU_MANIFEST_WAIT_RESET

```
#define STATE_DFU_MANIFEST_WAIT_RESET 0x08
```

8.17.1.44 STATE_DFU_UPLOAD_IDLE

```
#define STATE_DFU_UPLOAD_IDLE 0x09
```

8.17.2 Enumeration Type Documentation

8.17.2.1 dfuse_command

```
enum dfuse_command
```

Enumerator

SET_ADDRESS	
ERASE_PAGE	
MASS_ERASE	
READ_UNPROTECT	

8.17.3 Function Documentation

8.17.3.1 pageAddress()

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

8.17.3.2 pageNumber()

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

8.17.4 Variable Documentation

8.17.4.1 BFirmwareInfoEncrypt1

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

8.17.4.2 BFirmwareInfoMagic

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


8.18 Dfu.h File Reference

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

Classes

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

The [Dfu](#) access class.

8.19 overview.dox File Reference

Index

- ~BMdns
 - BMdns, [19](#)
- ~BMeasureUnit
 - BMeasureApi::BMeasureUnit, [23](#)
- ~BMeasureUnits
 - BMeasureApi::BMeasureUnits, [35](#)
- ~BMeasureUnitsDataBlock
 - BMeasureApi::BMeasureUnitsDataBlock, [46](#)
- ~CommsNet
 - BMeasureApi::CommsNet, [48](#)
- ~CommsSerial
 - BMeasureApi::CommsSerial, [52](#)
- ~CommsUsb
 - BMeasureApi::CommsUsb, [55](#)
- ~DataFile
 - BMeasureApi::DataFile, [58](#)
- ~Dfu
 - Dfu, [62](#)
- address
 - BMdnsService, [20](#)
- alarmsClear
 - BMeasureApi::BMeasureUnits, [36](#)
- BDEBUGL1
 - BMdns.cpp, [66](#)
 - BMeasureLib.cpp, [68](#)
 - BMeasureUnit.cpp, [70](#)
 - BMeasureUnits.cpp, [71](#)
 - CommsNet.cpp, [72](#)
 - CommsUsb.cpp, [74](#)
 - DataFile.cpp, [75](#)
 - Dfu.cpp, [77](#)
- BDEBUGL2
 - BMeasureLib.cpp, [68](#)
 - BMeasureUnit.cpp, [70](#)
 - BMeasureUnits.cpp, [71](#)
 - CommsNet.cpp, [72](#)
 - CommsUsb.cpp, [74](#)
 - DataFile.cpp, [75](#)
 - Dfu.cpp, [77](#)
- BDEBUGL3
 - BMeasureUnit.cpp, [70](#)
 - BMeasureUnits.cpp, [71](#)
 - CommsNet.cpp, [72](#)
- BFirmwareInfoEncrypt1
 - Dfu.cpp, [82](#)
- BFirmwareInfoMagic
 - Dfu.cpp, [82](#)
- blockNumChannels
 - BMeasureApi::BMeasureUnit, [26](#)
- blockNumSamples
 - BMeasureApi::BMeasureUnit, [26](#)
- BMdns, [19](#)
 - ~BMdns, [19](#)
 - BMdns, [19](#)
 - findServices, [20](#)
 - init, [20](#)
 - osocket, [20](#)
 - otransactionId, [20](#)
- BMdns.cpp, [65](#)
 - BDEBUGL1, [66](#)
 - MDNS_CLASS_IN, [66](#)
 - MDNS_ENTRYTYPE_ADDITIONAL, [66](#)
 - MDNS_ENTRYTYPE_ANSWER, [66](#)
 - MDNS_ENTRYTYPE_AUTHORITY, [66](#)
 - mdns_read_string, [66](#)
 - mdns_read_strings, [66](#)
 - MDNS_RECORDTYPE_A, [66](#)
 - MDNS_RECORDTYPE_AAAA, [66](#)
 - MDNS_RECORDTYPE_IGNORE, [66](#)
 - MDNS_RECORDTYPE_PTR, [66](#)
 - MDNS_RECORDTYPE_SRV, [66](#)
 - MDNS_RECORDTYPE_TXT, [66](#)
 - mdns_write_string, [67](#)
 - MdnsClass, [66](#)
 - MdnsEntryType, [66](#)
 - MdnsRecordType, [66](#)
- BMdns.h, [67](#)
- BMdnsService, [20](#)
 - address, [20](#)
 - extra, [20](#)
 - hostname, [21](#)
 - name, [21](#)
- BMeasure-lib, [1](#)
- BMeasureApi, [15](#)
 - ChannelConfigs, [16](#)
 - channelTypeString, [17](#)
 - round512, [17](#)
 - roundDown512, [17](#)
 - sampleTypeString, [17](#)
 - TdsDataType, [16](#)
 - TdsTypeBoolean, [16](#)
 - TdsTypeComplexDoubleFloat, [16](#)
 - TdsTypeComplexSingleFloat, [16](#)
 - TdsTypeDAQmxRawData, [16](#)
 - TdsTypeDoubleFloat, [16](#)
 - TdsTypeDoubleFloatWithUnit, [16](#)
 - TdsTypeExtendedFloat, [16](#)

- TdsTypeExtendedFloatWithUnit, 16
- TdsTypeFixedPoint, 16
- TdsType16, 16
- TdsType32, 16
- TdsType64, 16
- TdsType8, 16
- TdsTypeSingleFloat, 16
- TdsTypeSingleFloatWithUnit, 16
- TdsTypeString, 16
- TdsTypeTimeStamp, 16
- TdsTypeU16, 16
- TdsTypeU32, 16
- TdsTypeU64, 16
- TdsTypeU8, 16
- TdsTypeVoid, 16
- TocBigEndian, 17
- TocDaqRawData, 17
- TocInterleavedData, 17
- TocMetaData, 17
- TocNewObjList, 17
- TocRawData, 18
- toFloat, 18
- unitSort, 18
- BMeasureApi::BMeasureUnit, 21
 - ~BMeasureUnit, 23
 - blockNumChannels, 26
 - blockNumSamples, 26
 - BMeasureUnit, 23
 - connect, 23
 - device, 23
 - disconnect, 23
 - disconnected, 23
 - findDevices, 23
 - findDevicesNetwork, 24
 - findDevicesUsb, 24
 - getNodeInfo, 24
 - info, 24
 - numChannels, 24
 - oblockCount, 26
 - ochannels, 26
 - oconfigMeasurement, 26
 - odataBlockFloat, 26
 - odevice, 26
 - odisconnecting, 26
 - oinfo, 26
 - onodeInfo, 26
 - oprocEnable, 27
 - oprocRunning, 27
 - osampleCount, 27
 - osequenceNext, 27
 - processdataBlock, 24
 - run, 24
 - sendDataFloatServe, 25
 - sendDataProcServe, 25
 - sendDataServe, 25
 - sendStreamDataServe, 25
 - serialNumber, 25
 - setChannelConfig, 25
 - setMeasurementConfig, 25
- BMeasureApi::BMeasureUnit1, 27
 - BMeasureUnit1, 30
 - disconnected, 30
 - oconnected, 31
 - oenabled, 31
 - omeasureUnits, 31
 - oorder, 31
 - oserialNumber, 31
 - osource, 31
 - sendDataFloatServe, 30
 - sendDataProcServe, 30
 - sendMessageServe, 30
 - sendStatusServe, 30
 - serialNumber, 30
 - setSerialNumber, 30
- BMeasureApi::BMeasureUnitDevice, 31
 - BMeasureUnitDevice, 32
 - device, 32
 - serialNumber, 32
- BMeasureApi::BMeasureUnits, 32
 - ~BMeasureUnits, 35
 - alarmsClear, 36
 - BMeasureUnits, 35
 - changePassword, 36
 - clear, 36
 - dataAvailable, 36
 - dataClear, 36
 - dataDone, 36
 - dataEvent, 36
 - dataProcDone, 36
 - dataProcEvent, 36
 - dataProcRead, 37
 - dataRead, 37
 - dataSetNumStreams, 37
 - dataStreamEnable, 37
 - dataWait, 37
 - debugPrint, 37
 - disconnected, 37
 - getAwgConfig, 37
 - getChannelConfig, 38
 - getConfig, 38
 - getFreeBlock, 38
 - getInfoBlock, 38
 - getInformation, 38
 - getMeasurementConfig, 38
 - getNodeInfo, 38
 - getStatus, 39
 - login, 39
 - logout, 39
 - numChannels, 39
 - odataBlocksFree, 43
 - odataBlocksIn, 43
 - odataBlocksOut, 43
 - odataBlocksOutCount, 43
 - odataBlocksProcess, 43
 - odataBlocksProcessNum, 43
 - odataProcBlocks, 43

- odataStreamNum, 44
- ofill, 44
- olocalTrigger, 44
- olockInput, 44
- olockOutput, 44
- olockProcInput, 44
- olockUnits, 44
- omulti, 44
- onumBlocks, 44
- onumChannels, 44
- onumConnected, 45
- oprocEnable, 45
- oprocRunning, 45
- ostartSample, 45
- otriggered, 45
- ounitMaster, 45
- ounits, 45
- outputBlock, 39
- run, 39
- sendDataEnable, 39
- sendDataFloatQueue, 39
- sendDataFloatServe, 40
- sendDataProcess, 40
- sendDataProcessTrigger, 40
- sendDataProcQueue, 40
- sendDataProcServe, 40
- sendMessage, 40
- sendMessageServe, 40
- sendStatusServe, 40
- sendTime, 40
- setAwgConfig, 41
- setChannelConfig, 41
- setConfig, 41
- setMeasurementConfig, 41
- setMode, 41
- setMulti, 41
- unit, 41
- unitAdd, 42
- unitDelete, 42
- unitMaster, 42
- unitsConnect, 42
- unitsConnected, 42
- unitsConnectedNum, 42
- unitsDisconnect, 42
- unitSetEnabled, 42
- unitSetOrder, 42
- unitsFind, 43
- unitsNum, 43
- BMeasureApi::BMeasureUnitsDataBlock, 46
 - ~BMeasureUnitsDataBlock, 46
 - BMeasureUnitsDataBlock, 46
 - init, 46
 - odataBlock, 46
 - ofill, 46
 - oinUse, 46
- BMeasureApi::CommsNet, 47
 - ~CommsNet, 48
 - CommsNet, 48
 - connect, 48
 - disconnect, 48
 - init, 49
 - oinWait, 50
 - osocket, 50
 - oterminating, 50
 - read, 49
 - readAvailable, 49
 - wait, 49
 - write, 49
 - writeAvailable, 49
 - writeChunks, 50
- BMeasureApi::CommsSerial, 50
 - ~CommsSerial, 52
 - CommsSerial, 52
 - connect, 52
 - disconnect, 52
 - odevice, 53
 - oserialPort, 53
 - read, 52
 - readAvailable, 52
 - wait, 52
 - write, 53
- BMeasureApi::CommsUsb, 53
 - ~CommsUsb, 55
 - CommsUsb, 55
 - connect, 55
 - disconnect, 55
 - obuffer, 56
 - ocontext, 56
 - odev, 56
 - odevice, 56
 - onum, 56
 - oterminated, 57
 - oterminating, 57
 - ousbDisconnected, 57
 - read, 55
 - readAvailable, 55
 - readChunk, 56
 - wait, 56
 - write, 56
- BMeasureApi::DataFile, 57
 - ~DataFile, 58
 - close, 58
 - DataFile, 58
 - getFileName, 58
 - init, 58
 - ofile, 60
 - ofileName, 60
 - ofileType, 60
 - oformat, 60
 - omode, 61
 - opacket, 61
 - opacketLen, 61
 - open, 58
 - readData, 59
 - readInfo, 59
 - validateFormat, 59

- writeData, 59
- writeEnd, 59
- writeInfo, 60
- writeInfoBMeas, 60
- writeInfoCsv, 60
- writeInfoTdms, 60
- BMeasureLib.cpp, 67
 - BDEBUGL1, 68
 - BDEBUGL2, 68
 - toBStringJson, 68
- BMeasureLib.h, 68
 - toBStringJson, 69
- BMeasureUnit
 - BMeasureApi::BMeasureUnit, 23
- BMeasureUnit.cpp, 69
 - BDEBUGL1, 70
 - BDEBUGL2, 70
 - BDEBUGL3, 70
 - CONVERT_FLOAT, 70
- BMeasureUnit.h, 70
- BMeasureUnit1
 - BMeasureApi::BMeasureUnit1, 30
- BMeasureUnitDevice
 - BMeasureApi::BMeasureUnitDevice, 32
- BMeasureUnits
 - BMeasureApi::BMeasureUnits, 35
- BMeasureUnits.cpp, 71
 - BDEBUGL1, 71
 - BDEBUGL2, 71
 - BDEBUGL3, 71
- BMeasureUnits.h, 72
- BMeasureUnitsDataBlock
 - BMeasureApi::BMeasureUnitsDataBlock, 46
- changePassword
 - BMeasureApi::BMeasureUnits, 36
- ChannelConfigs
 - BMeasureApi, 16
- channelTypeString
 - BMeasureApi, 17
- clear
 - BMeasureApi::BMeasureUnits, 36
- clearStatus
 - Dfu, 62
- close
 - BMeasureApi::DataFile, 58
- CommsNet
 - BMeasureApi::CommsNet, 48
- CommsNet.cpp, 72
 - BDEBUGL1, 72
 - BDEBUGL2, 72
 - BDEBUGL3, 72
- CommsNet.h, 73
- CommsSerial
 - BMeasureApi::CommsSerial, 52
- CommsSerial.cpp, 73
- CommsSerial.h, 73
- CommsUsb
 - BMeasureApi::CommsUsb, 55
- CommsUsb.cpp, 73
 - BDEBUGL1, 74
 - BDEBUGL2, 74
- CommsUsb.h, 74
- connect
 - BMeasureApi::BMeasureUnit, 23
 - BMeasureApi::CommsNet, 48
 - BMeasureApi::CommsSerial, 52
 - BMeasureApi::CommsUsb, 55
 - Dfu, 62
- CONVERT_FLOAT
 - BMeasureUnit.cpp, 70
- dataAvailable
 - BMeasureApi::BMeasureUnits, 36
- dataClear
 - BMeasureApi::BMeasureUnits, 36
- dataDone
 - BMeasureApi::BMeasureUnits, 36
- dataEvent
 - BMeasureApi::BMeasureUnits, 36
- DataFile
 - BMeasureApi::DataFile, 58
- DataFile.cpp, 74
 - BDEBUGL1, 75
 - BDEBUGL2, 75
- DataFile.h, 76
- dataProcDone
 - BMeasureApi::BMeasureUnits, 36
- dataProcEvent
 - BMeasureApi::BMeasureUnits, 36
- dataProcRead
 - BMeasureApi::BMeasureUnits, 37
- dataRead
 - BMeasureApi::BMeasureUnits, 37
- dataSetNumStreams
 - BMeasureApi::BMeasureUnits, 37
- dataStreamEnable
 - BMeasureApi::BMeasureUnits, 37
- dataWait
 - BMeasureApi::BMeasureUnits, 37
- debugPrint
 - BMeasureApi::BMeasureUnits, 37
- detectDevice
 - Dfu, 62
- device
 - BMeasureApi::BMeasureUnit, 23
 - BMeasureApi::BMeasureUnitDevice, 32
- Dfu, 61
 - ~Dfu, 62
 - clearStatus, 62
 - connect, 62
 - detectDevice, 62
 - Dfu, 62
 - disconnect, 62
 - getStatus, 62
 - init, 63
 - oconnected, 63
 - ocontext, 63

- odev, [64](#)
- overbose, [64](#)
- reset, [63](#)
- upload, [63](#)
- upload_cmd, [63](#)
- validateFile, [63](#)
- Dfu.cpp, [76](#)
 - BDEBUGL1, [77](#)
 - BDEBUGL2, [77](#)
 - BFirmwareInfoEncrypt1, [82](#)
 - BFirmwareInfoMagic, [82](#)
 - DFU_ABORT, [77](#)
 - DFU_CLRSTATUS, [78](#)
 - DFU_DETACH, [78](#)
 - DFU_DNLOAD, [78](#)
 - DFU_GETSTATE, [78](#)
 - DFU_GETSTATUS, [78](#)
 - DFU_IFF_ALT, [78](#)
 - DFU_IFF_CONFIG, [78](#)
 - DFU_IFF_DEVNUM, [78](#)
 - DFU_IFF_DFU, [78](#)
 - DFU_IFF_IFACE, [78](#)
 - DFU_IFF_PATH, [79](#)
 - DFU_IFF_PRODUCT, [79](#)
 - DFU_IFF_VENDOR, [79](#)
 - DFU_STATUS_ERROR_ADDRESS, [79](#)
 - DFU_STATUS_ERROR_CHECK_ERASED, [79](#)
 - DFU_STATUS_ERROR_ERASE, [79](#)
 - DFU_STATUS_ERROR_FILE, [79](#)
 - DFU_STATUS_ERROR_FIRMWARE, [79](#)
 - DFU_STATUS_ERROR_NOTDONE, [79](#)
 - DFU_STATUS_ERROR_POR, [79](#)
 - DFU_STATUS_ERROR_PROG, [80](#)
 - DFU_STATUS_ERROR_STALLEDPKT, [80](#)
 - DFU_STATUS_ERROR_TARGET, [80](#)
 - DFU_STATUS_ERROR_UNKNOWN, [80](#)
 - DFU_STATUS_ERROR_USBR, [80](#)
 - DFU_STATUS_ERROR_VENDOR, [80](#)
 - DFU_STATUS_ERROR_VERIFY, [80](#)
 - DFU_STATUS_ERROR_WRITE, [80](#)
 - DFU_STATUS_OK, [80](#)
 - DFU_UPLOAD, [80](#)
 - dfuse_command, [82](#)
 - ERASE_PAGE, [82](#)
 - MASS_ERASE, [82](#)
 - pageAddress, [82](#)
 - pageNumber, [82](#)
 - READ_UNPROTECT, [82](#)
 - SET_ADDRESS, [82](#)
 - STATE_APP_DETACH, [81](#)
 - STATE_APP_IDLE, [81](#)
 - STATE_DFU_DOWNLOAD_BUSY, [81](#)
 - STATE_DFU_DOWNLOAD_IDLE, [81](#)
 - STATE_DFU_DOWNLOAD_SYNC, [81](#)
 - STATE_DFU_ERROR, [81](#)
 - STATE_DFU_IDLE, [81](#)
 - STATE_DFU_MANIFEST, [81](#)
 - STATE_DFU_MANIFEST_SYNC, [81](#)
 - STATE_DFU_MANIFEST_WAIT_RESET, [81](#)
 - STATE_DFU_UPLOAD_IDLE, [82](#)
- Dfu.h, [83](#)
- DFU_ABORT
 - Dfu.cpp, [77](#)
- DFU_CLRSTATUS
 - Dfu.cpp, [78](#)
- DFU_DETACH
 - Dfu.cpp, [78](#)
- DFU_DNLOAD
 - Dfu.cpp, [78](#)
- DFU_GETSTATE
 - Dfu.cpp, [78](#)
- DFU_GETSTATUS
 - Dfu.cpp, [78](#)
- DFU_IFF_ALT
 - Dfu.cpp, [78](#)
- DFU_IFF_CONFIG
 - Dfu.cpp, [78](#)
- DFU_IFF_DEVNUM
 - Dfu.cpp, [78](#)
- DFU_IFF_DFU
 - Dfu.cpp, [78](#)
- DFU_IFF_IFACE
 - Dfu.cpp, [78](#)
- DFU_IFF_PATH
 - Dfu.cpp, [79](#)
- DFU_IFF_PRODUCT
 - Dfu.cpp, [79](#)
- DFU_IFF_VENDOR
 - Dfu.cpp, [79](#)
- DFU_STATUS_ERROR_ADDRESS
 - Dfu.cpp, [79](#)
- DFU_STATUS_ERROR_CHECK_ERASED
 - Dfu.cpp, [79](#)
- DFU_STATUS_ERROR_ERASE
 - Dfu.cpp, [79](#)
- DFU_STATUS_ERROR_FILE
 - Dfu.cpp, [79](#)
- DFU_STATUS_ERROR_FIRMWARE
 - Dfu.cpp, [79](#)
- DFU_STATUS_ERROR_NOTDONE
 - Dfu.cpp, [79](#)
- DFU_STATUS_ERROR_POR
 - Dfu.cpp, [79](#)
- DFU_STATUS_ERROR_PROG
 - Dfu.cpp, [80](#)
- DFU_STATUS_ERROR_STALLEDPKT
 - Dfu.cpp, [80](#)
- DFU_STATUS_ERROR_TARGET
 - Dfu.cpp, [80](#)
- DFU_STATUS_ERROR_UNKNOWN
 - Dfu.cpp, [80](#)
- DFU_STATUS_ERROR_USBR
 - Dfu.cpp, [80](#)
- DFU_STATUS_ERROR_VENDOR
 - Dfu.cpp, [80](#)
- DFU_STATUS_ERROR_VERIFY

- Dfu.cpp, 80
- DFU_STATUS_ERROR_WRITE
 - Dfu.cpp, 80
- DFU_STATUS_OK
 - Dfu.cpp, 80
- DFU_UPLOAD
 - Dfu.cpp, 80
- dfuse_command
 - Dfu.cpp, 82
- DfuStatus, 64
 - iString, 64
 - pollTimeout, 64
 - state, 64
 - status, 64
- disconnect
 - BMeasureApi::BMeasureUnit, 23
 - BMeasureApi::CommsNet, 48
 - BMeasureApi::CommsSerial, 52
 - BMeasureApi::CommsUsb, 55
 - Dfu, 62
- disconnected
 - BMeasureApi::BMeasureUnit, 23
 - BMeasureApi::BMeasureUnit1, 30
 - BMeasureApi::BMeasureUnits, 37
- ERASE_PAGE
 - Dfu.cpp, 82
- extra
 - BMdnsService, 20
- findDevices
 - BMeasureApi::BMeasureUnit, 23
- findDevicesNetwork
 - BMeasureApi::BMeasureUnit, 24
- findDevicesUsb
 - BMeasureApi::BMeasureUnit, 24
- findServices
 - BMdns, 20
- getAwgConfig
 - BMeasureApi::BMeasureUnits, 37
- getChannelConfig
 - BMeasureApi::BMeasureUnits, 38
- getConfig
 - BMeasureApi::BMeasureUnits, 38
- getFileName
 - BMeasureApi::DataFile, 58
- getFreeBlock
 - BMeasureApi::BMeasureUnits, 38
- getInfoBlock
 - BMeasureApi::BMeasureUnits, 38
- getInformation
 - BMeasureApi::BMeasureUnits, 38
- getMeasurementConfig
 - BMeasureApi::BMeasureUnits, 38
- getNodeInfo
 - BMeasureApi::BMeasureUnit, 24
 - BMeasureApi::BMeasureUnits, 38
- getStatus
 - BMeasureApi::BMeasureUnits, 39
 - Dfu, 62
- hostname
 - BMdnsService, 21
- info
 - BMeasureApi::BMeasureUnit, 24
- init
 - BMdns, 20
 - BMeasureApi::BMeasureUnitsDataBlock, 46
 - BMeasureApi::CommsNet, 49
 - BMeasureApi::DataFile, 58
 - Dfu, 63
- iString
 - DfuStatus, 64
- login
 - BMeasureApi::BMeasureUnits, 39
- logout
 - BMeasureApi::BMeasureUnits, 39
- MASS_ERASE
 - Dfu.cpp, 82
- MDNS_CLASS_IN
 - BMdns.cpp, 66
- MDNS_ENTRYTYPE_ADDITIONAL
 - BMdns.cpp, 66
- MDNS_ENTRYTYPE_ANSWER
 - BMdns.cpp, 66
- MDNS_ENTRYTYPE_AUTHORITY
 - BMdns.cpp, 66
- mdns_read_string
 - BMdns.cpp, 66
- mdns_read_strings
 - BMdns.cpp, 66
- MDNS_RECORDTYPE_A
 - BMdns.cpp, 66
- MDNS_RECORDTYPE_AAAA
 - BMdns.cpp, 66
- MDNS_RECORDTYPE_IGNORE
 - BMdns.cpp, 66
- MDNS_RECORDTYPE_PTR
 - BMdns.cpp, 66
- MDNS_RECORDTYPE_SRV
 - BMdns.cpp, 66
- MDNS_RECORDTYPE_TXT
 - BMdns.cpp, 66
- mdns_write_string
 - BMdns.cpp, 67
- MdnsClass
 - BMdns.cpp, 66
- MdnsEntryType
 - BMdns.cpp, 66
- MdnsRecordType
 - BMdns.cpp, 66
- name
 - BMdnsService, 21

- numChannels
 - BMeasureApi::BMeasureUnit, [24](#)
 - BMeasureApi::BMeasureUnits, [39](#)
- oblockCount
 - BMeasureApi::BMeasureUnit, [26](#)
- obuffer
 - BMeasureApi::CommsUsb, [56](#)
- ochannels
 - BMeasureApi::BMeasureUnit, [26](#)
- oconfigMeasurement
 - BMeasureApi::BMeasureUnit, [26](#)
- oconnected
 - BMeasureApi::BMeasureUnit1, [31](#)
 - Dfu, [63](#)
- ocontext
 - BMeasureApi::CommsUsb, [56](#)
 - Dfu, [63](#)
- odataBlock
 - BMeasureApi::BMeasureUnitsDataBlock, [46](#)
- odataBlockFloat
 - BMeasureApi::BMeasureUnit, [26](#)
- odataBlocksFree
 - BMeasureApi::BMeasureUnits, [43](#)
- odataBlocksIn
 - BMeasureApi::BMeasureUnits, [43](#)
- odataBlocksOut
 - BMeasureApi::BMeasureUnits, [43](#)
- odataBlocksOutCount
 - BMeasureApi::BMeasureUnits, [43](#)
- odataBlocksProcess
 - BMeasureApi::BMeasureUnits, [43](#)
- odataBlocksProcessNum
 - BMeasureApi::BMeasureUnits, [43](#)
- odataProcBlocks
 - BMeasureApi::BMeasureUnits, [43](#)
- odataStreamNum
 - BMeasureApi::BMeasureUnits, [44](#)
- odev
 - BMeasureApi::CommsUsb, [56](#)
 - Dfu, [64](#)
- odevice
 - BMeasureApi::BMeasureUnit, [26](#)
 - BMeasureApi::CommsSerial, [53](#)
 - BMeasureApi::CommsUsb, [56](#)
- odisconnecting
 - BMeasureApi::BMeasureUnit, [26](#)
- oenabled
 - BMeasureApi::BMeasureUnit1, [31](#)
- ofile
 - BMeasureApi::DataFile, [60](#)
- ofileName
 - BMeasureApi::DataFile, [60](#)
- ofileType
 - BMeasureApi::DataFile, [60](#)
- ofill
 - BMeasureApi::BMeasureUnits, [44](#)
 - BMeasureApi::BMeasureUnitsDataBlock, [46](#)
- oformat
 - BMeasureApi::DataFile, [60](#)
- oinfo
 - BMeasureApi::BMeasureUnit, [26](#)
- oinUse
 - BMeasureApi::BMeasureUnitsDataBlock, [46](#)
- oinWait
 - BMeasureApi::CommsNet, [50](#)
- olocalTrigger
 - BMeasureApi::BMeasureUnits, [44](#)
- olockInput
 - BMeasureApi::BMeasureUnits, [44](#)
- olockOutput
 - BMeasureApi::BMeasureUnits, [44](#)
- olockProcInput
 - BMeasureApi::BMeasureUnits, [44](#)
- olockUnits
 - BMeasureApi::BMeasureUnits, [44](#)
- omeasureUnits
 - BMeasureApi::BMeasureUnit1, [31](#)
- omode
 - BMeasureApi::DataFile, [61](#)
- omulti
 - BMeasureApi::BMeasureUnits, [44](#)
- onodeInfo
 - BMeasureApi::BMeasureUnit, [26](#)
- onum
 - BMeasureApi::CommsUsb, [56](#)
- onumBlocks
 - BMeasureApi::BMeasureUnits, [44](#)
- onumChannels
 - BMeasureApi::BMeasureUnits, [44](#)
- onumConnected
 - BMeasureApi::BMeasureUnits, [45](#)
- oorder
 - BMeasureApi::BMeasureUnit1, [31](#)
- opacket
 - BMeasureApi::DataFile, [61](#)
- opacketLen
 - BMeasureApi::DataFile, [61](#)
- open
 - BMeasureApi::DataFile, [58](#)
- oprocEnable
 - BMeasureApi::BMeasureUnit, [27](#)
 - BMeasureApi::BMeasureUnits, [45](#)
- oprocRunning
 - BMeasureApi::BMeasureUnit, [27](#)
 - BMeasureApi::BMeasureUnits, [45](#)
- osampleCount
 - BMeasureApi::BMeasureUnit, [27](#)
- osequenceNext
 - BMeasureApi::BMeasureUnit, [27](#)
- oserialNumber
 - BMeasureApi::BMeasureUnit1, [31](#)
- oserialPort
 - BMeasureApi::CommsSerial, [53](#)
- osocket
 - BMdns, [20](#)
 - BMeasureApi::CommsNet, [50](#)

- osource
 - BMeasureApi::BMeasureUnit1, 31
- ostartSample
 - BMeasureApi::BMeasureUnits, 45
- oterminated
 - BMeasureApi::CommsUsb, 57
- oterminating
 - BMeasureApi::CommsNet, 50
 - BMeasureApi::CommsUsb, 57
- otransactionId
 - BMdns, 20
- otriggered
 - BMeasureApi::BMeasureUnits, 45
- ounitMaster
 - BMeasureApi::BMeasureUnits, 45
- ounits
 - BMeasureApi::BMeasureUnits, 45
- ousbDisconnected
 - BMeasureApi::CommsUsb, 57
- outputBlock
 - BMeasureApi::BMeasureUnits, 39
- overbose
 - Dfu, 64
- overview.dox, 83
- pageAddress
 - Dfu.cpp, 82
- pageNumber
 - Dfu.cpp, 82
- pollTimeout
 - DfuStatus, 64
- processdataBlock
 - BMeasureApi::BMeasureUnit, 24
- read
 - BMeasureApi::CommsNet, 49
 - BMeasureApi::CommsSerial, 52
 - BMeasureApi::CommsUsb, 55
- READ_UNPROTECT
 - Dfu.cpp, 82
- readAvailable
 - BMeasureApi::CommsNet, 49
 - BMeasureApi::CommsSerial, 52
 - BMeasureApi::CommsUsb, 55
- readChunk
 - BMeasureApi::CommsUsb, 56
- readData
 - BMeasureApi::DataFile, 59
- readInfo
 - BMeasureApi::DataFile, 59
- reset
 - Dfu, 63
- round512
 - BMeasureApi, 17
- roundDown512
 - BMeasureApi, 17
- run
 - BMeasureApi::BMeasureUnit, 24
 - BMeasureApi::BMeasureUnits, 39
- sampleTypeString
 - BMeasureApi, 17
- sendDataEnable
 - BMeasureApi::BMeasureUnits, 39
- sendDataFloatQueue
 - BMeasureApi::BMeasureUnits, 39
- sendDataFloatServe
 - BMeasureApi::BMeasureUnit, 25
 - BMeasureApi::BMeasureUnit1, 30
 - BMeasureApi::BMeasureUnits, 40
- sendDataProcess
 - BMeasureApi::BMeasureUnits, 40
- sendDataProcessTrigger
 - BMeasureApi::BMeasureUnits, 40
- sendDataProcQueue
 - BMeasureApi::BMeasureUnits, 40
- sendDataProcServe
 - BMeasureApi::BMeasureUnit, 25
 - BMeasureApi::BMeasureUnit1, 30
 - BMeasureApi::BMeasureUnits, 40
- sendDataServe
 - BMeasureApi::BMeasureUnit, 25
- sendMessage
 - BMeasureApi::BMeasureUnits, 40
- sendMessageServe
 - BMeasureApi::BMeasureUnit1, 30
 - BMeasureApi::BMeasureUnits, 40
- sendStatusServe
 - BMeasureApi::BMeasureUnit1, 30
 - BMeasureApi::BMeasureUnits, 40
- sendStreamDataServe
 - BMeasureApi::BMeasureUnit, 25
- sendTime
 - BMeasureApi::BMeasureUnits, 40
- serialNumber
 - BMeasureApi::BMeasureUnit, 25
 - BMeasureApi::BMeasureUnit1, 30
 - BMeasureApi::BMeasureUnitDevice, 32
- SET_ADDRESS
 - Dfu.cpp, 82
- setAwgConfig
 - BMeasureApi::BMeasureUnits, 41
- setChannelConfig
 - BMeasureApi::BMeasureUnit, 25
 - BMeasureApi::BMeasureUnits, 41
- setConfig
 - BMeasureApi::BMeasureUnits, 41
- setMeasurementConfig
 - BMeasureApi::BMeasureUnit, 25
 - BMeasureApi::BMeasureUnits, 41
- setMode
 - BMeasureApi::BMeasureUnits, 41
- setMulti
 - BMeasureApi::BMeasureUnits, 41
- setSerialNumber
 - BMeasureApi::BMeasureUnit1, 30
- state
 - DfuStatus, 64

- STATE_APP_DETACH
 - Dfu.cpp, [81](#)
- STATE_APP_IDLE
 - Dfu.cpp, [81](#)
- STATE_DFU_DOWNLOAD_BUSY
 - Dfu.cpp, [81](#)
- STATE_DFU_DOWNLOAD_IDLE
 - Dfu.cpp, [81](#)
- STATE_DFU_DOWNLOAD_SYNC
 - Dfu.cpp, [81](#)
- STATE_DFU_ERROR
 - Dfu.cpp, [81](#)
- STATE_DFU_IDLE
 - Dfu.cpp, [81](#)
- STATE_DFU_MANIFEST
 - Dfu.cpp, [81](#)
- STATE_DFU_MANIFEST_SYNC
 - Dfu.cpp, [81](#)
- STATE_DFU_MANIFEST_WAIT_RESET
 - Dfu.cpp, [81](#)
- STATE_DFU_UPLOAD_IDLE
 - Dfu.cpp, [82](#)
- status
 - DfuStatus, [64](#)
- TdsDataType
 - BMeasureApi, [16](#)
- TdsTypeBoolean
 - BMeasureApi, [16](#)
- TdsTypeComplexDoubleFloat
 - BMeasureApi, [16](#)
- TdsTypeComplexSingleFloat
 - BMeasureApi, [16](#)
- TdsTypeDAQmxRawData
 - BMeasureApi, [16](#)
- TdsTypeDoubleFloat
 - BMeasureApi, [16](#)
- TdsTypeDoubleFloatWithUnit
 - BMeasureApi, [16](#)
- TdsTypeExtendedFloat
 - BMeasureApi, [16](#)
- TdsTypeExtendedFloatWithUnit
 - BMeasureApi, [16](#)
- TdsTypeFixedPoint
 - BMeasureApi, [16](#)
- TdsTypeI16
 - BMeasureApi, [16](#)
- TdsTypeI32
 - BMeasureApi, [16](#)
- TdsTypeI64
 - BMeasureApi, [16](#)
- TdsTypeI8
 - BMeasureApi, [16](#)
- TdsTypeSingleFloat
 - BMeasureApi, [16](#)
- TdsTypeSingleFloatWithUnit
 - BMeasureApi, [16](#)
- TdsTypeString
 - BMeasureApi, [16](#)
- TdsTypeTimeStamp
 - BMeasureApi, [16](#)
- TdsTypeU16
 - BMeasureApi, [16](#)
- TdsTypeU32
 - BMeasureApi, [16](#)
- TdsTypeU64
 - BMeasureApi, [16](#)
- TdsTypeU8
 - BMeasureApi, [16](#)
- TdsTypeVoid
 - BMeasureApi, [16](#)
- toBStringJson
 - BMeasureLib.cpp, [68](#)
 - BMeasureLib.h, [69](#)
- TocBigEndian
 - BMeasureApi, [17](#)
- TocDaqRawData
 - BMeasureApi, [17](#)
- TocInterleavedData
 - BMeasureApi, [17](#)
- TocMetaData
 - BMeasureApi, [17](#)
- TocNewObjList
 - BMeasureApi, [17](#)
- TocRawData
 - BMeasureApi, [18](#)
- toFloat
 - BMeasureApi, [18](#)
- unit
 - BMeasureApi::BMeasureUnits, [41](#)
- unitAdd
 - BMeasureApi::BMeasureUnits, [42](#)
- unitDelete
 - BMeasureApi::BMeasureUnits, [42](#)
- unitMaster
 - BMeasureApi::BMeasureUnits, [42](#)
- unitsConnect
 - BMeasureApi::BMeasureUnits, [42](#)
- unitsConnected
 - BMeasureApi::BMeasureUnits, [42](#)
- unitsConnectedNum
 - BMeasureApi::BMeasureUnits, [42](#)
- unitsDisconnect
 - BMeasureApi::BMeasureUnits, [42](#)
- unitSetEnabled
 - BMeasureApi::BMeasureUnits, [42](#)
- unitSetOrder
 - BMeasureApi::BMeasureUnits, [42](#)
- unitsFind
 - BMeasureApi::BMeasureUnits, [43](#)
- unitsNum
 - BMeasureApi::BMeasureUnits, [43](#)
- unitSort
 - BMeasureApi, [18](#)
- upload
 - Dfu, [63](#)
- upload_cmd

Dfu, [63](#)

validateFile
 Dfu, [63](#)

validateFormat
 BMeasureApi::DataFile, [59](#)

wait
 BMeasureApi::CommsNet, [49](#)
 BMeasureApi::CommsSerial, [52](#)
 BMeasureApi::CommsUsb, [56](#)

write
 BMeasureApi::CommsNet, [49](#)
 BMeasureApi::CommsSerial, [53](#)
 BMeasureApi::CommsUsb, [56](#)

writeAvailable
 BMeasureApi::CommsNet, [49](#)

writeChunks
 BMeasureApi::CommsNet, [50](#)

writeData
 BMeasureApi::DataFile, [59](#)

writeEnd
 BMeasureApi::DataFile, [59](#)

writeInfo
 BMeasureApi::DataFile, [60](#)

writeInfoBMeas
 BMeasureApi::DataFile, [60](#)

writeInfoCsv
 BMeasureApi::DataFile, [60](#)

writeInfoTdms
 BMeasureApi::DataFile, [60](#)