liburbi-cpp Reference Manual

1.0

Generated by Doxygen 1.3.7

Tue Oct 12 09:32:06 2004

Contents

Chapter 1

liburbi-cpp Hierarchical Index

1.1 liburbi-cpp Class Hierarchy

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

Chapter 2

liburbi-cpp Class Index

2.1 liburbi-cpp Class List

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

Chapter 3

liburbi-cpp File Index

3.1 liburbi-cpp File List

Here is a list of all documented files with brief descriptions:

Chapter 4

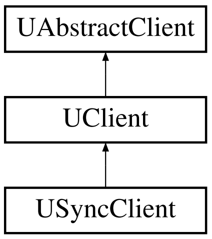
liburbi-cpp Class Documentation

4.1 UAbstractClient Class Reference

Interface for an URBI wrapper object.

`#include <uabstractclient.h>`

Inheritance diagram for UAbstractClient::



Public Member Functions

- **UAbstractClient** (const char ∗_host, int _port=**URBI_PORT**, int _buflen=**URBI_-BUFLEN**)

  Create a new instance and connect to the Urbi server.

- int **error** ()

  Return current error status, or zero if no error occurred.

- int **send** ()

  Function for backward compatibility. Will be removed in future versions.

- int **send** (const char ∗format,...)

  Send an Urbi command. The syntax is similar to the printf() function.

- int **sendBin** (const void ∗, int len)

  Send binary data.

- int **sendBin** (const void ∗, int len, const char ∗header,...)

Send an Urbi header followed by binary data.

- int **startPack** ()

  Lock the send buffer.

- int **endPack** ()

  Unlock the send buffer.

- int **pack** (const char ∗,...)

  Append Urbi commands to the send buffer.

- int **vpack** (const char ∗, va_list args)

  va_list version of pack.

- int **sendFile** (const char ∗filename)

  Send urbi commands contained in a file.

- UCallbackID **sendCommand** (**UCallback**, const char ∗,...)

  Send a command and associate its tag with a callback. ∗/.

- UCallbackID **sendCommand** (UCustomCallback, void ∗, const char ∗,...)

  Send a command and associate its tag with a callback. ∗/.

- int **sendSound** (const char ∗device, const **USound** &sound, const char ∗tag=NULL)

  Send sound data to the robot for immediate playback.

- int **putFile** (const char ∗localName, const char ∗remoteName=NULL)

  Put a file on the robot's mass storage device.

- int **putFile** (const void ∗buffer, int length, const char ∗remoteName)

  Save a buffer to a file on the robot.

- UCallbackID **setCallback** (**UCallback**, const char ∗tag)

  Associate a callback function with a tag.

- UCallbackID **setCallback** (UCustomCallback, void ∗callbackData, const char ∗tag)
- template<class C> UCallbackID **setCallback** (C &ref, **UCallbackAction**(C::∗)(const **UMessage** &), const char ∗tag)

  Callback to class member functions.

- template<class C, class P1> UCallbackID **setCallback** (C &ref, **UCallbackAction**(C::∗)(P1, const **UMessage** &), P1, const char ∗tag)
- template<class C, class P1, class P2> UCallbackID **setCallback** (C &ref, **UCallbackAction**(C::∗)(P1, P2, const **UMessage** &), P1, P2, const char ∗tag)
- template<class C, class P1, class P2, class P3> UCallbackID **setCallback** (C &ref, **UCallbackAction**(C::∗)(P1, P2, P3, const **UMessage** &), P1, P2, P3, const char ∗tag)
- template<class C, class P1, class P2, class P3, class P4> UCallbackID **setCallback** (C &ref, **UCallbackAction**(C::∗)(P1, P2, P3, P4, const **UMessage** &), P1, P2, P3, P4, const char ∗tag)
- bool **getAssociatedTag** (UCallbackID id, char ∗tag)

Get the tag associated with a registered callback.

- int **deleteCallback** (UCallbackID callBackID)

  Delete a callback.

- void **makeUniqueTag** (char ∗tag)

  Fill tag with a unique tag.

- void **notifyCallbacks** (const **UMessage** &msg)

  Simulate an Urbi message.

- virtual void **errorNotify** (const char ∗format,...)=0

  Notify of an error.

Protected Member Functions

- void **processRecvBuffer** ()

  Called each time new data is available in recvBuffer.

- virtual int **effectiveSend** (const void ∗buffer, int size)=0

  Queue data for sending, returns zero on success, nonzero on failure.

- virtual bool **canSend** (int size)=0

  Check if successive effectiveSend() of cumulated size 'size' will suceed.

- virtual void **lockList** ()=0

  Lock and unlock receive and send portions of the code.

- virtual void **lockSend** ()=0
- virtual void **unlockList** ()=0
- virtual void **unlockSend** ()=0
- UCallbackID **addCallback** (const char ∗tag, UCallbackWrapper &w)

  Add a callback to the list.

Protected Attributes

- char ∗ **host**

  Host name.

- int **port**

  URBI Port.

- int **buflen**

  URBI Buffer length.

- int **rc**

  System calls return value storage.

- char * **recvBuffer**

  Reception buffer.

- int **recvBufferPosition**

  Current position in reception buffer.

4.1.1 Detailed Description

Interface for an URBI wrapper object.

Implementations of this interface are wrappers around the URBI protocol. It handles URBI messages parsing, callback registration and various formatting functions. Implementations of this interface should:

- Redefine errorNotify() as a function able to notify the user of eventual errors.
- Redfine the four mutual exclusion functions.
- Redefine effectiveSend().
- Fill recvBuffer, update recvBufferPosition and call **processRecvBuffer()**(p. 11) when new data is available.
- Provide an execute() function in the namespace urbi, that never returns, and that will be called after initialization.

See the liburbi-cpp documentation for more informations on how to use this class.

Definition at line 227 of file uabstractclient.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 UAbstractClient::UAbstractClient (const char * _*host*, int _*port* = URBI_PORT, int _*buflen* = URBI_BUFLEN)

Create a new instance and connect to the Urbi server.

Initializes sendBuffer and recvBuffer, and copy _host and _port.

Parameters:

_***host*** IP address or name of the robot to connect to.

_***port*** TCP port to connect to.

_***buflen*** size of send and receive buffers. Implementations should establish the connection in their constructor.

Definition at line 127 of file uabstractclient.cpp.

References buflen, host, port, rc, recvBuffer, and UAbstractClient().

Referenced by UAbstractClient().

4.1.3 Member Function Documentation

4.1.3.1 int UAbstractClient::pack (const char ∗ *command*, ...)

Append Urbi commands to the send buffer.

This function must only be called between a **startPack()**(p. 11) and the corresponding **endPack()**(p. 8). Data is queued in the send buffer, and sent when endPack(à is called.

Definition at line 222 of file uabstractclient.cpp.

References pack(), rc, and vpack().

Referenced by pack().

4.1.3.2 void UAbstractClient::processRecvBuffer () [protected]

Called each time new data is available in recvBuffer.

As long as this function has not returned, neither recvBuffer nor recvBufferPos may be modified.

Definition at line 579 of file uabstractclient.cpp.

References errorNotify(), lockList(), notifyCallbacks(), recvBuffer, and recvBufferPosition.

Referenced by UClient::listenThread().

4.1.3.3 int UAbstractClient::send (const char ∗ *command*, ...)

Send an Urbi command. The syntax is similar to the printf() function.

Multiple commands can be sent in one call.

Definition at line 199 of file uabstractclient.cpp.

References effectiveSend(), rc, and vpack().

4.1.3.4 int UAbstractClient::sendSound (const char ∗ *device*, const USound & *sound*, const char ∗ *tag* = NULL)

Send sound data to the robot for immediate playback.

If tag is set, the corresponding callback will be called when sound play will be finished. The sound part of the message will be set with the content of the sound parameter. The sound data is copied in case of asynchronous send, and may be safely deleted as soon as this function returns.

Definition at line 424 of file uabstractclient.cpp.

References USound::channels, USound::data, deleteCallback(), makeUniqueTag(), USound::rate, USound::sampleFormat, USound::sampleSize, sendBin(), sendSound(), setCallback(), USound::size, and USound::soundFormat.

Referenced by sendSound().

4.1.3.5 int UAbstractClient::startPack ()

Lock the send buffer.

In threaded environnments, this function locks the send buffer so that only the calling thread can call the send functions. Otherwise do nothing.

Definition at line 180 of file uabstractclient.cpp.

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

- **uabstractclient.h**
- **uabstractclient.cpp**

4.2 UBinary Class Reference

Class containing binary data sent by the server, that could not be furtehr interpreted.

`#include <uabstractclient.h>`

Public Attributes

- void * **data**

  binary data

- char * **message**

  message as sent by the server

- int **size**

4.2.1 Detailed Description

Class containing binary data sent by the server, that could not be furtehr interpreted.

Definition at line 129 of file uabstractclient.h.

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

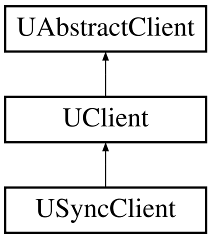
- **uabstractclient.h**

4.3 UClient Class Reference

Linux implementation of **UAbstractClient**(p. 7).

`#include <uclient.h>`

Inheritance diagram for UClient::



Public Member Functions

- **UClient** (const char ∗_host, int _port=**URBI_PORT**, int _buflen=**URBI_BUFLEN**)
- void **start** ()

  For compatibility with older versions of the library.

- void **listenThread** ()

  For internal use.

- virtual void **errorNotify** (const char ∗format,...)

  Notify of an error.

Protected Member Functions

- virtual int **effectiveSend** (const void ∗buffer, int size)

  Queue data for sending, returns zero on success, nonzero on failure.

- virtual bool **canSend** (int size)

  Check if successive effectiveSend() of cumulated size 'size' will suceed.

- virtual void **lockList** ()

  Lock and unlock receive and send portions of the code.

- virtual void **lockSend** ()
- virtual void **unlockList** ()
- virtual void **unlockSend** ()

Protected Attributes

- int **sd**

  Socket file descriptor.

4.3.1 Detailed Description

Linux implementation of **UAbstractClient**(p. 7).

This implementation creates a thread for each instance of UClient, which listens on the associated socket.

Definition at line 35 of file uclient.h.

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

- **uclient.h**
- uclient.cpp

4.4 UImage Class Reference

Class encapsulating an image.

`#include <uabstractclient.h>`

Public Attributes

- char * **data**

  pointer to image data

- int **size**

  image size in byte

- int **width**
- int **height**

  size of the image

- UImageFormat **imageFormat**

4.4.1 Detailed Description

Class encapsulating an image.

Definition at line 104 of file uabstractclient.h.

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

- **uabstractclient.h**

4.5 UMessage Class Reference

Class containing all informations related to an URBI message.

`#include <uabstractclient.h>`

Public Member Functions

- **UMessage** (**UAbstractClient** &**client**, int **timestamp**, char ∗**tag**, char ∗message, void ∗buffer=NULL, int length=0)

  This constructor steals the pointers, no copy is made.

- **UMessage** (const **UMessage** &source, bool alocate=true)

  If alocate is true, everything is copied, eles pointers are stolen.

- ∼**UMessage** ()

  Free everything if data was copied, doesn't free anything otherwise.

Public Attributes

- **UAbstractClient** & **client**

  connection from which originated the message

- int **timestamp**

  server-side timestamp

- char ∗ **tag**

  associated tag

- UMessageType **type**
- UBinaryMessageType **binaryType**

4.5.1 Detailed Description

Class containing all informations related to an URBI message.

Definition at line 138 of file uabstractclient.h.

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

- **uabstractclient.h**
- **uabstractclient.cpp**

4.6 USound Class Reference

Class encapsulating sound informations.

`#include <uabstractclient.h>`

Public Member Functions

- bool **operator==** (const **USound** &b) const

Public Attributes

- char * **data**

  pointer to sound data

- int **size**

  total size in byte

- int **channels**

  number of audio channels

- int **rate**

  rate in Hertz

- int **sampleSize**

  sample size in bit

- USoundFormat **soundFormat**

  format of the sound data

- USoundSampleFormat **sampleFormat**

  sample format

4.6.1 Detailed Description

Class encapsulating sound informations.

Definition at line 113 of file uabstractclient.h.

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

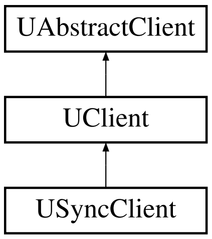
- **uabstractclient.h**

4.7 USyncClient Class Reference

UClient(p. 14) linux implementation with support for synchronous functions.

`#include <usyncclient.h>`

Inheritance diagram for USyncClient::



Public Member Functions

- **USyncClient** (const char *_host, int _port=**URBI_PORT**, int _buflen=**URBI_-BUFLEN**)
- int **syncSend** (const void *buffer, int length)

  Send given buffer without copying it.
- int **syncGetImage** (const char *cameraDevice, void *buffer, int &buffersize, int format, int transmitFormat, int &width, int &height)

  Get an image in a synchronous way.
- int **syncGetDevice** (const char *device, double &val)

  Get the value of a device in a synchronous way.
- int **syncGetNormalizedDevice** (const char *device, double &val)

  Get the normalized value of a device in a synchronous way.
- int **syncGetDevice** (const char *device, const char *field, double &val)

  Get a field of a device in a synchronous way.
- int **syncGetSound** (const char *device, int duration, **USound** &sound)

  Get sound for duration milliseconds in buffer.

4.7.1 Detailed Description

UClient(p. 14) linux implementation with support for synchronous functions.

These functions differs from the **UClient**(p. 14) interface in that they are synchronous. One must seriously ponder the fact that they are not easily portable before using them.

Definition at line 38 of file usyncclient.h.

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

- **usyncclient.h**
- usyncclient.cpp

Chapter 5

liburbi-cpp File Documentation

5.1 uabstractclient.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <sys/stat.h>
#include <errno.h>
#include <math.h>
#include <setjmp.h>
#include <algorithm>
#include "../../lib/jpeg-6b/jpeglib.h"
#include "../../lib/jpeg-6b/jerror.h"
#include "uabstractclient.h"
```

Defines

- #define **DEBUG** 0

Enumerations

- enum **UCallbackType** { **UCB_**, **UCB_C** }

Functions

- int **min** (int a, int b)
- void * **read_jpeg** (const char *jpgbuffer, int jpgbuffer_size, bool RGB, int &output_size)
- **UCallbackAction sendSound_** (void *cb, const **UMessage** &msg)
- unsigned char **clamp** (float v)
- int **convertYCrCbtoRGB** (const byte *sourceImage, int bufferSize, byte *destination-Image)

- int **convertJPEGtoYCrCb** (const byte ∗source, int sourcelen, byte ∗dest, int &size)
- int **convertJPEGtoRGB** (const byte ∗source, int sourcelen, byte ∗dest, int &size)
- void **init_source** (j_decompress_ptr cinfo)
- boolean **fill_input_buffer** (j_decompress_ptr cinfo)
- void **term_source** (j_decompress_ptr cinfo)
- void **skip_input_data** (j_decompress_ptr cinfo, long num_bytes)
- **urbi_jpeg_error_exit** (j_common_ptr cinfo)
- template<class S, class D> void **copy** (S ∗src, D ∗dst, int sc, int dc, int sr, int dr, int count, bool sf, bool df)
- int **convert** (const **USound** &source, **USound** &dest)

  Conversion between various sound formats.

Variables

- UCallbackID **nextId**

5.1.1 Detailed Description

Id

uabstractclient.cpp(p. 21),v 1.3 2004/09/24 08:48:44 nottale Exp

Definition of the URBI interface class

Definition in file **uabstractclient.cpp**.

5.1.2 Function Documentation

5.1.2.1 int convert (const USound & *source*, USound & *dest*)

Conversion between various sound formats.

If any of destination,'s channel, sampleSize, rate or sampleFormat parameter is 0, values from source will be used.

Definition at line 969 of file uabstractclient.cpp.

References USound::channels, USound::data, USound::rate, USound::sampleFormat, USound::sampleSize, USound::size, and USound::soundFormat.

Referenced by USyncClient::syncGetSound().

5.1.2.2 void ∗ read_jpeg (const char ∗ *jpgbuffer*, int *jpgbuffer_size*, bool *RGB*, int & *output_size*) `[static]`

Convert a jpeg image to YCrCb or RGB. Allocate the buffer with malloc.

Definition at line 862 of file uabstractclient.cpp.

5.2 uabstractclient.h File Reference

```
#include <stdio.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdarg.h>
#include <list>
```

Namespaces

- namespace **std**

Classes

- class **UImage**
  Class encapsulating an image.
- class **USound**
  Class encapsulating sound informations.
- class **UBinary**
  Class containing binary data sent by the server, that could not be furtehr interpreted.
- class **UMessage**
  Class containing all informations related to an URBI message.
- class **UAbstractClient**
  Interface for an URBI wrapper object.

Defines

- #define **UINVALIDCALLBACKID** 0

Typedefs

- typedef unsigned char **byte**
- typedef unsigned int **UCallbackID**
- typedef **UCallbackAction**(∗ **UCallback**)(const **UMessage** &msg)

Callback prototypes.

- typedef **UCallbackAction**(∗ **UCustomCallback**)(void ∗callbackData, const **UMessage** &msg)

Enumerations

- enum **UCallbackAction** { **URBI_CONTINUE** = 0, **URBI_REMOVE** }

  Return values for the callack functions.

- enum **UMessageType** {

  MESSAGE_DOUBLE, **MESSAGE_STRING**, **MESSAGE_BINARY**, **MESSAGE_SYSTEM**,

  MESSAGE_UNKNOWN }
- enum **UBinaryMessageType** { **BINARYMESSAGE_NONE**, **BINARYMESSAGE_UNKNOWN**, **BINARYMESSAGE_IMAGE**, **BINARYMESSAGE_SOUND** }
- enum **UImageFormat** { **IMAGE_RGB** = 1, **IMAGE_YCbCr** = 2, **IMAGE_JPEG** = 3, **IMAGE_PPM** = 4 }
- enum **USoundFormat** { **SOUND_RAW**, **SOUND_WAV**, **SOUND_MP3**, **SOUND_OGG** }
- enum **USoundSampleFormat** { **SAMPLE_SIGNED** = 1, **SAMPLE_UNSIGNED** = 2 }

Functions

- int **convertRGBtoYCrCb** (const byte ∗source, int sourcelen, byte ∗dest)

  Image format conversion functions.

- int **convertYCrCbtoRGB** (const byte ∗source, int sourcelen, byte ∗dest)
- int **convertJPEGtoYCrCb** (const byte ∗source, int sourcelen, byte ∗dest, int &size)
- int **convertJPEGtoRGB** (const byte ∗source, int sourcelen, byte ∗dest, int &size)
- int **convert** (const **USound** &source, **USound** &destination)

  Conversion between various sound formats.

Variables

- const int **URBI_BUFLEN** = 128000

  Connection Buffer size.

- const int **URBI_PORT** = 54000

  Standard port of URBI server.

- const int **URBI_MAX_TAG_LENGTH** = 64

  Maximum length of an URBI tag.

5.2.1 Detailed Description

Id

uabstractclient.h(p. 24),v 1.1 2004/07/08 14:24:12 nottale Exp

Definition of the URBI interface class

Definition in file **uabstractclient.h**.

5.2.2 Enumeration Type Documentation

5.2.2.1 enum UCallbackAction

Return values for the callack functions.

Each callback function, when called, must return with either URBI_CONTINUE or URBI_-REMOVE:

- URBI_CONTINUE means that the client should continue to call this callbak function.
- URBI_REMOVE means that the client should never call this callback again.

Definition at line 54 of file uabstractclient.h.

Referenced by UAbstractClient::notifyCallbacks().

5.2.3 Function Documentation

5.2.3.1 int convert (const USound & *source*, USound & *dest*)

Conversion between various sound formats.

If any of destination,'s channel, sampleSize, rate or sampleFormat parameter is 0, values from source will be used.

Definition at line 969 of file uabstractclient.cpp.

References USound::channels, USound::data, USound::rate, USound::sampleFormat, USound::sampleSize, USound::size, and USound::soundFormat.

Referenced by USyncClient::syncGetSound().

5.3 uclient.h File Reference

```
#include "uabstractclient.h"
```

Namespaces

- namespace **urbi**

Classes

- class **UClient**

  Linux implementation of **UAbstractClient**(p. 7).

5.3.1 Detailed Description

Id

uclient.h(p. 27),v 1.1 2004/07/08 14:24:12 nottale Exp

Definition of the URBI interface class

Definition in file **uclient.h**.

5.4 usyncclient.h File Reference

```
#include "uclient.h"
```

Classes

- class **USyncClient**

  UClient(p. 14) *linux implementation with support for synchronous functions.*

Enumerations

- enum **UTransmitFormat** { **URBI_TRANSMIT_JPEG**, **URBI_TRANSMIT_-YCbCr** }

5.4.1 Detailed Description

Id

usyncclient.h(p. 28),v 1.1 2004/07/08 14:24:12 nottale Exp

Definition of the URBI interface class

Definition in file **usyncclient.h**.

Index