Mamba RealTime Api Reference Manual

Automatically generated using doxygen

February 12, 2011

Contents

1 Data Structures **5**

2 File List **5**

3 MB_Image Struct Reference **5**
- 3.1 Detailed Description . . . 5
- 3.2 Field Documentation . . . 6
  - 3.2.1 depth . . . 6
  - 3.2.2 height . . . 6
  - 3.2.3 PIXARRAY . . . 6
  - 3.2.4 PLINES . . . 6
  - 3.2.5 width . . . 6

4 MBRT_avcvidT Struct Reference **6**

5 MBRT_Context Struct Reference **6**

6 MBRT_dshowvidT Struct Reference **7**

7 MBRT_vidUnion Union Reference **7**

8 mambaCommon.h File Reference **7**
- 8.1 Detailed Description . . . 8
- 8.2 Define Documentation . . . 8
  - 8.2.1 BIN_FILL_VALUE . . . 8
  - 8.2.2 GREY_FILL_VALUE . . . 8
  - 8.2.3 I32_FILL_VALUE . . . 8
  - 8.2.4 MB_LINE_COUNT . . . 9
  - 8.2.5 MB_LINE_OFFSET . . . 9
  - 8.2.6 MB_X_LEFT . . . 9
  - 8.2.7 MB_X_RIGHT . . . 9
  - 8.2.8 MB_Y_BOTTOM . . . 9
  - 8.2.9 MB_Y_TOP . . . 9
- 8.3 Typedef Documentation . . . 9
  - 8.3.1 PIX32 . . . 9
  - 8.3.2 PIX8 . . . 9
  - 8.3.3 PLINE . . . 9
  - 8.3.4 PLINE32 . . . 9
  - 8.3.5 Sint16 . . . 9
  - 8.3.6 Sint32 . . . 9
  - 8.3.7 Sint64 . . . 9
  - 8.3.8 Sint8 . . . 9
  - 8.3.9 Uint16 . . . 9
  - 8.3.10 Uint32 . . . 10
  - 8.3.11 Uint64 . . . 10
  - 8.3.12 Uint8 . . . 10
- 8.4 Enumeration Type Documentation . . . 10
  - 8.4.1 MB_edgemode_t . . . 10
  - 8.4.2 MB_grid_t . . . 10

9 mambaRTApi.h File Reference **10**
- 9.1 Detailed Description . . . 11
- 9.2 Enumeration Type Documentation . . . 11
  - 9.2.1 MBRT_eventcode . . . 11
  - 9.2.2 MBRT_vidType . . . 11
- 9.3 Function Documentation . . . 11
  - 9.3.1 MBRT_CreateContext . . . 11
  - 9.3.2 MBRT_CreateDisplay . . . 11

9.3.3 MBRT_CreateVideoAcq . . . 11
9.3.4 MBRT_DestroyContext . . . 11
9.3.5 MBRT_DestroyDisplay . . . 12
9.3.6 MBRT_DestroyVideoAcq . . . 12
9.3.7 MBRT_GetAcqFrameRate . . . 12
9.3.8 MBRT_GetAcqSize . . . 12
9.3.9 MBRT_GetImageFromAcq . . . 12
9.3.10 MBRT_IconDisplay . . . 12
9.3.11 MBRT_PaletteDisplay . . . 13
9.3.12 MBRT_PollDisplay . . . 13
9.3.13 MBRT_RecordEnd . . . 13
9.3.14 MBRT_RecordImage . . . 13
9.3.15 MBRT_RecordStart . . . 13
9.3.16 MBRT_StartAcq . . . 13
9.3.17 MBRT_StopAcq . . . 13
9.3.18 MBRT_UpdateDisplay . . . 14

10 mambaRTApi_loc.h File Reference 14
10.1 Detailed Description . . . 15
10.2 Function Documentation . . . 15
10.2.1 MBRT_CreateVideoAcq_avc . . . 15
10.2.2 MBRT_CreateVideoAcq_dshow . . . 15
10.2.3 MBRT_DestroyVideoAcq_avc . . . 15
10.2.4 MBRT_DestroyVideoAcq_dshow . . . 15
10.2.5 MBRT_GetAcqFrameRate_avc . . . 16
10.2.6 MBRT_GetAcqFrameRate_dshow . . . 16
10.2.7 MBRT_GetAcqSize_avc . . . 16
10.2.8 MBRT_GetAcqSize_dshow . . . 16
10.2.9 MBRT_GetImageFromAcq_avc . . . 16
10.2.10 MBRT_GetImageFromAcq_dshow . . . 16
10.2.11 MBRT_StartAcq_avc . . . 17
10.2.12 MBRT_StartAcq_dshow . . . 17
10.2.13 MBRT_StopAcq_avc . . . 17
10.2.14 MBRT_StopAcq_dshow . . . 17
10.3 Variable Documentation . . . 17
10.3.1 context . . . 17

11 MBRT_Display.cpp File Reference 17
11.1 Detailed Description . . . 17
11.2 Function Documentation . . . 18
11.2.1 MBRT_CreateDisplay . . . 18
11.2.2 MBRT_DestroyDisplay . . . 18
11.2.3 MBRT_IconDisplay . . . 18
11.2.4 MBRT_PaletteDisplay . . . 18
11.2.5 MBRT_PollDisplay . . . 18
11.2.6 MBRT_UpdateDisplay . . . 18

12 MBRT_error.cpp File Reference 19
12.1 Detailed Description . . . 19
12.2 Function Documentation . . . 19
12.2.1 MBRT_StrErr . . . 19
12.3 Variable Documentation . . . 19
12.3.1 err_str . . . 19

13 MBRT_error.h File Reference **19**
13.1 Detailed Description . . . 20
13.2 Enumeration Type Documentation . . . 20
13.2.1 MBRT_errcode . . . 20
13.3 Function Documentation . . . 21
13.3.1 MBRT_StrErr . . . 21

14 MBRT_Record.cpp File Reference **21**
14.1 Detailed Description . . . 21
14.2 Function Documentation . . . 21
14.2.1 MBRT_RecordEnd . . . 21
14.2.2 MBRT_RecordImage . . . 21
14.2.3 MBRT_RecordStart . . . 22

15 MBRT_VideoAcq.cpp File Reference **22**
15.1 Detailed Description . . . 22
15.2 Function Documentation . . . 22
15.2.1 MBRT_CreateVideoAcq . . . 22
15.2.2 MBRT_DestroyVideoAcq . . . 22
15.2.3 MBRT_GetAcqFrameRate . . . 22
15.2.4 MBRT_GetAcqSize . . . 23
15.2.5 MBRT_GetImageFromAcq . . . 23
15.2.6 MBRT_StartAcq . . . 23
15.2.7 MBRT_StopAcq . . . 23

16 MBRT_VideoAcq_avc.cpp File Reference **23**
16.1 Detailed Description . . . 23
16.2 Function Documentation . . . 24
16.2.1 MBRT_CreateVideoAcq_avc . . . 24
16.2.2 MBRT_DestroyVideoAcq_avc . . . 24
16.2.3 MBRT_GetAcqFrameRate_avc . . . 24
16.2.4 MBRT_GetAcqSize_avc . . . 24
16.2.5 MBRT_GetImageFromAcq_avc . . . 24
16.2.6 MBRT_StartAcq_avc . . . 24
16.2.7 MBRT_StopAcq_avc . . . 25

17 MBRT_VideoAcq_dshow.cpp File Reference **25**
17.1 Detailed Description . . . 25
17.2 Define Documentation . . . 25
17.2.1 RETURN_ON_ERR . . . 25
17.3 Function Documentation . . . 25
17.3.1 MBRT_CreateVideoAcq_dshow . . . 25
17.3.2 MBRT_DestroyVideoAcq_dshow . . . 25
17.3.3 MBRT_GetAcqFrameRate_dshow . . . 26
17.3.4 MBRT_GetAcqSize_dshow . . . 26
17.3.5 MBRT_GetImageFromAcq_dshow . . . 26
17.3.6 MBRT_StartAcq_dshow . . . 26
17.3.7 MBRT_StopAcq_dshow . . . 26

List of Figures

1 Image structure and variables 27

Data Structure Index

1 Data Structures

Here are the data structures with brief descriptions:

MB_Image 5

MBRT_avcvidT 6

MBRT_Context 6

MBRT_dshowvidT 7

MBRT_vidUnion 7

File Index

2 File List

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

mambaCommon.h 7

mambaRTApi.h 10

mambaRTApi_loc.h 14

MBRT_Display.cpp 17

MBRT_error.cpp 19

MBRT_error.h 19

MBRT_Record.cpp 21

MBRT_VideoAcq.cpp 22

MBRT_VideoAcq_avc.cpp 23

MBRT_VideoAcq_dshow.cpp 25

Data Structure Documentation

3 MB_Image Struct Reference

```
#include <mambaCommon.h>
```

Data Fields

- **Uint32 width**
- **Uint32 height**
- **Uint32 depth**
- **PLINE * PLINES**
- **PIX8 * PIXARRAY**

3.1 Detailed Description

Images structure with the width, height and depth; the pixels array (PIXARRAY) and entry point array to each line of the image (PLINES)

3.2 Field Documentation

3.2.1 Uint32 MB_Image::depth

The depth of the image

3.2.2 Uint32 MB_Image::height

The height of the image

3.2.3 PIX8* MB_Image::PIXARRAY

pixel array

3.2.4 PLINE* MB_Image::PLINES

accesors to pixel lines

3.2.5 Uint32 MB_Image::width

The width of the image

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

- **mambaCommon.h**

4 MBRT_avcvidT Struct Reference

Data Fields

- AVFormatContext * **format_ctx**
- AVCodecContext * **codec_ctx**
- int **videoStream**
- AVFrame * **frame**
- AVPacket **packet**

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

- **mambaRTApi_loc.h**

5 MBRT_Context Struct Reference

Data Fields

- **MBRT_vidType type**
- int **fd**
- **MBRT_vidUnion video**
- SDL_Surface * **screen**
- **Uint32 sz_x**
- **Uint32 sz_y**
- SDL_Color **color_palette** [256]
- SDL_Color **standard_palette** [256]
- **Uint32 isPalettized**
- **Uint32 isFpsDisplayed**
- **Uint32 old_call** [FPS_MEAN_SIZE]
- int **index_fps**
- **Uint32 histo** [256]
- **Uint32 isHistoDisplayed**
- **Uint8 icon** [256]
- **Sint32 isFullscreen**

- **Uint32 isRecording**
- AVFormatContext * **rec_fmt_ctx**
- AVFrame * **pictureRGB**
- AVFrame * **picture**
- struct SwsContext * **img_convert_ctx**
- uint8_t * **video_outbuf**

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

- **mambaRTApi_loc.h**

6 MBRT_dshowvidT Struct Reference

Data Fields

- IGraphBuilder * **pGraph**
- ICaptureGraphBuilder2 * **pBuild**
- IBaseFilter * **pCap**
- ISampleGrabber * **pGrab**
- IMediaControl * **pCtrl**
- int **devnum**
- int **w**
- int **h**
- int **size**
- void * **buffer**

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

- **mambaRTApi_loc.h**

7 MBRT_vidUnion Union Reference

Data Fields

- **MBRT_dshowvidT dshow**
- **MBRT_avcvidT avc**

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

- **mambaRTApi_loc.h**

File Documentation

8 mambaCommon.h File Reference

```
#include <stdint.h>
```

Data Structures

- struct **MB_Image**

Defines

- #define **MB_X_LEFT**(im) X_LEFT
- #define **MB_X_RIGHT**(im) X_RIGHT
- #define **MB_Y_TOP**(im) Y_TOP
- #define **MB_Y_BOTTOM**(im) Y_BOTTOM
- #define **MB_LINE_COUNT**(im) ((im->width*im->depth)/CHARBIT)
- #define **MB_LINE_OFFSET**(im) (MB_X_LEFT(im))
- #define **BIN_FILL_VALUE**(edge) ((edge==MB_FILLED_EDGE) ? UINT32_MAX:0)
- #define **GREY_FILL_VALUE**(edge) ((edge==MB_FILLED_EDGE) ? UINT32_MAX:0)
- #define **I32_FILL_VALUE**(edge) ((edge==MB_FILLED_EDGE) ? UINT32_MAX:0)

Typedefs

- typedef uint8_t **Uint8**
- typedef uint16_t **Uint16**
- typedef uint32_t **Uint32**
- typedef uint64_t **Uint64**
- typedef int8_t **Sint8**
- typedef int16_t **Sint16**
- typedef int32_t **Sint32**
- typedef int64_t **Sint64**
- typedef uint8_t **PIX8**
- typedef **PIX8** ∗ **PLINE**
- typedef uint32_t **PIX32**
- typedef **PIX32** ∗ **PLINE32**

Enumerations

- enum **MB_grid_t** { **MB_HEXAGONAL_GRID** = 1, **MB_SQUARE_GRID** = 0 }
- enum **MB_edgemode_t** { **MB_EMPTY_EDGE** = 0, **MB_FILLED_EDGE** = 1 }

8.1 Detailed Description

Date:

31-03-2009

This file contains the various definitions, macro, struct that are commons between the various modules of the library

The copyright license of Mamba is reminded here :

Copyright (c) <2009>, <Nicolas BEUCHER and ARMINES for the Centre de Morphologie Mathématique(CMM), common research center to ARMINES and MINES Paristech>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Except as contained in this notice, the names of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without their prior written authorization.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

8.2 Define Documentation

8.2.1 #define BIN_FILL_VALUE(*edge*) ((edge==MB_FILLED_EDGE) ? UINT32_MAX:0)

How to fill the edge (binary bits images)

8.2.2 #define GREY_FILL_VALUE(*edge*) ((edge==MB_FILLED_EDGE) ? UINT32_MAX:0)

How to fill the edge (8 bits images)

8.2.3 #define I32_FILL_VALUE(*edge*) ((edge==MB_FILLED_EDGE) ? UINT32_MAX:0)

How to fill the edge (32 bits images)

8.2.4 #define MB_LINE_COUNT(*im*) ((im->width*im->depth)/CHARBIT)

Returns the size in bytes of an image line

8.2.5 #define MB_LINE_OFFSET(*im*) (MB_X_LEFT(im))

Returns the size in bytes of the line offset

8.2.6 #define MB_X_LEFT(*im*) X_LEFT

Getting image frame offset from left

8.2.7 #define MB_X_RIGHT(*im*) X_RIGHT

Getting image frame offset from right

8.2.8 #define MB_Y_BOTTOM(*im*) Y_BOTTOM

Getting image frame offset from bottom

8.2.9 #define MB_Y_TOP(*im*) Y_TOP

Getting image frame offset from top

8.3 Typedef Documentation

8.3.1 typedef uint32_t PIX32

Signed 32-bit pixels value type

8.3.2 typedef uint8_t PIX8

grey-scale pixels value type

8.3.3 typedef PIX8* PLINE

Pixels line pointers type

8.3.4 typedef PIX32* PLINE32

32-bit pixels line pointers type

8.3.5 typedef int16_t Sint16

Signed 16 bit value type

8.3.6 typedef int32_t Sint32

Signed 32 bit value type

8.3.7 typedef int64_t Sint64

Signed 64 bit value type

8.3.8 typedef int8_t Sint8

Signed 8 bit value type

8.3.9 typedef uint16_t Uint16

Unsigned 16 bit value type

8.3.10 typedef uint32_t Uint32

Unsigned 32 bit value type

8.3.11 typedef uint64_t Uint64

Unsigned 64 bit value type

8.3.12 typedef uint8_t Uint8

Unsigned 8 bit value type

8.4 Enumeration Type Documentation

8.4.1 enum MB_edgemode_t

enumerate for edge mode : either empty or filled

8.4.2 enum MB_grid_t

enumerate for grid : either square or hexagonal the value is supposed to give the number of directions allowed

9 mambaRTApi.h File Reference

```
#include "mambaCommon.h"
   #include "MBRT_error.h"
   #include <dshow.h>
```

Enumerations

- enum **MBRT_eventcode** { **NO_EVENT**, **EVENT_CLOSE**, **EVENT_PROCESS**, **EVENT_PAUSE** }
- enum **MBRT_vidType** { **DSHOW_TYPE**, **AVC_TYPE**, **NONE_TYPE** }

Functions

- **MBRT_errcode MBRT_CreateContext** (void)
- **MBRT_errcode MBRT_DestroyContext** (void)
- **MBRT_errcode MBRT_CreateVideoAcq** (char *device, **MBRT_vidType** type)
- **MBRT_errcode MBRT_DestroyVideoAcq** (void)
- **MBRT_errcode MBRT_GetAcqSize** (int *acq_w, int *acq_h)
- **MBRT_errcode MBRT_GetAcqFrameRate** (double *ofps)
- **MBRT_errcode MBRT_GetImageFromAcq** (**MB_Image** *dest)
- **MBRT_errcode MBRT_StartAcq** ()
- **MBRT_errcode MBRT_StopAcq** ()
- **MBRT_errcode MBRT_RecordStart** (char *filename)
- **MBRT_errcode MBRT_RecordEnd** (void)
- **MBRT_errcode MBRT_RecordImage** (**MB_Image** *src)
- **MBRT_errcode MBRT_CreateDisplay** (int w, int h)
- **MBRT_errcode MBRT_DestroyDisplay** (void)
- **MBRT_errcode MBRT_UpdateDisplay** (**MB_Image** *src, double wfps, double *ofps)
- **MBRT_errcode MBRT_PaletteDisplay** (**Uint8** *palette)
- **MBRT_errcode MBRT_IconDisplay** (**Uint8** *icon)
- **MBRT_errcode MBRT_PollDisplay** (**MBRT_eventcode** *event_code)

9.1 Detailed Description

Date:

03-27-2009

This file contains the various definitions, global variables macro, struct and functions created for the library.

9.2 Enumeration Type Documentation

9.2.1 enum MBRT_eventcode

Type definition for display event code

Enumerator:

NO_EVENT No events

EVENT_CLOSE Close event occurs when the user closes the window or press esc inside it

EVENT_PROCESS Event to toggle on/off the process

EVENT_PAUSE Event to activate or deactive the pause

9.2.2 enum MBRT_vidType

Type definition for the acquisition device

Enumerator:

DSHOW_TYPE DirectShow API type device

AVC_TYPE Audio video codec API type device (video file)

NONE_TYPE No device

9.3 Function Documentation

9.3.1 MBRT_errcode MBRT_CreateContext (void)

Creates the context used throughout the library. The context describes the data displayed, the device used for acquisition, and contains general information needed by all the functions

9.3.2 MBRT_errcode MBRT_CreateDisplay (int *w*, int *h*)

Initializes SDL and creates the video display (SDL screen)

Parameters:

w width of the display (resolution)

h height of the display (resolution)

Returns:

An error code (NO_ERR if successful)

9.3.3 MBRT_errcode MBRT_CreateVideoAcq (char * *device*, MBRT_vidType *type*)

Fillss the video acquisition structure with the parameters of the given device and initializes it.

Parameters:

device the video device (usually /dev/video0) or the video file.

types the type of device (e.g video4linux, video4linux2, avc ...)

Returns:

an error code (NO_ERR if successful)

9.3.4 MBRT_errcode MBRT_DestroyContext (void)

Destroys the context and all the data associated to it

9.3.5 MBRT_errcode MBRT_DestroyDisplay (void)

Destroys the video display (SDL screen) and quits SDL

Returns:

An error code (NO_ERR if successful)

9.3.6 MBRT_errcode MBRT_DestroyVideoAcq (void)

Closes the acquisition device and resets the structure

Returns:

NO_ERR if successful

9.3.7 MBRT_errcode MBRT_GetAcqFrameRate (double * *ofps*)

Returns the acquisition device default framerate.

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

9.3.8 MBRT_errcode MBRT_GetAcqSize (int * *acq_w*, int * *acq_h*)

Returns the acquisition device resolution.

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

9.3.9 MBRT_errcode MBRT_GetImageFromAcq (MB_Image * *dest*)

Obtains an image form the acquisition device

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

9.3.10 MBRT_errcode MBRT_IconDisplay (Uint8 * *icon*)

Changes the small (16x16) icon (black and white) in the upper left corner of the display. This allows to inform the user of some events. The icon is an array of 256 integers. A 0 value will means the pixel is not drawn, non 0 are drawn using the OSD color.

Parameters:

icon the 16x16 icon pixels array

Returns:

An error code (NO_ERR if successful)

9.3.11 MBRT_errcode MBRT_PaletteDisplay (Uint8 * *palette*)

Changes the palette associated with the display

Parameters:

palette an array containing the complete palette definition (256*3 integers)

Returns:

An error code (NO_ERR if successful)

9.3.12 MBRT_errcode MBRT_PollDisplay (MBRT_eventcode * *event_code*)

Handles event that have occurred in the display

Parameters:

event_code an integer representing a specific event (output)

Returns:

An error code (NO_ERR if successful)

9.3.13 MBRT_errcode MBRT_RecordEnd (void)

Ends the recording. Closes the codec and terminates the file.

Returns:

An error code (NO_ERR if successful)

9.3.14 MBRT_errcode MBRT_RecordImage (MB_Image * *src*)

Records an image

Parameters:

src the mamba image to record

Returns:

An error code (NO_ERR if successful)

9.3.15 MBRT_errcode MBRT_RecordStart (char * *filename*)

Starts the recording. This function activates the libavformat and libavcodec libraries to encode the image using MPEG2 codec (DVD format).

Parameters:

filename the path to the created video file

Returns:

An error code (NO_ERR if successful)

9.3.16 MBRT_errcode MBRT_StartAcq ( )

Starts the acquisition device capture process

Returns:

NO_ERR if successful

9.3.17 MBRT_errcode MBRT_StopAcq ( )

Stops the acquisition device capture process

Returns:

NO_ERR if successful

9.3.18 MBRT_errcode MBRT_UpdateDisplay (MB_Image * *src*, double *wfps*, double * *ofps*)

Updates the display with the content of a given mamba image structure

Parameters:

src the image displayed

wfps input the desired framerate

ofps output the framerate

Returns:

An error code (NO_ERR if successful)

10 mambaRTApi_loc.h File Reference

```
#include <qedit.h>
   #include "mambaRTApi.h"
   #include <SDL.h>
   #include <libavcodec/avcodec.h>
   #include <libavformat/avformat.h>
   #include <libswscale/swscale.h>
```

Data Structures

- struct **MBRT_dshowvidT**
- struct **MBRT_avcvidT**
- union **MBRT_vidUnion**
- struct **MBRT_Context**

Defines

- #define **__IDxtCompositor_INTERFACE_DEFINED__**
- #define **__IDxtAlphaSetter_INTERFACE_DEFINED__**
- #define **__IDxtJpeg_INTERFACE_DEFINED__**
- #define **__IDxtKey_INTERFACE_DEFINED__**
- #define **MBRT_TITLE** "Mamba RealTime"
- #define **FRAME_COLOR** 0xa00000
- #define **FPS_VALUE_COLOR** 0xa00000
- #define **FPS_THICKNESS** 3
- #define **FPS_MEAN_SIZE** 20
- #define **HISTO_BLACKENING** 60
- #define **HISTO_COLOR** 0xffffff
- #define **REC_SIZE** 16
- #define **REC_COLOR** 0xb00000

Functions

- **MBRT_errcode MBRT_CreateVideoAcq_dshow** (int device)
- **MBRT_errcode MBRT_DestroyVideoAcq_dshow** (void)
- **MBRT_errcode MBRT_GetAcqSize_dshow** (int *acq_w, int *acq_h)
- **MBRT_errcode MBRT_GetAcqFrameRate_dshow** (double *fps)
- **MBRT_errcode MBRT_GetImageFromAcq_dshow** (**MB_Image** *dest)
- **MBRT_errcode MBRT_StopAcq_dshow** ()
- **MBRT_errcode MBRT_StartAcq_dshow** ()
- **MBRT_errcode MBRT_CreateVideoAcq_avc** (char *video_path)
- **MBRT_errcode MBRT_DestroyVideoAcq_avc** (void)

- **MBRT_errcode MBRT_GetAcqSize_avc** (int *acq_w, int *acq_h)
- **MBRT_errcode MBRT_GetAcqFrameRate_avc** (double *fps)
- **MBRT_errcode MBRT_GetImageFromAcq_avc** (**MB_Image** *dest)
- **MBRT_errcode MBRT_StopAcq_avc** ()
- **MBRT_errcode MBRT_StartAcq_avc** ()

Variables

- **MBRT_Context * context**

10.1 Detailed Description

Date:

03-27-2009

This file contains the various definitions, global variables macro, struct and functions that are shared between components of the library but are not meant to be exported to the outside world.

10.2 Function Documentation

10.2.1 MBRT_errcode MBRT_CreateVideoAcq_avc (char * *video_path*)

Opens a video file as a video acquisition device using the advanced video codec (AVC) library.

Parameters:

video_path path to the video file (supported codec depends on your local implementation of the libavcodec)

Returns:

an error code (NO_ERR if successful)

10.2.2 MBRT_errcode MBRT_CreateVideoAcq_dshow (int *device*)

Fills the video acquisition structure with the parameters of the given device and initializes it (DSHOW).

Parameters:

device the video device (usually 0)

Returns:

an error code (NO_ERR if successful)

10.2.3 MBRT_errcode MBRT_DestroyVideoAcq_avc (void)

Closes the video file playback (AVC) and reset the structure

Returns:

NO_ERR if successful

10.2.4 MBRT_errcode MBRT_DestroyVideoAcq_dshow (void)

Closes the acquisition device and reset the structure (DSHOW).

Returns:

NO_ERR if successful

10.2.5 MBRT_errcode MBRT_GetAcqFrameRate_avc (double * *ofps*)

Returns the acquisition device default framerate (AVC).

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

10.2.6 MBRT_errcode MBRT_GetAcqFrameRate_dshow (double * *ofps*)

Returns the acquisition device default framerate (DSHOW).

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

10.2.7 MBRT_errcode MBRT_GetAcqSize_avc (int * *acq_w*, int * *acq_h*)

Returns the video played size (AVC).

Parameters:

acq_w the width (output)
acq_h the height (output)

Returns:

NO_ERR if successful

10.2.8 MBRT_errcode MBRT_GetAcqSize_dshow (int * *acq_w*, int * *acq_h*)

Returns the acquisition device resolution (DSHOW).

Parameters:

acq_w the width (output)
acq_h the height (output)

Returns:

NO_ERR if successful

10.2.9 MBRT_errcode MBRT_GetImageFromAcq_avc (MB_Image * *dest*)

Obtains an image from the video (AVC)

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

10.2.10 MBRT_errcode MBRT_GetImageFromAcq_dshow (MB_Image * *dest*)

Obtains an image from the acquisition device (DSHOW).

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

10.2.11 MBRT_errcode MBRT_StartAcq_avc ( )

Starts the acquisition device capture process (AVC)

Returns:

NO_ERR if successful

10.2.12 MBRT_errcode MBRT_StartAcq_dshow ( )

Starts the acquisition device capture process (DSHOW).

Returns:

NO_ERR if successful

10.2.13 MBRT_errcode MBRT_StopAcq_avc ( )

Stops the acquisition device capture process (AVC)

Returns:

NO_ERR if successful

10.2.14 MBRT_errcode MBRT_StopAcq_dshow ( )

Stops the acquisition device capture process (DSHOW).

Returns:

NO_ERR if successful

10.3 Variable Documentation

10.3.1 MBRT_Context* context

Structure holding the complete context information (display, device, ...)

11 MBRT_Display.cpp File Reference

```
#include "mambaRTApi_loc.h"
```

Defines

- #define **RED**(pix) (*(pix+2))
- #define **GREEN**(pix) (*(pix+1))
- #define **BLUE**(pix) (*pix)

Functions

- **MBRT_errcode MBRT_CreateDisplay** (int w, int h)
- **MBRT_errcode MBRT_DestroyDisplay** ()
- **MBRT_errcode MBRT_UpdateDisplay** (**MB_Image** *src, double wfps, double *ofps)
- **MBRT_errcode MBRT_IconDisplay** (**Uint8** *icon)
- **MBRT_errcode MBRT_PaletteDisplay** (**Uint8** *palette)
- **MBRT_errcode MBRT_PollDisplay** (**MBRT_eventcode** *event_code)

11.1 Detailed Description

Author:

Nicolas Beucher

Date:

03-29-2009

11.2 Function Documentation

11.2.1 MBRT_errcode MBRT_CreateDisplay (int *w*, int *h*)

Initializes SDL and creates the video display (SDL screen)

Parameters:

w width of the display (resolution)

h height of the display (resolution)

Returns:

An error code (NO_ERR if successful)

11.2.2 MBRT_errcode MBRT_DestroyDisplay (void)

Destroys the video display (SDL screen) and quits SDL

Returns:

An error code (NO_ERR if successful)

11.2.3 MBRT_errcode MBRT_IconDisplay (Uint8 * *icon*)

Changes the small (16x16) icon (black and white) in the upper left corner of the display. This allows to inform the user of some events. The icon is an array of 256 integers. A 0 value will means the pixel is not drawn, non 0 are drawn using the OSD color.

Parameters:

icon the 16x16 icon pixels array

Returns:

An error code (NO_ERR if successful)

11.2.4 MBRT_errcode MBRT_PaletteDisplay (Uint8 * *palette*)

Changes the palette associated with the display

Parameters:

palette an array containing the complete palette definition (256*3 integers)

Returns:

An error code (NO_ERR if successful)

11.2.5 MBRT_errcode MBRT_PollDisplay (MBRT_eventcode * *event_code*)

Handles event that have occurred in the display

Parameters:

event_code an integer representing a specific event (output)

Returns:

An error code (NO_ERR if successful)

11.2.6 MBRT_errcode MBRT_UpdateDisplay (MB_Image * *src*, double *wfps*, double * *ofps*)

Updates the display with the content of a given mamba image structure

Parameters:

src the image displayed

wfps input the desired framerate

ofps output the framerate

Returns:

An error code (NO_ERR if successful)

12 MBRT_error.cpp File Reference

```
#include "MBRT_error.h"
```

Functions

- char * **MBRT_StrErr** (**MBRT_errcode** error_nb)

Variables

- char * **err_str** []

12.1 Detailed Description

Author:

Nicolas Beucher

Date:

3-28-2009

12.2 Function Documentation

12.2.1 char* MBRT_StrErr (MBRT_errcode *error_nb*)

Returns an explanation of the error code

12.3 Variable Documentation

12.3.1 char* err_str[]

Error value interpretation

13 MBRT_error.h File Reference

Enumerations

- enum **MBRT_errcode** {

  NO_ERR, **ERR_CANT_CREATE_CONTEXT**, **ERR_INVD_CTX**, **ERR_INIT_DISPLAY**,

  ERR_FORMAT_DISPLAY, **ERR_LOCK_DISPLAY**, **ERR_INVALID_DISPLAY**, **ERR_VID**,

  ERR_DSHOW_FILTER_GRAPH, **ERR_DSHOW_CAPT_GRAPH_BUILD**, **ERR_DSHOW_INIT_CAPT_GRAPH_BUILD**, **ERR_DSHOW_DEV_ENUM**,

  ERR_DSHOW_VIDCAP_ENUM, **ERR_DSHOW_DEV_NOT_FOUND**, **ERR_PAL_VID**, **ERR_VID_TYPE**,

  ERR_DEPTH, **ERR_SIZE**, **ERR_AVC_VID_OPEN**, **ERR_AVC_STREAM_INFO**,

  ERR_AVC_NO_VID_STREAM, **ERR_AVC_NO_CODEC**, **ERR_AVC_CODEC_OPEN**, **ERR_AVC_FRAME_ALLOC**,

  ERR_AVC_DECODING, **ERR_AVC_REC_FORMAT**, **ERR_AVC_REC_FMTCTX**, **ERR_AVC_REC_STREAM**,

ERR_AVC_REC_PARAM_SET, **ERR_AVC_REC_NO_CODEC**, **ERR_AVC_REC_CODEC_OPEN**, **ERR_AVC_REC_PICT_ALLOC**,

ERR_AVC_REC_FILE_OPEN, **ERR_AVC_REC_ENCODE**, **ERR_AVC_REC_INV_CTX**, **ERR_UNKNOWN** }

Functions

- char * **MBRT_StrErr** (**MBRT_errcode** error_nb)

13.1 Detailed Description

Date:

3-28-2009

This file contains the complete liste of error code returned by the API functions.

13.2 Enumeration Type Documentation

13.2.1 enum MBRT_errcode

Type definition for error code

Enumerator:

NO_ERR Value returned by function when no error was encountered.

ERR_CANT_CREATE_CONTEXT Cannot create (memory allocation) context

ERR_INVD_CTX Context is not properly initialized

ERR_INIT_DISPLAY Init SDL display error

ERR_FORMAT_DISPLAY SDL display cannot support requested format

ERR_LOCK_DISPLAY Locking screen for updating failure

ERR_INVALID_DISPLAY Incorrect display (not initialized) error

ERR_VID Video acquisition module error

ERR_DSHOW_FILTER_GRAPH Cannot create the directshow filter graph

ERR_DSHOW_CAPT_GRAPH_BUILD Cannot create the directshow capture graph builder

ERR_DSHOW_INIT_CAPT_GRAPH_BUILD Cannot initialize the directshow capture graph builder

ERR_DSHOW_DEV_ENUM Cannot create the directshow device enumerator

ERR_DSHOW_VIDCAP_ENUM Cannot create a directshow video capture device enumerator

ERR_DSHOW_DEV_NOT_FOUND Cannot found the directshow device

ERR_PAL_VID Unsupported palette format

ERR_VID_TYPE Type of the video acquisition is incorrect

ERR_DEPTH The depth of the mamba image given in argument is incompatible

ERR_SIZE The size of the mamba image given in argument is incompatible

ERR_AVC_VID_OPEN Cannot open the video file

ERR_AVC_STREAM_INFO Cannot retrieve video stream information

ERR_AVC_NO_VID_STREAM No video stream present in video file

ERR_AVC_NO_CODEC Cannot find appropriate video decoder

ERR_AVC_CODEC_OPEN Cannot open the video codec

ERR_AVC_FRAME_ALLOC Cannot allocate the video frame

ERR_AVC_DECODING Cannot decode video frame

ERR_AVC_REC_FORMAT Cannot find format for recording

ERR_AVC_REC_FMTCTX Cannot allocate format context for recording

ERR_AVC_REC_STREAM Cannot create stream for recording
ERR_AVC_REC_PARAM_SET Invalid output format parameters for recording
ERR_AVC_REC_NO_CODEC Cannot find appropriate video decoder for recording
ERR_AVC_REC_CODEC_OPEN Cannot open the video codec for recording
ERR_AVC_REC_PICT_ALLOC Cannot allocate picture for recording
ERR_AVC_REC_FILE_OPEN Cannot open the output file for recording
ERR_AVC_REC_ENCODE Cannot encode the image for recording
ERR_AVC_REC_INV_CTX Invalid recording context (not started)
ERR_UNKNOWN Unknown error

13.3 Function Documentation

13.3.1 char* MBRT_StrErr (MBRT_errcode *error_nb*)

Returns an explanation of the error code

14 MBRT_Record.cpp File Reference

```
#include "mambaRTApi_loc.h"
```

Functions

- **MBRT_errcode MBRT_RecordStart** (char *filename)
- **MBRT_errcode MBRT_RecordEnd** ()
- **MBRT_errcode MBRT_RecordImage** (**MB_Image** *src)

14.1 Detailed Description

Author:

Nicolas Beucher

Date:

17-04-2010

14.2 Function Documentation

14.2.1 MBRT_errcode MBRT_RecordEnd (void)

Ends the recording. Closes the codec and terminates the file.

Returns:

An error code (NO_ERR if successful)

14.2.2 MBRT_errcode MBRT_RecordImage (MB_Image * *src*)

Records an image

Parameters:

src the mamba image to record

Returns:

An error code (NO_ERR if successful)

14.2.3 MBRT_errcode MBRT_RecordStart (char * *filename*)

Starts the recording. This function activates the libavformat and libavcodec libraries to encode the image using MPEG2 codec (DVD format).

Parameters:

filename the path to the created video file

Returns:

An error code (NO_ERR if successful)

15 MBRT_VideoAcq.cpp File Reference

```
#include "mambaRTApi_loc.h"
```

Functions

- **MBRT_errcode MBRT_CreateVideoAcq** (char *device, **MBRT_vidType** type)
- **MBRT_errcode MBRT_DestroyVideoAcq** ()
- **MBRT_errcode MBRT_GetAcqSize** (int *acq_w, int *acq_h)
- **MBRT_errcode MBRT_GetAcqFrameRate** (double *ofps)
- **MBRT_errcode MBRT_GetImageFromAcq** (**MB_Image** *dest)
- **MBRT_errcode MBRT_StartAcq** ()
- **MBRT_errcode MBRT_StopAcq** ()

15.1 Detailed Description

Author:

Nicolas Beucher

Date:

03-27-2009

15.2 Function Documentation

15.2.1 MBRT_errcode MBRT_CreateVideoAcq (char * *device*, MBRT_vidType *type*)

Fillss the video acquisition structure with the parameters of the given device and initializes it.

Parameters:

device the video device (usually /dev/video0) or the video file.

types the type of device (e.g video4linux, video4linux2, avc ...)

Returns:

an error code (NO_ERR if successful)

15.2.2 MBRT_errcode MBRT_DestroyVideoAcq (void)

Closes the acquisition device and resets the structure

Returns:

NO_ERR if successful

15.2.3 MBRT_errcode MBRT_GetAcqFrameRate (double * *ofps*)

Returns the acquisition device default framerate.

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

15.2.4 MBRT_errcode MBRT_GetAcqSize (int * *acq_w*, int * *acq_h*)

Returns the acquisition device resolution.

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

15.2.5 MBRT_errcode MBRT_GetImageFromAcq (MB_Image * *dest*)

Obtains an image form the acquisition device

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

15.2.6 MBRT_errcode MBRT_StartAcq ( )

Starts the acquisition device capture process

Returns:

NO_ERR if successful

15.2.7 MBRT_errcode MBRT_StopAcq ( )

Stops the acquisition device capture process

Returns:

NO_ERR if successful

16 MBRT_VideoAcq_avc.cpp File Reference

```
#include "mambaRTApi_loc.h"
```

Functions

- **MBRT_errcode MBRT_CreateVideoAcq_avc** (char *video_path)
- **MBRT_errcode MBRT_DestroyVideoAcq_avc** ()
- **MBRT_errcode MBRT_GetAcqSize_avc** (int *acq_w, int *acq_h)
- **MBRT_errcode MBRT_GetAcqFrameRate_avc** (double *ofps)
- **MBRT_errcode MBRT_GetImageFromAcq_avc** (**MB_Image** *dest)
- **MBRT_errcode MBRT_StartAcq_avc** ()
- **MBRT_errcode MBRT_StopAcq_avc** ()

16.1 Detailed Description

Author:

Nicolas Beucher

Date:

04-01-2010

16.2 Function Documentation

16.2.1 MBRT_errcode MBRT_CreateVideoAcq_avc (char * *video_path*)

Opens a video file as a video acquisition device using the advanced video codec (AVC) library.

Parameters:

video_path path to the video file (supported codec depends on your local implementation of the libavcodec)

Returns:

an error code (NO_ERR if successful)

16.2.2 MBRT_errcode MBRT_DestroyVideoAcq_avc (void)

Closes the video file playback (AVC) and reset the structure

Returns:

NO_ERR if successful

16.2.3 MBRT_errcode MBRT_GetAcqFrameRate_avc (double * *ofps*)

Returns the acquisition device default framerate (AVC).

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

16.2.4 MBRT_errcode MBRT_GetAcqSize_avc (int * *acq_w*, int * *acq_h*)

Returns the video played size (AVC).

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

16.2.5 MBRT_errcode MBRT_GetImageFromAcq_avc (MB_Image * *dest*)

Obtains an image from the video (AVC)

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

16.2.6 MBRT_errcode MBRT_StartAcq_avc ( )

Starts the acquisition device capture process (AVC)

Returns:

NO_ERR if successful

16.2.7 MBRT_errcode MBRT_StopAcq_avc ( )

Stops the acquisition device capture process (AVC)

Returns:

NO_ERR if successful

17 MBRT_VideoAcq_dshow.cpp File Reference

```
#include "mambaRTApi_loc.h"
```

Defines

- #define **RETURN_ON_ERR**(hr, err_type)

Functions

- **MBRT_errcode MBRT_CreateVideoAcq_dshow** (int device)
- **MBRT_errcode MBRT_StopAcq_dshow** ()
- **MBRT_errcode MBRT_StartAcq_dshow** ()
- **MBRT_errcode MBRT_GetImageFromAcq_dshow** (**MB_Image** *dest)
- **MBRT_errcode MBRT_DestroyVideoAcq_dshow** ()
- **MBRT_errcode MBRT_GetAcqSize_dshow** (int *acq_w, int *acq_h)
- **MBRT_errcode MBRT_GetAcqFrameRate_dshow** (double *ofps)

17.1 Detailed Description

Author:

Nicolas Beucher

Date:

03-27-2009

17.2 Define Documentation

17.2.1 #define RETURN_ON_ERR(*hr*, *err_type*)

Value:

```
if (FAILED(hr)) { \
        STOP_AND_CLEANUP(); \
        return err_type; \
    } \
```

17.3 Function Documentation

17.3.1 MBRT_errcode MBRT_CreateVideoAcq_dshow (int *device*)

Fills the video acquisition structure with the parameters of the given device and initializes it (DSHOW).

Parameters:

device the video device (usually 0)

Returns:

an error code (NO_ERR if successful)

17.3.2 MBRT_errcode MBRT_DestroyVideoAcq_dshow (void)

Closes the acquisition device and reset the structure (DSHOW).

Returns:

NO_ERR if successful

17.3.3 MBRT_errcode MBRT_GetAcqFrameRate_dshow (double * *ofps*)

Returns the acquisition device default framerate (DSHOW).

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

17.3.4 MBRT_errcode MBRT_GetAcqSize_dshow (int * *acq_w*, int * *acq_h*)

Returns the acquisition device resolution (DSHOW).

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

17.3.5 MBRT_errcode MBRT_GetImageFromAcq_dshow (MB_Image * *dest*)

Obtains an image from the acquisition device (DSHOW).

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

17.3.6 MBRT_errcode MBRT_StartAcq_dshow ( )

Starts the acquisition device capture process (DSHOW).

Returns:

NO_ERR if successful

17.3.7 MBRT_errcode MBRT_StopAcq_dshow ( )

Stops the acquisition device capture process (DSHOW).

Returns:

NO_ERR if successful

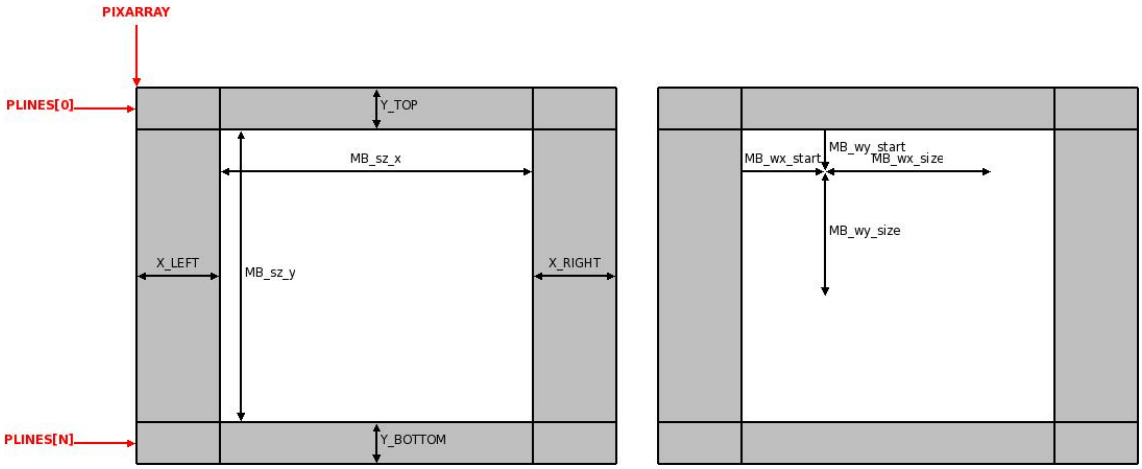

Figure 1: Image structure and variables