BebopDroneDecodeStream

Generated by Doxygen 1.8.9.1

Wed May 25 2016 18:10:31

Contents

Chapter 1

Purpose of BebopDroneDecodeStream

This sample shows how to receive video stream from a Bebop Drone, decode it, display the decoded stream with mplayer and get inputs from users to control the Bebop drone.

To compile SDK Example BebopDroneDecodeStream

On Linux and MacOS X platform : make

Dependencies of BebopDroneDecodeStream

You will need **mplayer** to show the video stream, **ffmpeg** to get the video decoded and **ncurse** to get inputs from console

On Linux you can get ncurses-dev apt-get: apt-get install ncurses-dev

To run SDK Example BebopDroneDecodeStream

On Linux and MacOS X platform : make run

To clean the compilation of BebopDroneDecodeStream

On Linux and MacOS X platform : make clean

Discussion about BebopDroneDecodeStream

This project is separated into 3 classes :

- BebopDroneDecodeStream : This is the main class. It will operate the connexion to the drone, the setup of the network and video part. It will also register for commands callback and send commands. If you need to add callbacks, add it in registerARCommandsCallbacks.

In this class, the callback for a new frame received will be called. When it does, it will take a free empty frame from a pool (to avoid an allocation each time a frame is received), put the received data in this free frame and add the free frame to a frame buffer. An other thread will loop, and try to take a frame and pass it to the decoder. Once the

frame has been decoded, it will write the decoded frame in a pipe file. MPlayer will read this pipe file and display the frames.

- DecoderManager This is an util class that will perform h264 frame decoding thanks to ffmpeg.
- ihm This is an util class that will catch inputs from console and send these events to BebopDroneDecode↩ Stream. It will also display some pieces of information about the drone, like its battery level and its flying state.

When you run this sample, be sure to be in the console to catch your keyboard event. As MPlayer will open a new window, you could have to click again in the console. The ihm inputs are implemented on an azerty keyboard. Feel free to adapt it just by changing the key comparison in the function IHM_InputProcessing.

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

Chapter 4

Class Documentation

4.1 _ARCODECS_Manager_Component_t_ Struct Reference

Component of a frame.

`#include <DecoderManager.h>`

Public Attributes

- uint8_t * data
- uint32_t lineSize
- uint32_t size

4.1.1 Detailed Description

Component of a frame.

4.1.2 Member Data Documentation

4.1.2.1 uint8_t* _ARCODECS_Manager_Component_t_::data

data buffer

4.1.2.2 uint32_t _ARCODECS_Manager_Component_t_::lineSize

size of each line of the component

4.1.2.3 uint32_t _ARCODECS_Manager_Component_t_::size

size of the buffer

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

- DecoderManager.h

4.2 _ARCODECS_Manager_FFMPEGDecoder_t_ Struct Reference

Public Attributes

- AVCodec * **codec**
- AVCodecContext * **codecCtx**
- AVFrame * **decodedFrame**
- AVPacket **avpkt**
- uint8_t * **outputData**
- int **outputDataSize**

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

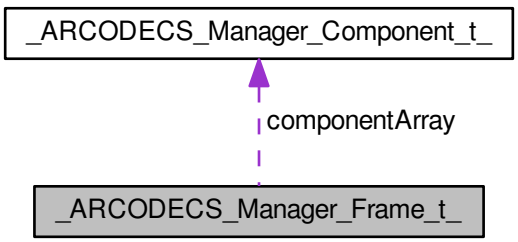
- DecoderManager.c

4.3 _ARCODECS_Manager_Frame_t_ Struct Reference

Video and audio codecs manager allow to decode video and audio.

`#include <DecoderManager.h>`

Collaboration diagram for _ARCODECS_Manager_Frame_t_:



Public Attributes

- eARCODECS_FORMAT **format**
- uint32_t **width**
- uint32_t **height**
- uint32_t **numberOfComponent**
- ARCODECS_Manager_Component_t * **componentArray**

4.3.1 Detailed Description

Video and audio codecs manager allow to decode video and audio.

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

- DecoderManager.h

4.4 _ARDrone3CameraData_t_ Struct Reference

Public Attributes

- int **tilt**
- int **pan**

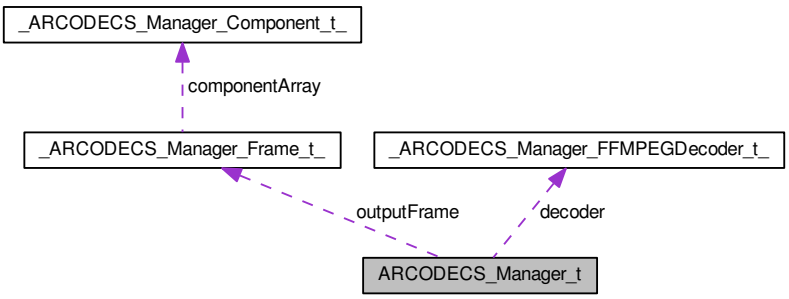
The documentation for this struct was generated from the following file:

- BebopDroneDecodeStream.h

4.5 ARCODECS_Manager_t Struct Reference

Video and audio codecs manager structure allow to decode video and audio.

Collaboration diagram for ARCODECS_Manager_t:



Public Attributes

- ARCODECS_Manager_GetNextDataCallback_t **callback**
- void * **callbackCustomData**
- ARCODECS_Manager_FFMPEGDecoder_t * **decoder**
- ARCODECS_Manager_Frame_t **outputFrame**

4.5.1 Detailed Description

Video and audio codecs manager structure allow to decode video and audio.

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

- DecoderManager.c

4.6 BD_MANAGER_t Struct Reference

Collaboration diagram for BD_MANAGER_t:

Public Attributes

- ARNETWORKAL_Manager_t * **alManager**
- ARNETWORK_Manager_t * **netManager**
- ARSTREAM_Reader_t * **streamReader**
- ARSAL_Thread_t **looperThread**
- ARSAL_Thread_t **rxThread**
- ARSAL_Thread_t **txThread**
- ARSAL_Thread_t **videoTxThread**
- ARSAL_Thread_t **videoRxThread**
- int **d2cPort**
- int **c2dPort**
- int **arstreamFragSize**
- int **arstreamFragNb**
- int **arstreamAckDelay**
- uint8_t * **videoFrame**
- uint32_t **videoFrameSize**
- BD_PCMD_t **dataPCMD**
- BD_Cam_t **dataCam**
- ARCODECS_Manager_t * **decoder**
- int **decodingCanceled**
- ARSAL_Thread_t **decodingThread**
- int **hasReceivedFirstIFrame**
- RawFrame_t ** **freeRawFramePool**
- int **rawFramePoolCapacity**
- int **lastRawFrameFreeIdx**
- RawFrame_t ** **rawFrameFifo**
- int **fifoReadIdx**
- int **fifoWriteIdx**
- eARCOMMANDS_ARDRONE3_PILOTINGSTATE_FLYINGSTATECHANGED_STATE **flyingState**
- FILE * **video_out**
- ARSAL_Mutex_t **mutex**
- ARSAL_Thread_t * **readerThreads**
- READER_THREAD_DATA_t * **readerThreadsData**
- int **run**
- IHM_t * **ihm**

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

- BebopDroneDecodeStream.h

4.7 BD_PCMD_t Struct Reference

Public Attributes

- int **flag**
- int **roll**
- int **pitch**
- int **yaw**
- int **gaz**

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

- BebopDroneDecodeStream.h

4.8 IHM_t Struct Reference

Public Attributes

- WINDOW * **mainWindow**
- ARSAL_Thread_t **inputThread**
- int **run**
- IHM_onInputEvent_t **onInputEventCallback**
- void * **customData**

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

- ihm.h

4.9 RawFrame_t Struct Reference

Component of a frame.

Public Attributes

- uint8_t * data
- uint32_t size
- uint8_t **isIframe**

4.9.1 Detailed Description

Component of a frame.

4.9.2 Member Data Documentation

4.9.2.1 uint8_t* RawFrame_t::data

data buffer

4.9.2.2 uint32_t RawFrame_t::size

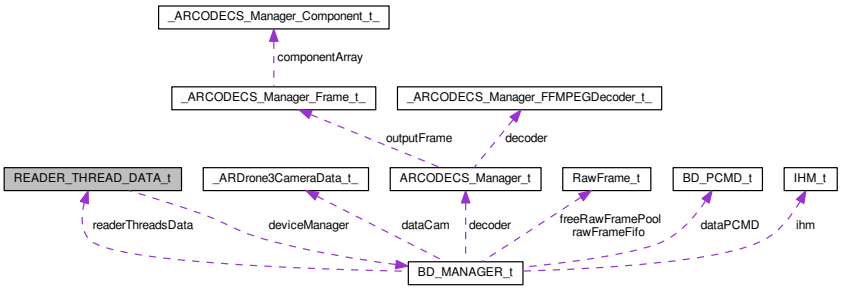
size of the buffer

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

- BebopDroneDecodeStream.c

4.10 READER_THREAD_DATA_t Struct Reference

Collaboration diagram for READER_THREAD_DATA_t:



Public Attributes

- BD_MANAGER_t * **deviceManager**
- int **readerBufferId**

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

- BebopDroneDecodeStream.h

Chapter 5

File Documentation

5.1 BebopDroneDecodeStream.c File Reference

This file contains sources about basic arsdk example decoding video stream from a BebopDrone with ffmpeg.

```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
#include <libavformat/avformat.h>
#include <libavcodec/avcodec.h>
#include <libavutil/imgutils.h>
#include <libARSAL/ARSAL.h>
#include <libARSAL/ARSAL_Print.h>
#include <libARNetwork/ARNetwork.h>
#include <libARNetworkAL/ARNetworkAL.h>
#include <libARDiscovery/ARDiscovery.h>
#include <libARStream/ARStream.h>
#include "BebopDroneDecodeStream.h"
```

Include dependency graph for BebopDroneDecodeStream.c:

Classes

- struct RawFrame_t

  Component of a frame.

Macros

- #define **TAG** "BebopDroneReceiveStream"
- #define **BD_IP_ADDRESS** "192.168.42.1"
- #define **BD_DISCOVERY_PORT** 44444
- #define **BD_C2D_PORT** 54321
- #define **BD_D2C_PORT** 43210

- #define **BD_NET_CD_NONACK_ID** 10
- #define **BD_NET_CD_ACK_ID** 11
- #define **BD_NET_CD_EMERGENCY_ID** 12
- #define **BD_NET_CD_VIDEO_ACK_ID** 13
- #define **BD_NET_DC_NAVDATA_ID** 127
- #define **BD_NET_DC_EVENT_ID** 126
- #define **BD_NET_DC_VIDEO_DATA_ID** 125
- #define **BD_NET_DC_VIDEO_FRAG_SIZE** 65000
- #define **BD_NET_DC_VIDEO_MAX_NUMBER_OF_FRAG** 4
- #define **BD_RAW_FRAME_BUFFER_SIZE** 50
- #define **BD_RAW_FRAME_POOL_SIZE** 50
- #define **ERROR_STR_LENGTH** 2048
- #define **FIFO_DIR_PATTERN** "/tmp/arsdk_XXXXXX"
- #define **FIFO_NAME** "arsdk_fifo"

Functions

- int getNextDataCallback (uint8_t **data, void *customData)
- void * **Decode_RunDataThread** (void *customData)
- RawFrame_t * **getFreeRawFrame** (BD_MANAGER_t *deviceManager)
- void **addFreeRawFrameToFifo** (BD_MANAGER_t *deviceManager, RawFrame_t *rawFrame)
- void **flushFifo** (BD_MANAGER_t *deviceManager)
- void **putRawFrameBackToPool** (BD_MANAGER_t *deviceManager, int fifoIdx)
- RawFrame_t * **getFrameFromData** (BD_MANAGER_t *deviceManager, uint8_t *data)
- int **main** (int argc, char *argv[])
- int ardiscoveryConnect (BD_MANAGER_t *deviceManager)
- eARDISCOVERY_ERROR **ARDISCOVERY_Connection_SendJsonCallback** (uint8_t *dataTx, uint32_t *dataTxSize, void *customData)
- eARDISCOVERY_ERROR **ARDISCOVERY_Connection_ReceiveJsonCallback** (uint8_t *dataRx, uint32_t dataRxSize, char *ip, void *customData)
- int startNetwork (BD_MANAGER_t *deviceManager)
- void **stopNetwork** (BD_MANAGER_t *deviceManager)
- void **onDisconnectNetwork** (ARNETWORK_Manager_t *manager, ARNETWORKAL_Manager_t *alManager, void *customData)
- int startVideo (BD_MANAGER_t *deviceManager)
- void **stopVideo** (BD_MANAGER_t *deviceManager)
- uint8_t * **frameCompleteCallback** (eARSTREAM_READER_CAUSE cause, uint8_t *frame, uint32_t frameSize, int numberOfSkippedFrames, int isFlushFrame, uint32_t *newBufferCapacity, void *custom)
- int **sendPCMD** (BD_MANAGER_t *deviceManager)
- int **sendCameraOrientation** (BD_MANAGER_t *deviceManager)
- int **sendDate** (BD_MANAGER_t *deviceManager)
- int **sendAllSettings** (BD_MANAGER_t *deviceManager)
- int **sendAllStates** (BD_MANAGER_t *deviceManager)
- int **sendBeginStream** (BD_MANAGER_t *deviceManager)
- int **sendTakeoff** (BD_MANAGER_t *deviceManager)
- int **sendLanding** (BD_MANAGER_t *deviceManager)
- int **sendEmergency** (BD_MANAGER_t *deviceManager)
- eARNETWORK_MANAGER_CALLBACK_RETURN arnetworkCmdCallback (int buffer_id, uint8_t *data, void *custom, eARNETWORK_MANAGER_CALLBACK_STATUS cause)
- void registerARCommandsCallbacks (BD_MANAGER_t *deviceManager)
- void **unregisterARCommandsCallbacks** (void)
- void **batteryStateChangedCallback** (uint8_t percent, void *custom)
- void **flyingStateChangedCallback** (eARCOMMANDS_ARDRONE3_PILOTINGSTATE_FLYINGSTATECHANGED_STATE state, void *custom)
- int startDecoder (BD_MANAGER_t *deviceManager)
- void **stopDecoder** (BD_MANAGER_t *deviceManager)
- void onInputEvent (eIHM_INPUT_EVENT event, void *customData)
- int **customPrintCallback** (eARSAL_PRINT_LEVEL level, const char *tag, const char *format, va_list va)

Variables

- int **gIHMRun** = 0
- char **gErrorStr** [ERROR_STR_LENGTH]

5.1.1 Detailed Description

This file contains sources about basic arsdk example decoding video stream from a BebopDrone with ffmpeg.

Date

08/01/2015

5.1.2 Function Documentation

5.1.2.1 int ardiscoveryConnect (BD_MANAGER_t * *deviceManager*)

Connection part

5.1.2.2 eARNETWORK_MANAGER_CALLBACK_RETURN arnetworkCmdCallback (int *buffer_id,* uint8_t * *data,* void * *custom,* eARNETWORK_MANAGER_CALLBACK_STATUS *cause*)

Commands part

5.1.2.3 int getNextDataCallback (uint8_t ** *data,* void * *customData*)

Handle older ffmpeg/libav versions

5.1.2.4 void onInputEvent (eIHM_INPUT_EVENT *event,* void * *customData*)

IHM callbacks

5.1.2.5 void registerARCommandsCallbacks (BD_MANAGER_t * *deviceManager*)

Commands callback part

5.1.2.6 int startDecoder (BD_MANAGER_t * *deviceManager*)

decoding part

5.1.2.7 int startNetwork (BD_MANAGER_t * *deviceManager*)

Network part

5.1.2.8 int startVideo (BD_MANAGER_t * *deviceManager*)

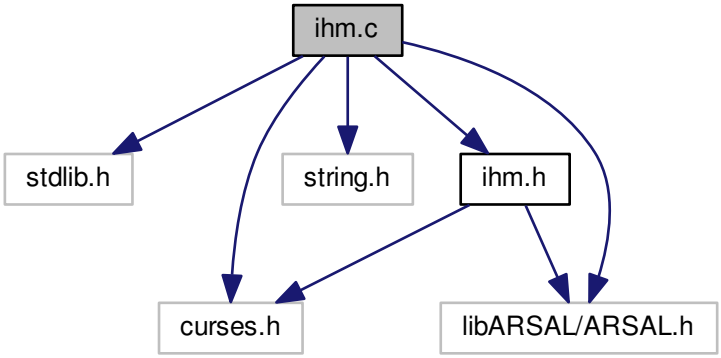
Video part

5.2 ihm.c File Reference

This file contains sources about ncurses IHM used by arsdk example "BebopDroneDecodeStream".

```
#include <stdlib.h>
#include <curses.h>
#include <string.h>
#include <libARSAL/ARSAL.h>
#include "ihm.h"
```

Include dependency graph for ihm.c:



Macros

- #define **HEADER_X** 0
- #define **HEADER_Y** 0
- #define **INSTRUCTION_X** 0
- #define **INSTRUCTION_Y** 2
- #define **BATTERY_X** 0
- #define **BATTERY_Y** 10
- #define **INFO_X** 0
- #define **INFO_Y** 12

Functions

- void * **IHM_InputProcessing** (void *data)
- IHM_t * **IHM_New** (IHM_onInputEvent_t onInputEventCallback)
- void **IHM_Delete** (IHM_t **ihm)
- void **IHM_setCustomData** (IHM_t *ihm, void *customData)
- void **IHM_PrintHeader** (IHM_t *ihm, char *headerStr)
- void **IHM_PrintInstruction** (IHM_t *ihm, char *instructionStr)
- void **IHM_PrintInfo** (IHM_t *ihm, char *infoStr)
- void **IHM_PrintBattery** (IHM_t *ihm, uint8_t percent)

5.2.1 Detailed Description

This file contains sources about ncurses IHM used by arsdk example "BebopDroneDecodeStream".

Date

15/01/2015

Index