SDK API User's Guide (Movie Player)

Version 0.92.0

Display Audio

Solution Team

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Release information

The following changes have been make to this document.

Change History

Date	Change
24 January 2018	V0.92.0 Modified API functions
11 May 2015	v0.91.0. Insert Scenario
21 April 2015	First release for v0.90.0

Proprietary Notice

Information in this document is provided solely to enable system and software implementers to use Nexell products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Nexell reserves the right to make changes without further notice to any products herein.

Nexell makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Nexell assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Nexell data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Nexell does not convey any license under its patent rights nor the rights of others. Nexell products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Nexell product could create a situation where personal injury or death may occur. Should Buyer purchase or use Nexell products for any such unintended or unauthorized application, Buyer shall indemnify and hold Nexell and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Nexell was negligent regarding the design or manufacture of the part.

Copyrightⓒ 2017 Nexell Co.,Ltd. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Nexell.

CONFIDENTIAL

Contact us

[11595] BundangYemiji Bldg. 12F, 31 Hwangsaeul-ro 258 beon gil, Bundang-gu, Sungnam-city, Gyeonggi-do, Korea.

TEL: 82-31-698-7400

FAX:82-31-698-7455

http://www.nexell.co.kr

CONFIDENTIAL

Contents

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Chap 1. Overview

1.1 Overview

이 문서는 Android/Linux 환경에서 Filter Library 를 사용하여 각종 동영상 및 음악 파일을 재생하기 위한 API 의 사용법을 설명한 문서이다.

1.2 지원범위

1.2.1 . Container

1.2.1.1 Movie Container

- ASF, AVI, MKV, MP4, RM(RealMdeia), FLV, MPEG-PS, MPEG-TS

1.2.1.2 Audio Container

- Mp3, flac, aac, flac, ogg, wav, wma

1.2.2 Video Codec

- H.264 : 1920x1080, 30fps
- H.263 : 1920x1088, 30fps
- MPEG2: 1920x1080, 30fps
- MPEG4: 1920x1080, 30fps
- FLV: 1920x1080, 30fps
- RealVideo: 1920x1080, 30fps
- VC1(WMV9): 1920x1080, 30fps
- Theora: 1280x720, 30fps

1.2.3 Audio Codec

- MP3 : ~48KHz, ~320Kbps, 2Ch
- AAC : ~96KHz, ~320Kbps, 5.1Ch
- AC3 : ~48KHz, ~256Kbps, 5.1Ch
- OGG : ~48KHz, 2Ch
- RealAudio : ~48KHz, ~256Kbps, 2Ch
- WMA : ~48KHz, ~192Kbps, 2Ch
- FLAC : ~96KHz, 2Ch

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

- PCM : ~96KHz, 2Ch
- DTS : ~96KHz, ~256Kbps, 5.1Ch
- COOK ; ~48KHz, ~256Kbps, 2Ch

1.3 Environment

이 API 의 동작 환경은 다음과 같다.

1.3.1 CPU

- S5P4418
- S5P6818

1.3.2 OS

- Kitkat
- Lollipop
- Linux(Kernel Version 3.4.39)

1.3.3 External Depend

- FFmpeg library (2.1.4)
- Audio Decoder (FFmpeg)

1.3.4 Library

- FFmpeg library: libavutil-2.1.4.so, libavresampld-2.1.4.so, libavdevice-2.1.4.so, libavcodec-2.1.4.so, libswscale-2.1.4.so, libswresample-2.1.4.so, libavfilter-2.1.4.so, libavformat-2.1.4.so
- Filter library :libnxfilter.so, libnxfilterhelper.so, libnxmpmanager.so
- Video decoder module driver & library : nx_vpu.ko, libnxvpu.so
- Video memory allocator library : libnxvmem.so
- Theora parser library: libtheoraparser.so, libtheoraparser_and.so
- Additional library (Linux)
  - V4L2 library : libnxv4l2.so, libnxdsp.so
  - Fine scaler library : libnxscaler.so

CONFIDENTIAL

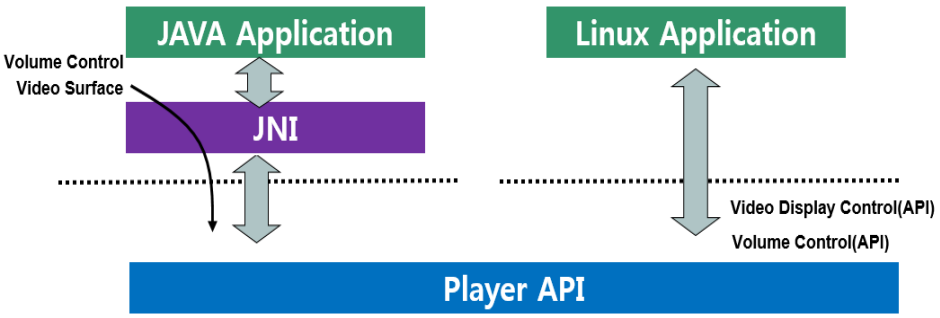
Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Chap 2. Structure

2.1 Overview

Library 의 사용에 대한 전체 구조를 설명한다.

2.2 Structure



Android : Video Control, Volume Control 은 Android 에서 제어한다.

Example Code 는 hardware\samsung_slsi\slsiap\apps\NxPlayerBasedFilter 에 있다.

Linux : Video Control, Volume Control 은 Player API 를 통해서 제어한다

Example Code 는 linux/platform/s5p4418/app/NxFilterPlayers 에 있다.

Consol 환경에서 테스트할수 있다.

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Chap 3. APIs

3.1 Overview

API 에 대한 세부 설명이다.

3.2 API Details

3.2.1 NX_MPOpen

Description:

Memory allocation 및 handle initialization

Prototype:

```
MP_RET NX_MPOpen(
    MP_HANDLE * phMp,
    void (*cbEvent)(void *pObj, uint32_t EventType, uint32_t EventData, uint32_t param),
    void *cbPrivate )
```

Parameters;

MP_ HANDLE *phMp: Movie player handle (input/output).

void (*cbEvent) :callback function(input) : End Of Stream/ Error 가 발생했을 때 callback function 을 통해 메시지가 전달된다.

CallBack Message 는 NX_MoviePlay.h 참고.

Return value:

Error Code.

```
MP_ERR_NONE
MP_ERR
```

3.2.2 NX_MPClose

Description:

Memory free.

Prototype:

```
void NX_MPClose( MP_HANDLE hMp)
```

Parameters:

MP_HANDLE hMp: Movie player handle (input/output)

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Return value:

None.

3.2.3 NX_MPSetUri

Description:

Uri setting

Prototype:

MP_RET NX_MPSetUri(MP_HANDLE hMp, const char *pUri)

Parameters;

MP_ HANDLE hMp: Movie player handle (input/output).

Const char *pUri: Uri(input).

Return value:

Error Code.

```
MP_ERR_NONE
MP_ERR
MP_ERR_INPUT_FILE
```

3.2.4 NX_MPGetMediaInfo

Description:

Media 정보를 얻어 옴.

Prototype:

MP_RESULT NX_MPGetMediaInfo(MP_HANDLE hMp, MP_MEDIA_INFO *pInfo)

Parameters:

MP_HANDLE hMp: Movie player handler(input/output).

MP_MEDIA_INFO *pInfo : media information(output).

Return value:

If exist media information , return MP_ERR_NONE. Otherwise return MP_ERR.

```
#define PROGRAM_MAX                 16
#define MAX_TRACK_NUM               10

typedef struct MP_TRACK_INFO {
        int32_t         iTrackIndex;    // Track Index
        int32_t         iTrackType;     // VIDEO:0, AUDIO: 1
        int32_t         iCodecId;
        int64_t         iDuration;      // Track Duration
        int32_t         iWidth;         // Only VideoTrack
```

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

```
        int32_t             iHeight;        // Only VideoTrack
        int32_t             iFrameRate;     // Only VideoTrack
        int32_t             iChannels;      // Only AudioTrack
        int32_t             iSampleRate;    // Only AudioTrack
        int32_t             iBitrate;       // Only AudioTrack
} MP_TRACK_INFO;

typedef struct MP_PROGRAM_INFO {
        int32_t             iAudioNum;
        int32_t             iVideoNum;
        int32_t             iSubTitleNum;
        int32_t             iDataNum;
        int64_t             iDuration;
        MP_TRACK_INFO       TrackInfo[MAX_TRACK_NUM];
} MP_PROGRAM_INFO;

typedef struct MP_MEDIA_INFO{
        int32_t             iProgramNum;
        int32_t             iAudioTrackNum;
        int32_t             iVideoTrackNum;
        int32_t             iSubTitleTrackNum;
        int32_t             iDataTrackNum;
        MP_PROGRAM_INFO     ProgramInfo[PROGRAM_MAX];
} MP_MEDIA_INFO;
```

3.2.5 NX_MPAddVideoTrack

3.2.5.1 Android

Description:

이 함수는 Video Track 을 추가하는 함수이다. NX_MPGetMediaInfo ()로부터 얻어온 media information 을 기반으로 track 의 index 를 추가하여 재생하고자 하는 track 의 pin 을 생성한다.

Track Index 는 MP_TRACK_INFO 구조체의 iTrackIndex 이다.

Prototype:

-. Android Case

MP_RESULT NX_ MPAddVideoTrack (

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

MP_HANDLE hMp,
int32_t iTrack,
ANativeWindow *pWindow,
MP_DSP_CONFIG *pInfo)

-. Linux Case

MP_RESULT NX_MPAddTrack(
MP_HANDLE hMp,
int32_t iTrack,
MP_DSP_CONFIG *pInfo,
Int32_t bHdmi = false)

Parameters:

MP_HANDLE hMp: Movie player handler(input/output).

int32_t iTrack: TrackInfo 의 iTrackIndex 를 의미한다(input).

ANativeWindow *pWindow : NativeWindow(input).

-. Android Surface 의 Native Window 를 사용할 때 사용한다. 만약 사용하지 않으면 NULL 입력한다.

-. Video Track 인 경우 재생하고자 하는 Surface 의 Native Window 를 할당한다.

-. Audio Track 인 경우에는 NULL 을 입력한다.

MP_DSP_CONFIG *pInfo : DSP Config(input)

-. MLC 를 사용할 때 사용한다. 만약 사용하지 않으면 NULL 입력한다.

-. Video Track 인 경우 재생하고자 하는 MLC 의 Display Information 을 할당한다.

-. Audio Track 인 경우에는 NULL 을 입력한다.

bHdmi : Linux 에서 Hdmi 를 사용시 사용(input) (true➔HdmiOn, false➔HdmiOff)
Android 경우는 false 설정한다.

Return value:

Error Code.

```
MP_ERR_NONE
MP_ERR
MP_NOT_SUPPORT_AUDIOCODEC
MP_NOT_SUPPORT_VIDEOCODEC
MP_NOT_SUPPORT_VIDEOWIDTH
MP_NOT_SUPPORT_VIDEOHEIGHT
```

.

```
typedef struct MP_DSP_RECT {
```

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

```
        int32_t                 iX;
        int32_t                 iY;
        int32_t                 iWidth;
        int32_t                 iHeight;
} MP_DSP_RECT;

typedef struct MP_DSP_CONFIG {
        int32_t                 iPort;     // 0:LCD, 1:HDMI
        int32_t                 iModule;   // 0:MLC0, 1:MLC1
        MP_DSP_RECT             srcRect;   // Source Crop Region
        MP_DSP_RECT             dstRect;   // Destination Position Region
} MP_DSP_CONFIG;
```

3.2.6 NX_MPAddAudioTrack

3.2.6.1 Android

Description:

이 함수는 Audio Track 을 추가하는 함수이다. NX_MPGetMediaInfo ()로부터 얻어온 media information 을 기반으로 track 의 index 를 추가하여 재생하고자 하는 track 의 pin 을 생성한다.

Track Index 는 MP_TRACK_INFO 구조체의 iTrackIndex 이다.

Prototype:

-. Linux Case

```
MP_RESULT     NX_MPAddTrack(
                        MP_HANDLE hMp,
                        int32_t iTrack,
                        MP_DSP_CONFIG *pInfo,
                        const char *pDeviceName)
```

Parameters:

MP_HANDLE hMp: Movie player handler(input/output).

int32_t iTrack: TrackInfo 의 iTrackIndex 를 의미한다(input).

MP_DSP_CONFIG *pInfo : DSP Config(input).

char *pDeviceName : audio device name

Return value:

Error Code.

```
MP_ERR_NONE
MP_ERR
MP_NOT_SUPPORT_AUDIOCODEC
MP_NOT_SUPPORT_VIDEOCODEC
```

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

```
   MP_NOT_SUPPORT_VIDEOWIDTH
   MP_NOT_SUPPORT_VIDEOHEIGHT
```

.

```
typedef struct MP_DSP_RECT {
        int32_t                 iX;
        int32_t                 iY;
        int32_t                 iWidth;
        int32_t                 iHeight;
} MP_DSP_RECT;

typedef struct MP_DSP_CONFIG {
        int32_t                 iPort;     // 0:LCD, 1:HDMI
        int32_t                 iModule;   // 0:MLC0, 1:MLC1
        MP_DSP_RECT             srcRect;   // Source Crop Region
        MP_DSP_RECT             dstRect;   // Destination Position Region
} MP_DSP_CONFIG;
```

3.2.7 NX_MPClearTrack

Description:

모든 track 을 delete 함.

Prototype:

MP_RESULT NX_MPClearTrack(MP_HANDLE hMp,)

Parameters:

MP_HANDLE hMp: Movie player handler(input/output).

Return value:

If success returns MP_ERR_NONE, otherwise returns MP_ERR.

3.2.8 NX_MPPlay

Description:

Play start.

Prototype:

MP_RESULT NX_MPPlay(MP_HANDLE hMp, float speed)

Parameters:

MP_HANDLE hMp: Movie player handler (input/outpu).

float speed : playing speed. The value 1.0 means normal speed(input). (other value not support yet)

(This functionality is not available yet.)

Return value:

If success returns MP_ERR_NONE, otherwise returns MP_ERR.

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

3.2.9 NX_MPStop

Description:

Play stop.

Prototype:

MP_RESULT NX_MPStop(MP_HANDLE hMp)

Parameter:

MP_HANDLE hMp: Movie player handler(input/output).

Return value:

If success returns MP_ERR_NONE, otherwise returns MP_ERR.

3.2.10 NX_MPPause

Description:

Paly pause.

Prototype:

MP_RESULT NX_MPPause(MP_HANDLE hMp)

Parameter:

MP_HANDLE hMp: Movie player handler(input/output).

Return value:

If success returns MP_ERR_NONE, otherwise returns MP_ERR.

3.2.11 NX_MPSeek

Description:

Play seeking

Prototype:

MP_RESULT NX_MPSeek(MP_HANDLE hMp, int64_t iSeekTime)

Parameter:

MP_HANDLE hMp: Movie player handler (input/output).

Int64_t iSeekTime: Seek time in milli-seconds(input).

Return value:

If success returns MP_ERR_NONE, otherwise returns MP_ERR.

3.2.12 NX_MPGetDuration

Description:

Media 의 play duration 을 얻어 옴.

Prototype:

MP_RESULT NX_MPGetDuration(MP_HANDLE hMp, int64_t *pDuration)

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Parameters:

MP_HANDLE hMp: Movie player handler (input/output)

Int64 *position: Contents duration in milli-seconds. (output)

Return value:

If success returns MP_ERR_NONE, otherwise returns MP_ERR.

3.2.13 NX_MPGetPosition

Description:

Current play position 을 얻어 옴.

Prototype:

MP_RESULT NX_MPGetPosition(MP_HANDLE hMp,int64_t *pPosition)

Parameters:

MP_HANDLE hMp: Movie player handler (input/output).

Int64_t *pPosition: Current play time in milli-seconds (output).

Return value:

If success returns MP_ERR_NONE, otherwise returns MP_ERR.

3.2.14 NX_MPAddSubDisplay

Description:

이 함수는 Video 를 MLC 에 직접 rendering 하는 경우 다른 MLC 장치에 복제하여 display 하기 위한 함수이다.

Prototype:

MP_RESULT NX_MPAddSubDisplay (
MP_HANDLE hMp,
Int32_t iTrack,
MP_DSP_CONFIG *pInfo
)

Parameters:

MP_HANDLE hMp: Movie player handler(input/output).

Int32_t iTrack: NX_MPAddTrack() 에서 사용한 iTrack 을 사용한다. (input).

MP_DSP_CONFIG *pInfo: (input).

Return value:

If success returns MP_ERR_NONE, otherwise return MP_ERR.

```
typedef struct MP_DSP_RECT {
        int32_t                 iX;
```

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

```
        int32_t                 iY;
        int32_t                 iWidth;
        int32_t                 iHeight;
} MP_DSP_RECT;

typedef struct MP_DSP_CONFIG {
        int32_t                 iPort;     // 0:LCD, 1:HDMI
        int32_t                 iModule;   // 0:MLC0, 1:MLC1
        MP_DSP_RECT             srcRect;   // Source Crop Region
        MP_DSP_RECT             dstRect;   // Destination Position Region
} MP_DSP_CONFIG;
```

3.2.15 NX_MPClearSubDisplay

Description:

Add 된 SubDisplay 장치를 제거한다.

Prototype:

MP_RESULT NX_MPClearSubDisplay (
MP_HANDLE hMp,
Int32_t iTrack
)

Parameters:

MP_HANDLE hMp: Movie player handler(input/output).

Int32_t iTrack: NX_MPAddTrack() 에서 사용한 iTrack 을 사용한다. (input).

Return value:

If success returns MP_ERR_NONE, otherwise return MP_ERR.

3.2.16 NX_MPSetDspCrop

Description:

이 함수는 MLC 로 직접 redering 하는 경우의 source image 를 crop 하는 함수이다.

Prototype:

MP_RESULT NX_MPSetDspCrop (
MP_HANDLE hMp,
Int32_t iTrack,
MP_DSP_RECT *pRect
)

Parameters:

MP_HANDLE hMp: Movie player handler(input/output).

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Int32_t iTrack: NX_MPAddTrack() 에서 사용한 iTrack 을 사용한다. (input).

MP_DSP_RECT *pRect: (input).

Return value:

If success returns MP_ERR_NONE, otherwise return MP_ERR.

3.2.17 NX_MPSetDspPosition

Description:

이 함수는 MLC 로 직접 rendering 하는 경우 rendering 되는 image 의 position 을 조절하는 함수이다.

Prototype:

```
MP_RESULT NX_MPSetDspPosition (
                                MP_HANDLE hMp,
                                Int32_t iTrack,
                                MP_DSP_RECT *pRect
                                )
```

Parameters:

MP_HANDLE hMp: Movie player handler(input/output).

Int32_t iTrack: NX_MPAddTrack() 에서 사용한 iTrack 을 사용한다. (input).

MP_DSP_RECT *pRect: (input).

Return value:

If success returns MP_ERR_NONE, otherwise return MP_ERR.

3.2.18 NX_MPSetVideoLayerPriority

Description:

이 함수는 MLC 로 직접 rendering 하는 경우 Video Layer 의 priority 를 조절하기 위한 함수이다.

Prototype:

```
MP_RESULT NX_MPSetVideoLayerPriority (
                                MP_HANDLE hMp,
                                Int32_t iTrack,
                                Int32_t iModule,
                                Int32_t iPriority
                                )
```

Parameters:

MP_HANDLE hMp: Movie player handler(input/output).

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Int32_t iTrack: NX_MPAddTrack() 에서 사용한 iTrack 을 사용한다. (input).

Int32_t iModule: Display Module (input). MLC0➔0, MLC1➔1

Int32_t iPriority: (input). 0, 1, or 2

Return value:

If success returns MP_ERR_NONE, otherwise return MP_ERR.

3.2.19 NX_MPSetVolume (Linux Only)

Description:

audio volume 조정.

Prototype:

MP_RESULT NX_MPSetVolume(MP_HANDLE hMp, int32_t iLevel)

Parameters:

MP_HANDLE hMp: Movie player handler. (input/output)

Int32_t iLevel: Volume value.(range 0 ~ 100, 0 means mute) (input)

Return value:

If success return MP_ERR_NONE, otherwise return MP_ERR.

3.2.20 NX_MPMakeThumbnail

Description:

Thumbnail 을 만듬

Jpeg(VPU).

Prototype:

```
int32_t    NX_MPMakeThumbnail(
                                const char *pInFile,
                                const char *pOutFile,
                                int32_t maxWidth,
                                int32_t maxHeight,
                                int32_t timeRatio
                                )
```

Parameters:

const char *pInFile: In File (input).

const char *pOutFile: Out File(input).

int32_t maxWidth: Max Width (input).

int32_t maxHeight: Max Height (input).

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

int32_t timeRatio: Time Ratio (input).

Return value:

If success return MP_ERR_NONE, otherwise return MP_ERR.

3.2.21 NX_MPGetVersion

Description:

Version 정보를 얻어 옴.

Prototype:

int32_t NX_MPGetVersion (void)

Parameters:

None.

Return value:

MSB| Major(8bit) - Minor(8bit) - Revision(8bit) - Reserved(8bit) |LSB.

3.2.22 NX_MPChgDebugLevel

Description:

Debug level 을 변경함.

Prototype:

void NX_MPChgDebugLevel (int32_t iLevel)

Parameters:

Int32_T iLevel: Debug Level. (input)

0(Verbose), 1(Debug), 2(Info), 3(Warn), 4(Error), 5(Disable) :: Default(Error)

Return value:

None.

3.2.23 NX_MPSetDspMode

Description:

Display mode 를 setting 함.

Prototype:

```
MP_RESULT       NX_MPSetDspMode (
                        MP_HANDLE hMp,
                        int32_t iTrack,
                        MP_DSP_CONFIG *pInfo,
                        int32_t iDspMode )
```

Parameters:

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

MP_HANDLE hMp : Movie player handler. (input/output)

int32_T iTrack: track number. (input)

MP_DSP_CONFIG *pInfo : display configurations. (input)

int32_t iDspMode : display mode(input).

0: default, 1: only LCD, 2: only HDMI, 3: Only TVOUT,

4: LCD+HDMI, 5: LCD_TVOUT

Return value:

If success return MP_ERR_NONE, otherwise return MP_ERR.

3.2.24 NX_MPSetRenderCallBack

Description:

외부 rendering 을 하는 경우 rendering callback 함수를 setting 함.

Prototype:

```
MP_RESULT       NX_MPSetRenderCallBack (
                        MP_HANDLE hMp,
                        int32_t iTrack,
                        void (*cbQtUpdateImg)(void *pImg))
```

Parameters:

MP_HANDLE hMp : Movie player handler. (input/output)

int32_T iTrack: track number. (input)

MP_DSP_CONFIG *pInfo : display configurations. (input)

void (*cbQtUpdateImg)(void *pImg) : 외부 rendering function (input)

Return value:

If success return MP_ERR_NONE, otherwise return MP_ERR.

3.2.25 NX_GetState

Description:

Player 의 status 를 얻어 옴.

Prototype:

```
int32_t    NX_GetState (MP_HANDLE hMp )
```

Parameters:

MP_HANDLE hMp : Movie player handler. (input/output)

Return value:

Player status (0: stop, 1: play, 2: pause, 3: ready).

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

3.2.26 NX_MPVideoMute

Description:

Video mute on/off 및 video mute off 인 경우 display init.

Prototype:

```
MP_RESULT       NX_MPVideoMute (
                        MP_HANDLE hMp,
                        int32_t bOnoff,
                        MP_DSP_CONFIG *pInfo )
```

Parameters:

MP_HANDLE hMp : Movie player handler. (input/output)\

int32_t bOnoff : video mute on/off flag.(input)

MP_DSP_CONFIG *pInfo : display configurations.(input)

Return value:

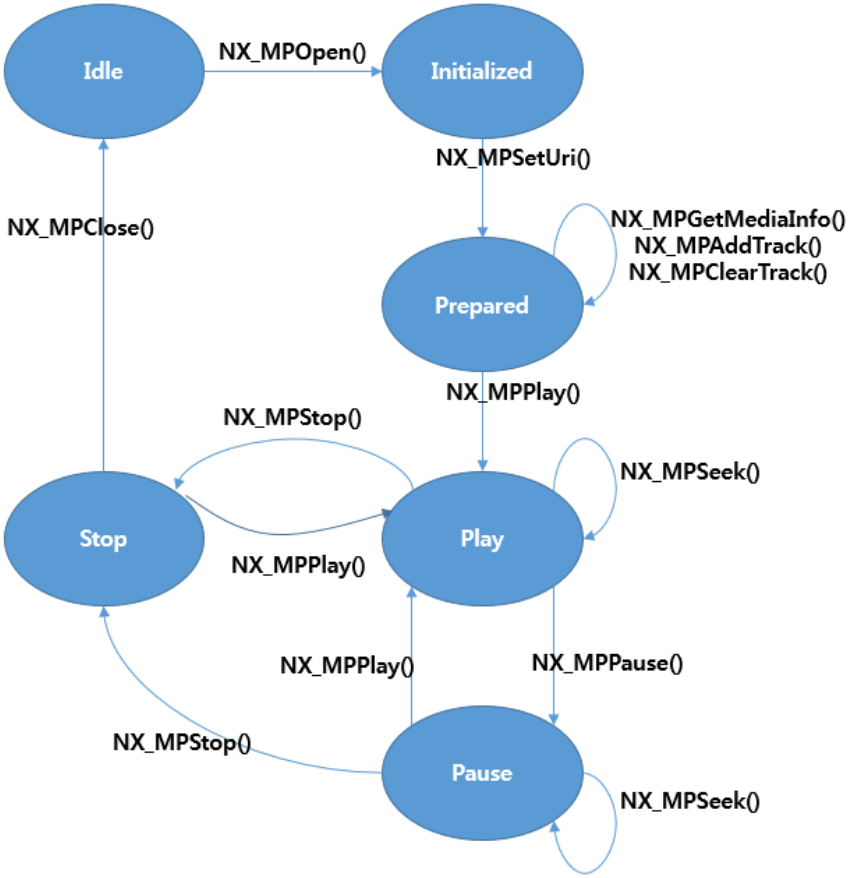
None

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Chap 4. State Diagram

4.1 Media Player State Diagram



Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

Chap 5. Scenario

5.1 Video Only, Audio Only, Video + Audio (Android)

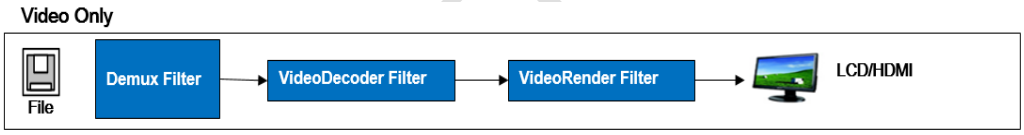
5.1.1 Video Surface

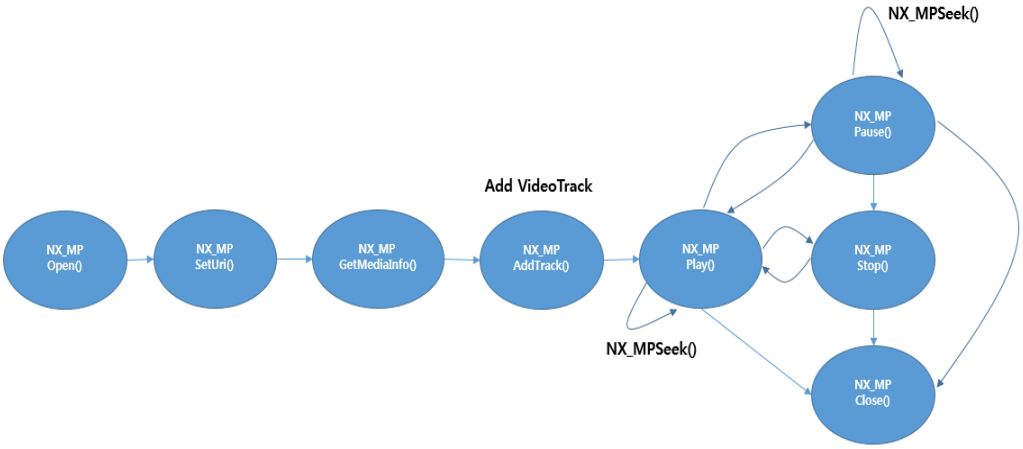
Android Surface 를 사용해서 Display 하는 Scenario 이다.

5.1.1.1 Video Only

Surface 를 사용하는 경우는 NX_MPAddTrack() 함수 인자를 Android Suface(NativeWindow)을 전달해야 한다.

아래 그림은 Video Only 함수 호출 순서이다.



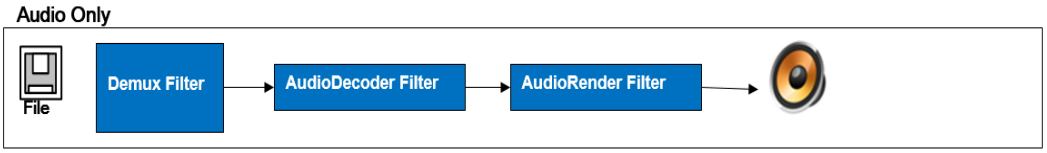


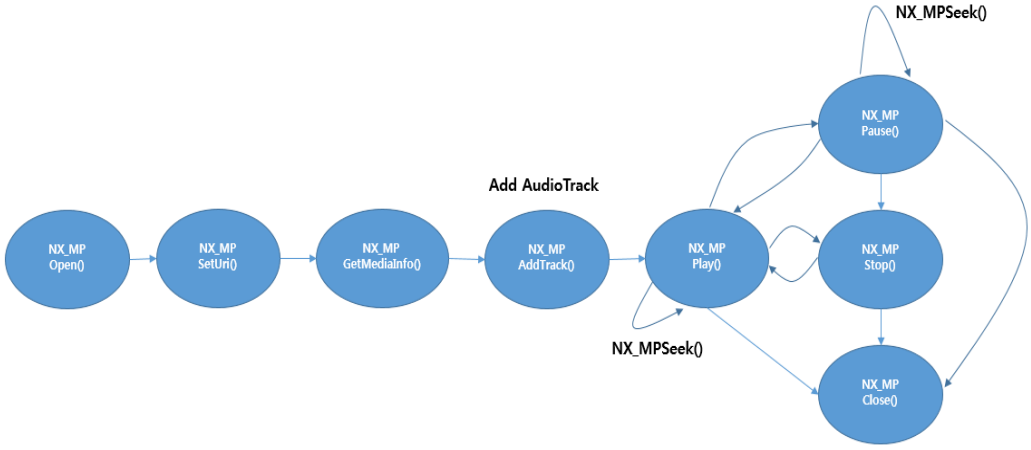
5.1.1.2 Audio Only

Android 경우 Volume 제어는 Android System 에서 제어한다.

아래 그림은 Audio Only 함수 호출 순서이다.

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.



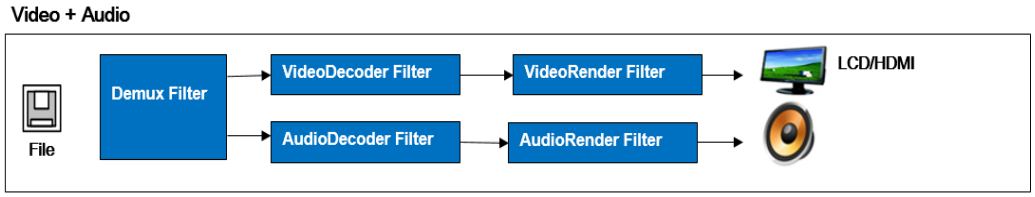


5.1.1.3 Video + Audio

Surface 를 사용하는 경우는 NX_MPAddTrack() 함수 인자를 Android Suface(NativeWindow)을 전달해야 한다.

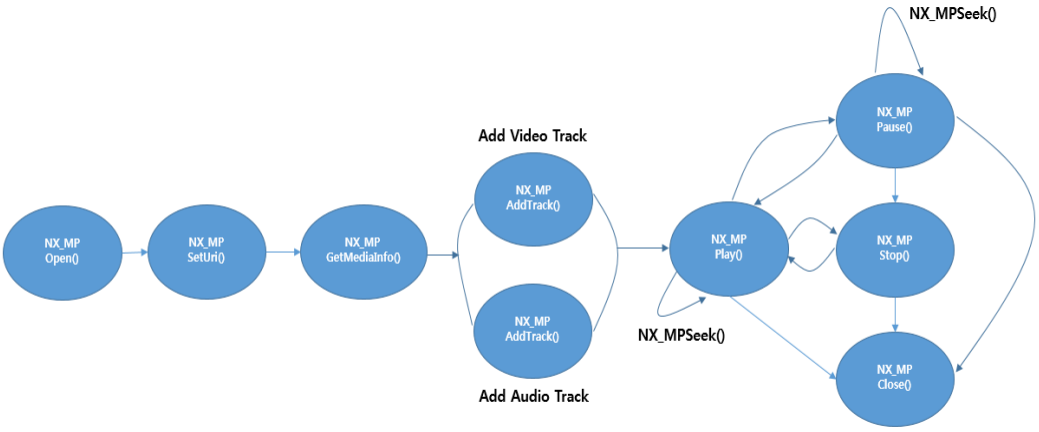
Android 경우 Volume 제어는 Android System 에서 제어한다.

아래 그림은 Video + Audio 인 경우 함수 호출 순서이다.



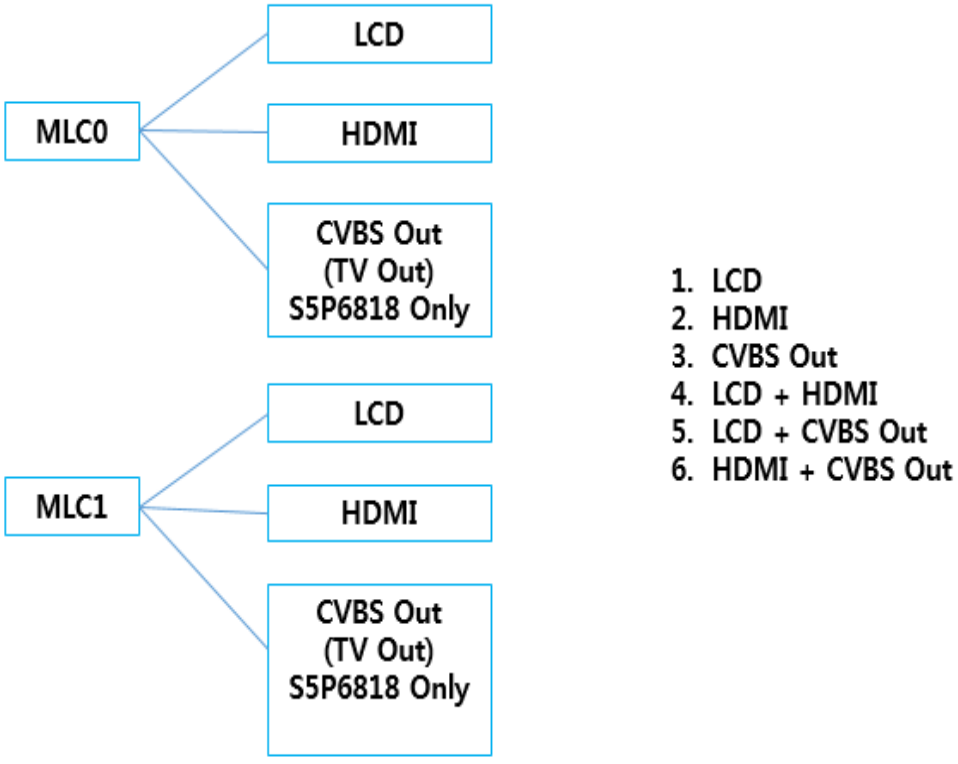
CONFIDEN

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.



5.1.2 Video MLC

Android 에서 MLC 를 사용해서 Display 하는 Scenario 이다.



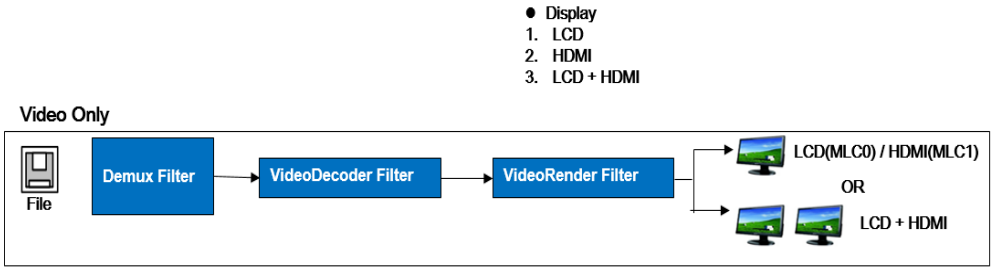
Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

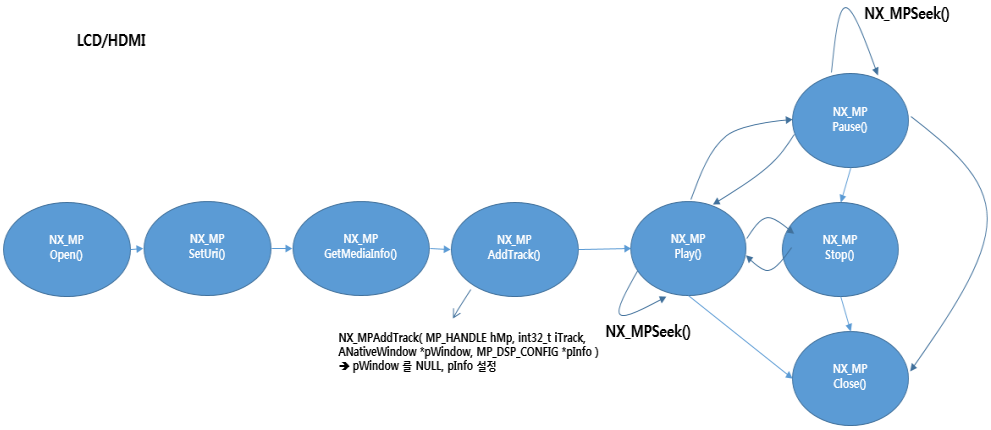
5.1.2.1 Video Only

MLC 를 사용하는 경우는 NX_MPAddTrack() 함수 인자를 MP_DSP_CONFIG *pInfo 을 설정한후 전달해야 한다.

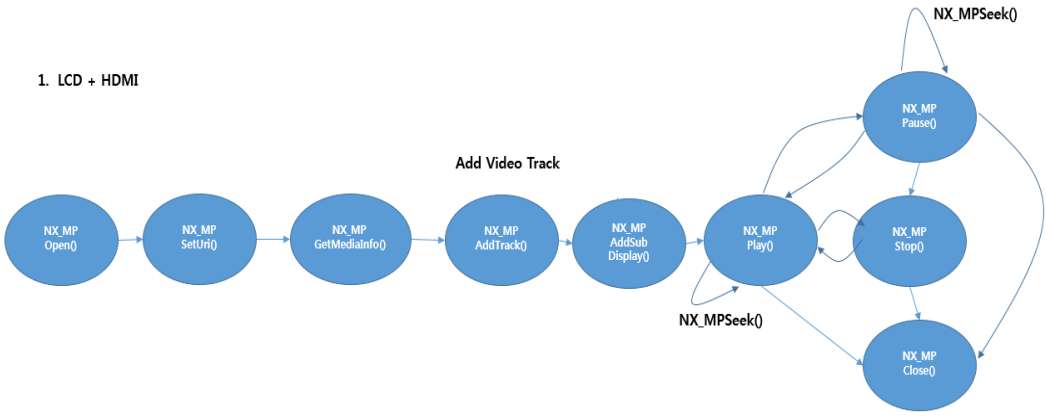
아래그림은 MLC 를 사용해서 Display 할때 사용하는 함수 호출 순서이다.

DualDisplay(LCD + HDMI) 인 경우는 NX_MPAddSubDisplay() 함수를 사용한다.





Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

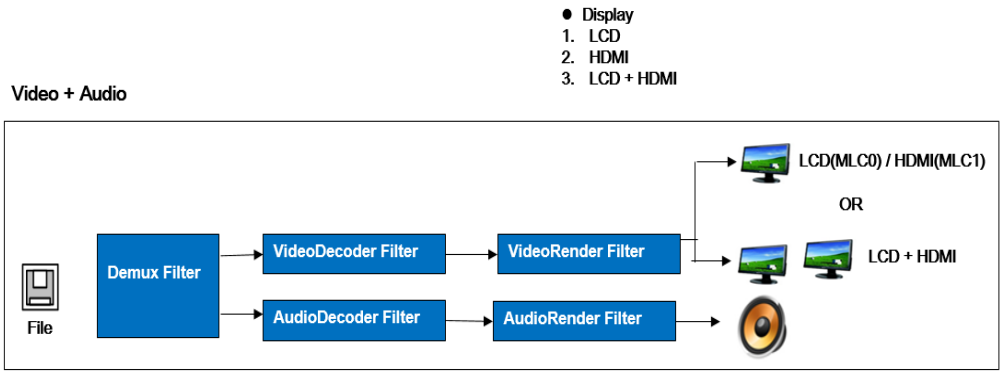


5.1.2.2 Video + Audio

MLC 를 사용하는 경우는 NX_MPAddTrack() 함수 인자를 MP_DSP_CONFIG *pInfo 을 설정한후 전달해야 한다.

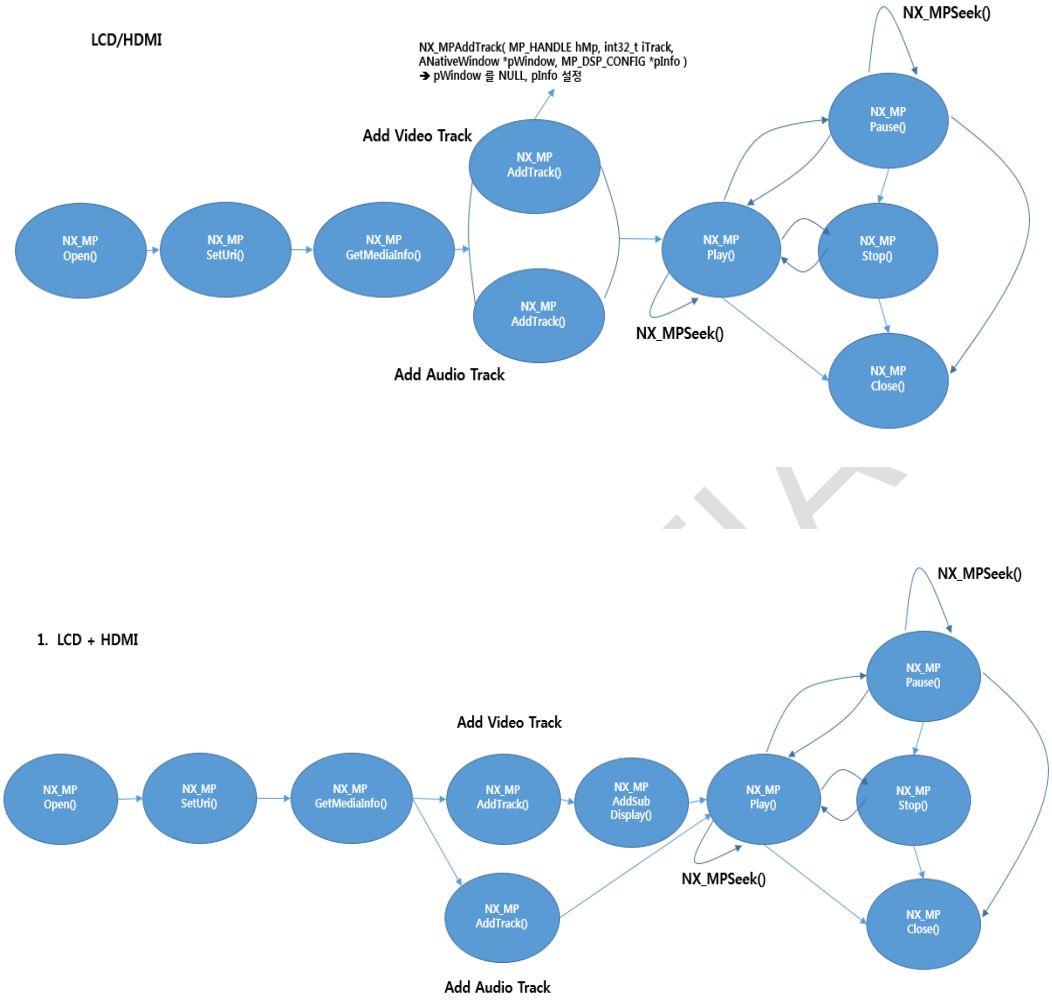
아래그림은 MLC 를 사용해서 Display 할때 사용하는 함수 호출 순서이다.

DualDisplay(LCD + HDMI) 인 경우는 NX_MPAddSubDisplay() 함수를 사용한다.



CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.



5.2 Video Only, Audio Only, Video + Audio (Linux)

5.2.1 Video MLC

Linux 에서 Display 는 MLC 를 사용한다.

아래는 Linux 에 관련된 Scenario 이다.

Linux 에서는 Volume Control 은 NX_MPSetVolume()함수를 통해서 제어한다.

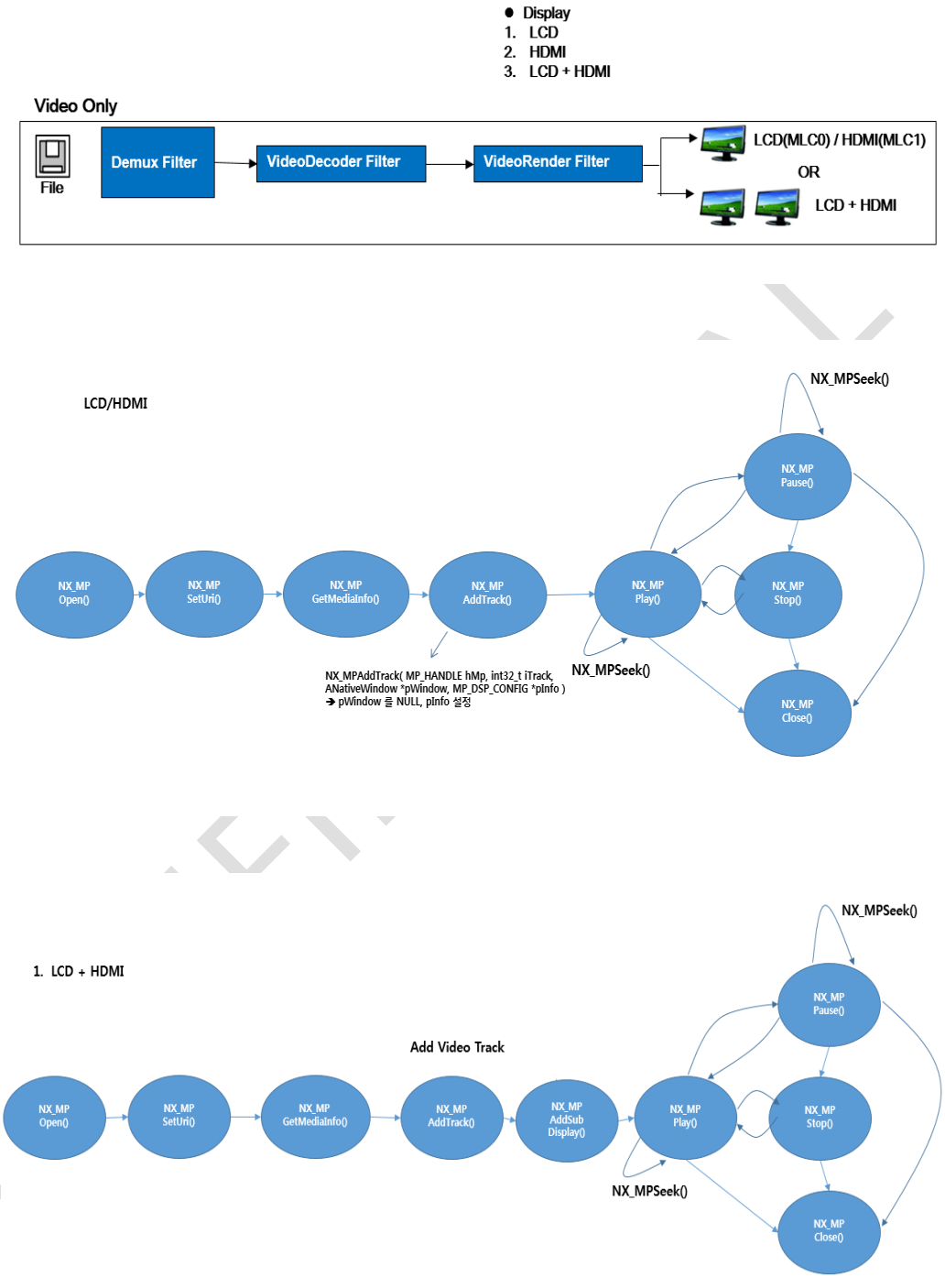
5.2.1.1 Video Only

NX_MPAddTrack() 함수 인자를 MP_DSP_CONFIG *pInfo 을 설정한후 전달해야 한다.

아래그림은 Video Only 일때 Display 하기위한 사용하는 함수 호출 순서이다.

DualDisplay(LCD + HDMI) 인 경우는 NX_MPAddSubDisplay() 함수를 사용한다.

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.



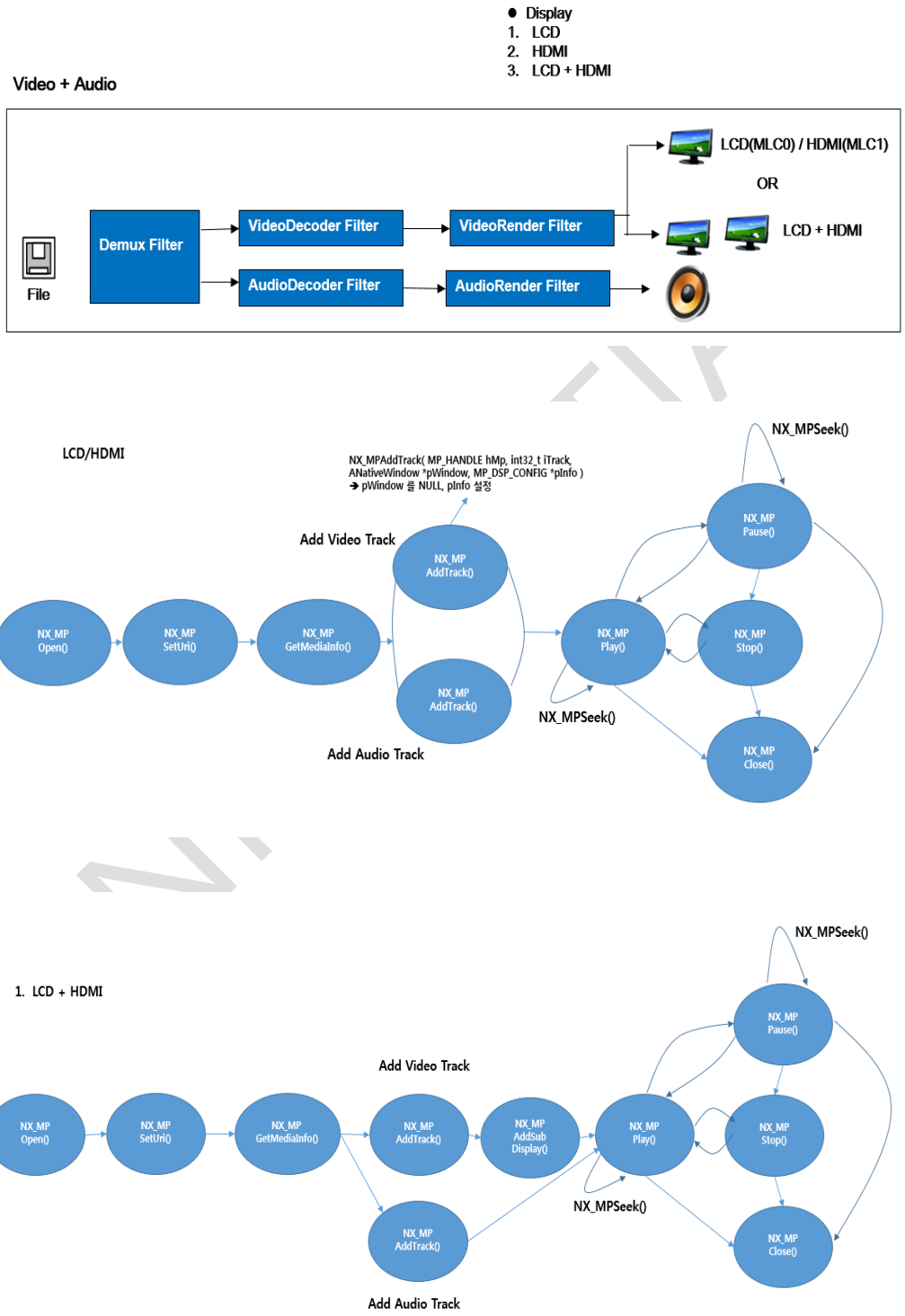
5.2.1.2 Video + Audio

NX_MPAddTrack() 함수 인자를 MP_DSP_CONFIG *pInfo 을 설정한후 전달해야 한다.

아래그림은 MLC 를 사용해서 Display 할때 사용하는 함수 호출 순서이다.

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

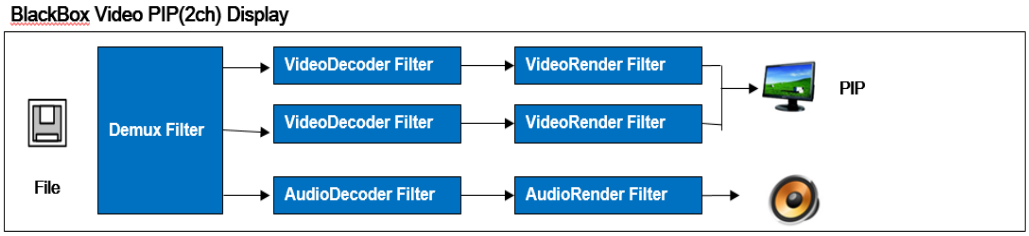
DualDisplay(LCD + HDMI) 인 경우는 NX_MPAddSubDisplay() 함수를 사용한다.

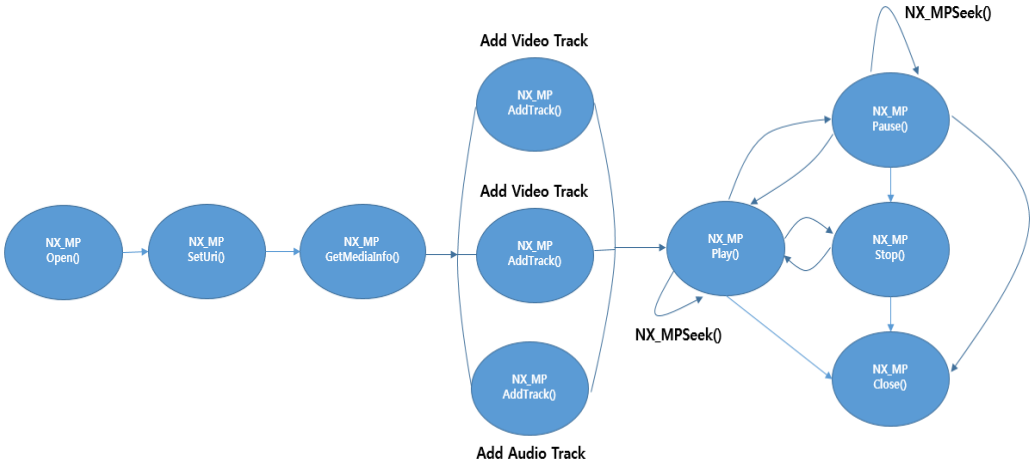


Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

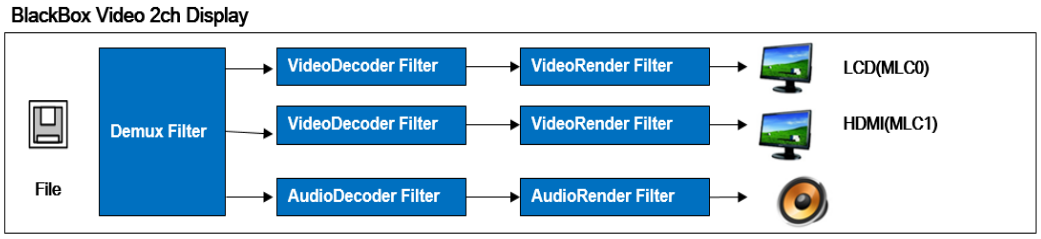
5.3 BlackBox Video PIP(2ch) Display (Android)

BlackBox 인 경우는 Nexell 에서 제공한 BlackBox 를 사용해서 인코딩한것만 PIP 를 제공한다.





5.4 BlackBox Video 2ch Display (Linux)

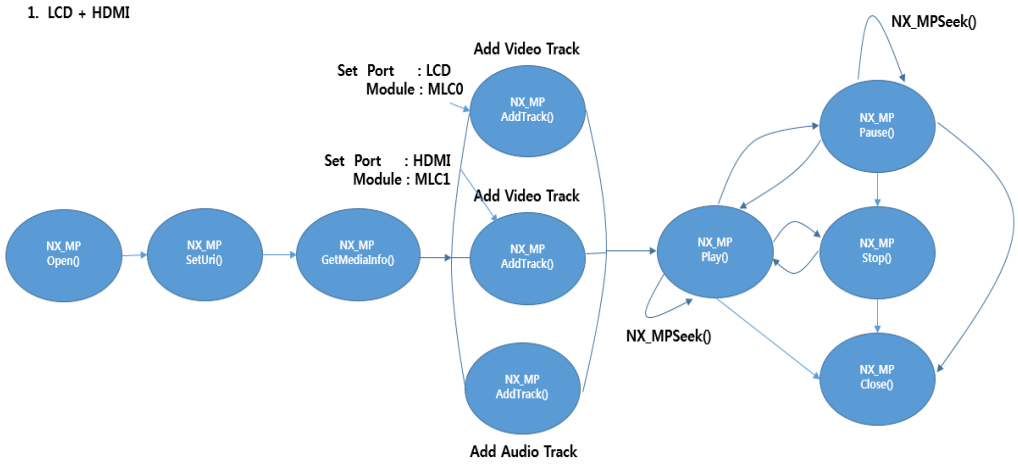


아래 그림은 BlackBox 에서 2ch Display 를 실행하기 위한 함수 호출순서이다.

1. 1ch Video Track 은 LCD, NX_MPAddTrack() 에서 Port: LCD, Module: MLC0 을 설정한다.

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.

2. 2ch Video Track 은 HDMI., NX_MPAddTrack() 에서 Port: HDMI, Module: MLC1 을 설정한다
3. Audio Track 을 NX_MPAddTrack() 사용해서 Audio Track 을 추가한다..
4. Play 를 실행한다.



CONFID

Chap 6. Known Issues

6.1 To Do List

- HEVC S/W Codec 지원.

6.2 Known Issues

None

CONFIDENTIAL

Copyright © 2018 Nexell Co.,Ltd. All rights reserved.