서식 있음: 들여쓰기: 왼쪽 0 글자

Demo Application UI Guide (NxQuickRearCam)

Version 01.60.0

Display Audio

Solution Team

Release information

The following changes have been make to this document.

Change History

Date	Change
28 Feb. 2019~~04 Dec. 2017~~	First release for v1.0.0~~First release for v0.6.0~~

서식 있는 표

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.

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

Contents

Chap 1. Overview 1

1.1 Overview 1

1.2 Block Diagram 1

1.3 Application UI 2

1.4 Configuration File 32

Chap 2. Camera Library 65

2.1 Overview 65

2.2 APIs 65

Chap 3. History 1310

3.1 Known Issue 1310

3.2 To do list 1310

Chap 1. Overview 1

1.1 Overview 1

1.2 Block Diagram 1

1.3 Application UI 1

Chap 2. Camera Library 3
2.1 Overview 3
2.2 APIs 3

Chap 3. History 6
3.1 Known Issue 6
3.2 To do list 6

Chap 1. Overview

1.1 Overview

이 문서는 Display Audio 의 Demo Application 인 NxQuickRearCam 에 대해서 설명한 문서이다. NxQuickRearCam 은 Kernel Layer 의 NxQuickRearCam Camera application 의 동작을 이어받아서 동작하도록 되어있다.

Demo Application 에서는 차량용 Rear Camera System 을 modeling 하기 위하여 GPIO pin 한 개를 후방 기어로 modeling 하였으며, 이를 통하여 application 이 camera 영상은 display 하도록 되어 있으며 software deinterlace engine 을 실행되도록 구성되어있다.포함하고 있다.

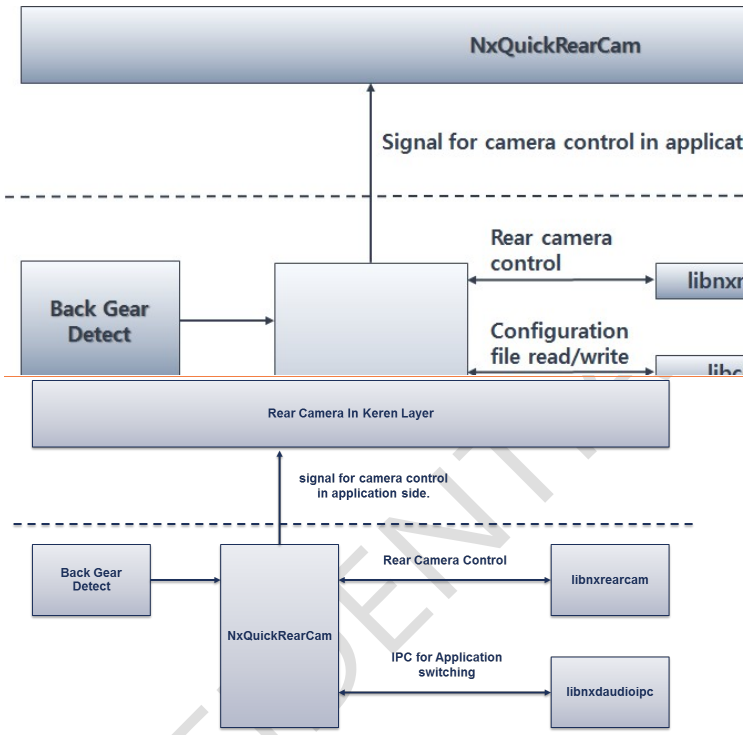
* NxQuickRearCam : booting 완료(NxLauncher 실행을 포함) 전, backgear 와 연동되는 camera service 를 제공하기 위한 application.

서식 있음: 양쪽

1.2 Block Diagram

NxQuickRearCam 은 아래와 같이 구성되어있다. Application 시작시에 kernel layer 의 CameraQuickRearCam Application 으로부터 제어권을 얻어 오기 위한 signal 을 발생시킨다. 이로부터 NxRearCamapplication 은 자체적으로 back gear 를 detection 하며 NxQuickRearCamcamera 영상을 display 하도록 구성되어 있다.아 실행되도록 구성되어있다.

서식 있음: 양쪽

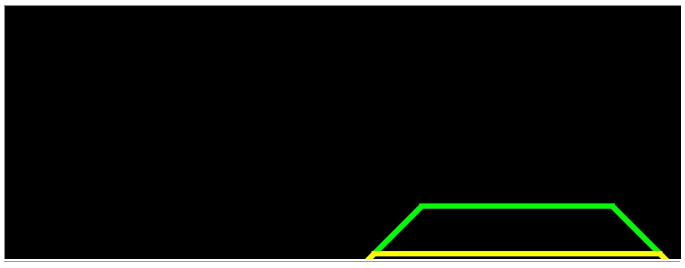


서식 있음: 가운데

1.3 Application UI

Application 화면은 다음과 같으며 화면상에 Camera 영상과 Parking Guideline 이 같이 주사된다.

서식 있음: 양쪽



서식 있음: 가운데

서식 있음: 왼쪽

1.4 Configuration File

서식 있음: 양쪽

Configuration file(rearcam_config.xml)은 camera 및 display 를 위한 configuration 값들이 포함되어 있다. Configuration file 은 "/nexell/daudio/NxRearCam/"에 위치한다 파일이 존재하지 않을 경우, application 은 default value 들로 실행된다.

Configuration file 의 형식은 다음과 같다.

[rearcam_config.xml]

```
<?xml version="1.0" encoding="UTF-8"?>
<map>
    <string name="module">1</string>
    <string name="use_intercam">1</string>
    <string name="cam_width">704</string>
    <string name="cam_height">480</string>
    <string name="video_layer_idx">0</string>
    <string name="crtc_idx">0</string>
    <string name="cam_display_x">0</string>
    <string name="cam_display_y">0</string>
    <string name="cam_display_width">1024</string>
    <string name="cam_display_height">600</string>
    <string name="deinterlace_engine">1</string>
    <string name="deinter_param">3</string>
    <string name="lcd_width">1024</string>
    <string name="lcd_height">600</string>
    <string name="pgl_enable">1</string>
    <string name="backgear_enable">1</string>
    <string name="gpioIdx">163</string>
    <string name="quick_running">1</string>
```

```
</map>
```

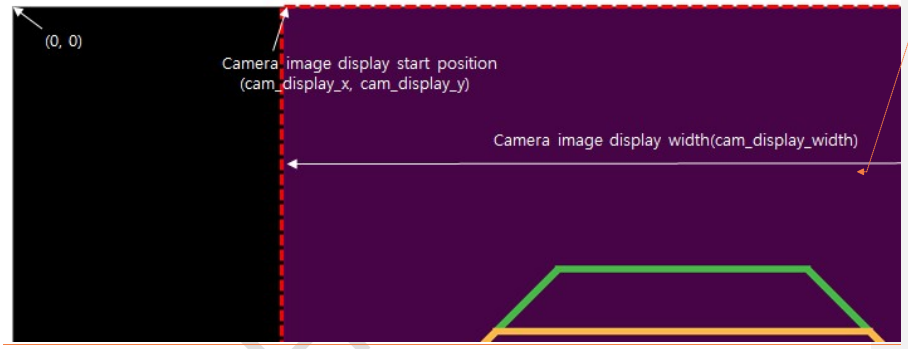
Configuration	Description
module	Camera senor module number
use_intercam	Interlace camera 사용 여부 0 : not use interlace camera 1 : use interlace camera
cam_width	Camera image width
cam_height	Camera image height
video_layer_idx	Camera image가 rendering될 layer index. 본 application에서는 video layer에 rendering되므로 0으로 setting되어야 함
crtc_idx	몇 번째 crtc를 사용할 것인지에 대한 index로 하나의 display device만 지원되면 0으로 setting되어야 하며 두 개 이상의 display device를 사용할 경우 선택적으로 원하는 display device의 index를 setting
cam_display_x	Camera image를 display할 위치의 시작 x position
cam_display_y	Camera image를 display할 위치의 시작 y position
cam_display_width	Camera image display width
cam_display_height	Camera image display height
deinterlace_engine	Interlace camera를 사용할 경우 적용할 deinterlace engine 0: none 1: Nexell deinterlace engine 2. TS deinterlace engine
deinter_param	Motion detect sensitivity. TS deinterlace engine을 사용할 경우 적용되는 parameter
lcd_width	display device의 display width
lcd_height	display device의 display height
pgl_enable	Parking guide line을 drawing할 것인지 결정 0 : parking guide line을 그리지 않음 1 : parking guide line을 그림
backgear_enable	Backgear 연동을 할 것인지 결정

	0 : backgear 연동하지 않음 1 : backgear 연동 함
gpioIdx	Backgear와 mapping되어 있는 gpio의 number
quick_running	QuickRearCam의 사용 여부 0 : QuickRearCam을 사용하지 않음 1 : QuickRearCam을 사용함

서식 있음

아래는 display position 및 size 와 관련된 configuration parameter 들을 설명하기 위한 그림이다.

서식 있음: 들여쓰기: 왼쪽 2.13 글자



Camera image 가 display 되는 region 은 display device 가 지원하는 full 영역을 사용할 수도 있고, camgera image 의 aspact ratio 를 고려하여 camera image 가 display 되는 region 을 설정할 수도 있다.

서식 있음: 들여쓰기: 왼쪽 5.67 글자

Chap 2. Camera Library

2.1 Overview

Camera 의 동작을 제어하기 위한 library libnxrearcam.so 가아 제공되며, 사용방법은 API 들은 아래와 같다.

2.2 APIs

2.2.1 NX_RegisterBackGearEventCallBack()

`void NX_RegisterBackGearEventCallBack(void *pAppData, void (*callback)( int32_t))`
Description Register callback function for action when backgear is detected.
Parameter void *pAppData : pointer of app data void (*callback)(int32_t) : callback function
Return Value None

서식 있음: 글꼴: 굵게 없음

서식 있음: 글꼴: 굵게 없음

서식 있음: 표준

2.2.2 NX_StartBackGearDetectService ()

`int32_t NX_StartBackGearDetectService( int32_t nGpio, int32_t nChkDelay )`
Description Start backgear detection service.
Parameter int32_t nGpio : index of gpio for backgear int32_t nChkDelay : interval time for checking gpio status
Return Value Zero is successful, -1 is failed

서식 있음: 표준

2.2.3 NX_StopBackGearDetectService()

`void NX_StopBackGearDetectService()`
Description Stop backgear detection service

Parameter
None
Return Value
None

~~2.2~~

서식 있음: 표준

2.2.4 NX_RearCamInit()

int32_t NX_QuickRearCamInit (NX_REARCAM_INFO *p_VipInfo, DISPLAY_INFO* p_dspInfo, DEINTERLACE_INFO *p_deinterInfo)
Description
Initialization
Parameter
NX_REARCAM_INFO* p_VipInfo : configuration for camera DISPLAY_INFO* p_dspInfo : configuration for display DEINTERLACE_INFO* p_deinterInfo : configuration for deinterlace
Return Value
Zero is successful, -1 is failed.

- NX_REARCAM_INFO

```
typedef struct _NX_REARCAM_INFO{
        int32_t iType;              //camera type : CAM_TYPE_VIP
        int32_t iModule;            //camera module index
        int32_t iSensor;            //sensor
        int32_t iClipper;           //clipper
        int32_t bUseMipi;           //using mipi
        int32_t bUseInterCam;       //using interlace camera
        int32_t iFpsNum;            //frame per sec
        int32_t iFpsDen;            //denominate value of fps
        int32_t iNumPlane;          //number of plane
        int32_t iWidth;             //camera input width
        int32_t iHeight;            //camera input height
        int32_t iCropX;             //crop x position
        int32_t iCropY;             //crop y position
        int32_t iCropWidth;         //crop width
        int32_t iCropHeight;        //crop height
        int32_t iOutWidth;          //decimator width
        int32_t iOutHeight;         //decimator height
} NX_REARCAM_INFO;
```

- DISPLAY_INFO

```
typedef struct tagDISPLAY_INFO{
        uint32_t iConnectorIdx;     //drm connector index
```

```
    int32_t iPlaneIdx;          //drm plane index
    int32_t iCrtcIdx;           //drm crtc indexc
    uint32_t uDrmFormat;        //drm data format
    int32_t iSrcWidth;          //width of input image
    int32_t iSrcHeight;         //height of input image
    int32_t iCropX;             //crop x position
    int32_t iCropY;             //crop y position
    int32_t iCropWidth;         //crop width
    int32_t iCropHeight;        //crop height
    int32_t iDspX;              //display position
    int32_t iDspY;              //crop start x position
    int32_t iDspWidth;          //crop start y position
    int32_t iDdspHeight;        //crop width
    int32_t iCropHeight;        //crop height
    int32_t iPlaneId_PGL;       //plane ID for drawing parking guide line
    int32_t uDrmFormat_PGL; //data format for drawing parking guide line
    void*       m_pNativeWindow;    //only for surface view rendering at android application
} DISPLAY_INFO;
```

- DEINTERLACE_INFO

```
typedef struct tagDEINTERLACE_INFO{
    int32_t iWidth;             //width of input image
    int32_t iHeight;            //height of input image
    int32_t iEngineSel;         //deinterlace engine – 0 : none 1:nexell deinterlace 2: Thunder soft deinterlace
    int32_t iCorr;              // correlation value of motion detection sensitivity for Thunder soft deinterlace
} DEINTERLACE_INFO;
```

~~int32_t NXDA_ShowRearCam(~~
~~CAMERA_INFO *pCamInfo,~~
~~DISPLAY_INFO *pDspInfo~~
~~);~~

~~Description~~

~~Show Rear Camera.~~

~~Parameter~~

~~- pCamInfo : Camera Information~~
~~- pDspInfo : Display Information~~

~~Return Value~~

~~Please describe return value.~~

서식 있음: 들여쓰기: 첫 줄: 4 글자

2.2.5 NX_RearCamDeInit ()

서식 있음: 다단계 번호 매기기 + 수준:3 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0 cm + 들여쓰기 위치: 2.5 cm

```
int32_t NX_RearCamDeInit()
```

Description

Rear cam deinit
Parameter
None
Return Value
Zero is successful, -1 is failed.

서식 있음: 표준, 글머리 기호 또는 번호 없이

2.2.6 NX_RearCamStart()

int32_t NX_QuickRearCamStart()
Description
Start rendering rear camera images.
Parameter
None
Return Value
Zero is successful, -1 is failed.

~~2.2.1 NXDA_HideRearCam()~~

~~void NXDA_HideRearCam(~~ ~~void~~ ~~);~~
~~**Description**~~
~~Hide Rear Camera.~~
~~**Parameter**~~
~~None.~~
~~**Return Value**~~
~~None~~

2.2.7 NX_RearCamGetStatus

int32_t NX_QuickRearCamGetStatus()
Description
Get status
Parameter
None
Return Value
0 : stop 1 : init 2 : running

~~2.2.2 NXDA_RegRenderCallback()~~

~~void NXDA_RegRenderCallback(~~

~~void *pApp~~
~~int32_t (callback)(void *, int32_t, void*, int32)~~
~~);~~

~~Description~~

~~Register Rear Camera render callback.~~

~~Parameter~~

~~. pApp : private handle.~~
~~. callback : redering callback.~~
~~int32_t callback(void* pApp, int32_t type, void* data, int32_t dataSize)~~
~~-. pApp : private handle.~~
~~. type : callback function type. (CB_TYPE_BUFFER, CB_TYPE_HIDE, CB_TYPE_SHOW)~~
~~. data : send data for callback.~~
~~-. dataSize : size of data~~

~~Return Value~~

~~None~~

2.2.8 NX_RearCamGetVersion

```
int32_t NX_QuickRearCamGetVersion()
```

Description

Get NxQuickRearCam version information

Parameter

None

Return Value

Version information

Major : ((return value) & 0xFF000000) >> 24
Minor : ((return value) & 0x00FF0000) >> 16
Revision : ((return value) & 0x0000FF00) >> 8
Reservation : ((return value) & 0x000000FF)

~~2.2.3 NXDA_RegControlCallback()~~

```
void NXDA_RegControlCallback(
        void *pApp,
        int32_t (callback)(void *, int32_t, void *, int32_t)
);
```

~~Description~~

~~Register Rear Camera control callback.~~

~~Parameter~~

~~. pApp : private handle.~~
~~. callback : redering callback.~~
~~int32_t callback(void* pApp, int32_t type, void* data, int32_t dataSize)~~
~~-. pApp : private handle.~~
~~-. type : callback function type. (CB_TYPE_BUFFER, CB_TYPE_HIDE, CB_TYPE_SHOW)~~

- data : send data for callback. - dataSize : size of data
Return Value None

2.2.4

서식 있음

2.2.52.2.9 NX_RearCamSetDisplayPosition NXDA_StartBackGearDetectService()

int32_t NXDA_StartBackGearDetectService(int32_t nGpiox, int32_t nChkDelayy, int32_t w, int32_t h);
Description Set display position of camera images Start back gear detection service.
Parameter - nGpiox : : GPIO port numberstart x-position. - nChkDelayy : start y-position - w : display width. - h : display heightGPIO check delay (mSec)
Return Value Zero is returned.

서식 있음: 들여쓰기: 첫 줄: 0.5 글자

2.2.6 NXDA_StopBackGearDetectService()

void NXDA_StopBackGearDetectService(void);
Description Start back gear detection service.
Parameter None.
Return Value None.

2.2.7 NXDA_RegisterBackGearEventCallback()

void NXDA_RegisterBackGearEventCallback(void *pAppData, void (*callback)(void *pAppData, int32_t nOnOff)

); **Description** Please describe this function.
Parameter - pAppData : private handle. - callback : register back gear detection callback. void (*callback)(void *pAppData, int32_t nOnOff) - pAppData : private handle. - nOnOff : back gear status.
Return Value None.

Chap 3. History

3.1 Known Issue

-. Not yet.

3.2 To do list

-. ~~Audio Focus 전환 시나리오 적용.~~