스타일 정의: 목차 2

Demo Application UI Guide (Nx~~Quick~~RearCam)

Version ~~0~~1.~~6~~0.0

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

변경된 필드 코드

서식 있음: 글꼴: 굵게

Chap 1. Overview

1.1 Overview

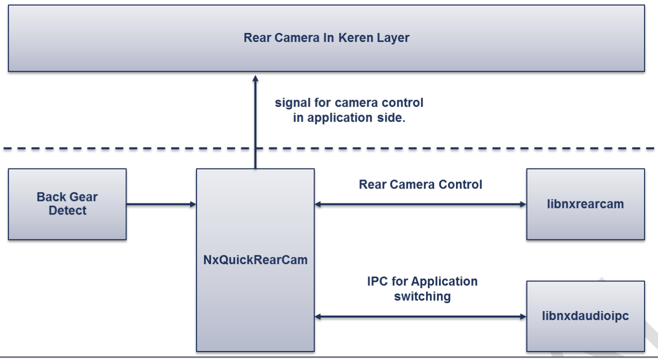
This document describes Nx~~Quick~~RearCam that is demo application for Display Audio. The Nx~~Quick~~RearCam is operated after taking over camera application of~~ kernel layer~~ NxQuickRearCam. In demo application, the GPIO pin is back-gear. The application is run by GPIO pin. And this application includes software deinterlace engine.

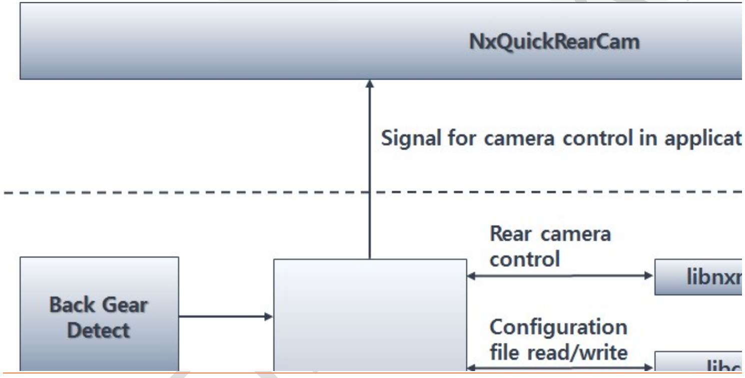
서식 있음: 양쪽

* NxQuickRearCam : the application for camera service interlocked a backgear before completion of booting(include execution of NxLauncher).

1.2 Block Diagram

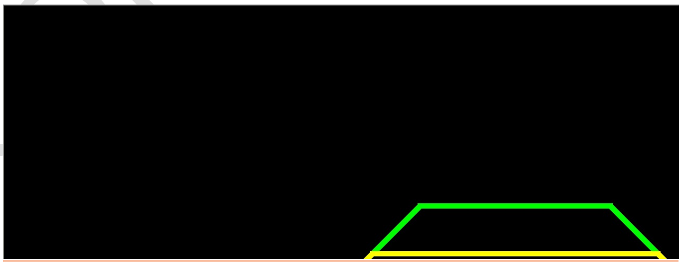
The Nx~~Quick~~RearCam structure see as below. The application send signal to NxQuickRearCam~~kernel~~ to ~~for ~~obtain~~–~~ control of GPIO pin when it is started. The application is run automatically by back-gear detection.

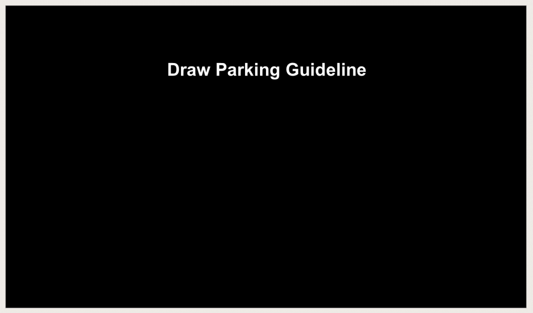

Rear Camera In Keren Layer
signal for camera control in application side.
Back Gear Detect
NxQuickRearCam
Rear Camera Control
libnxrearcam
IPC for Application switching
libnxdaudioipc


NxQuickRearCam
Signal for camera control in applicat
Back Gear Detect
Rear camera control
libnxr
Configuration file read/write
libc

1.3 Application UI

The Application see as below, this is displayed camera screen and parking guideline.





1.4 Configuration File

The configuration file (rearcam_config.xml) includes configuration values for display. It is in the folder "/nexell/daudio/NxRearCam". If the file does not exist, the application is run by default values The format is as in the following.

서식 있음: 양쪽

[rearcam _config.xml]

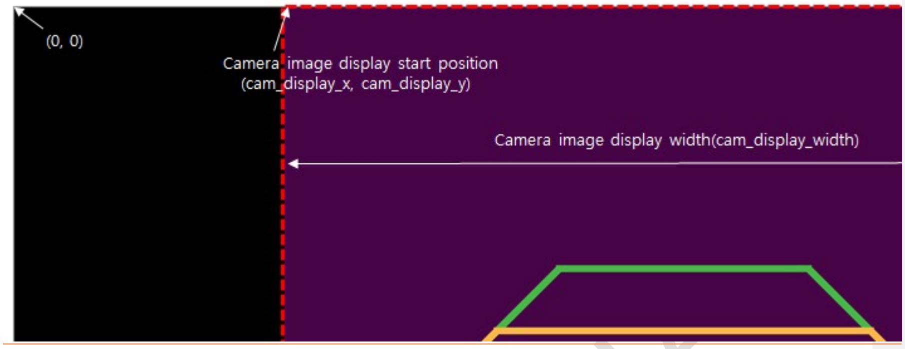
```
<?xml version="1.0" encoding="UTF-8"?>
<map>
    <string name="module">1</string>
    <string name="use_intercam">1</string>
    <string name="cam_width">704</string>
    <string name="cam_height">480</string>
    <string name="video_layer_idx">0</string>
    <string name="crtc_idx">0</string>
    <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	Using Interlace camera 0 : not use interlace camera 1 : use interlace camera
cam_width	camera image width
cam_height	camera image height
video_layer_idx	Layer(plane) index for rendering of camera images. At this application, camera images is rendering at video layer, so this parameter has to be set "0". It means that camera images will be rendering at 1st video layer.
crtc_idx	crtc index for rendering of camera images. If it is set "0", it means using 1st crtc. If display devices more than two are used, it has to be set index of proper crtc.
cam_display_x	start x-position for displaying of camera images
cam_display_y	start y-position for displaying of camera images
cam_display_width	display width of camera images
cam_display_height	display height camera images
deinterlace_engine	deinterlace engine selection This parameter is needed in case of using interlace camera. 0: none 1: Nexell deinterlace engine 2. TS deinterlace engine
deinter_param	Motion detect sensitivity. This parameter is used only for TS deinterlace engine.
lcd_width	display width of display device
lcd_height	display height of display device
pgl_enable	Enable/disable drawing Parking guide line 0 : drawing parking guide line 1 : not drawing parking guide line
backgear_enable	Enable/disable backgear detection 0 : enable backgear detection 1 : disable backgear disable
gpioIdx	Index of GPIO that is mapped as backgear
quick_running	Running NxQuickRearCam 0 : running NxQuickRearCam 1 : not running NxQuickRearCam

서식 있음: 위 첨자

서식 있음: 위 첨자

The figure below shows configuration parameters for display position and size.


(0, 0)
Camera image display start position
(cam_display_x, cam_display_y)
Camera image display width(cam_display_width)

The region for display camera images can be a region considering aspect ratio of camera images as well as full region of display device.

Chap 2. Camera Library

2.1 Overview

The library ~~–~~libnxrearcam.so is ~~–~~provided~~es~~ to manage camera. ~~This usage~~APIs of library see as below.

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

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

2.2.3 NX_StopBackGearDetectService()

`void NX_StopBackGearDetectService()`
Description Stop backgear detection service
Parameter None

Return Value

None

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;
```

2.2.5 NX_RearCamDeInit ()

`int32_t NX_RearCamDeInit()`
Description Rear cam deinit
Parameter None
Return Value Zero is successful, -1 is failed.

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.

2.2.7 NX_RearCamGetStatus

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

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)

2.2.9 NX_RearCamSetDisplayPosition ()

```
int32_t NXDA_StartBackGearDetectService(
          int32_t x,
          int32_t y,
          int32_t w,
          int32_t h
);
```

Description Set display position of camera images
Parameter - x : start x-position. - y : start y-position - w : display width. - h : display height

Return Value

Zero is returned.

~~2.2.1 NXDA_ShowRearCam()~~

```
~~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.~~

~~2.2.2 NXDA_HideRearCam()~~

```
~~void NXDA_HideRearCam(~~
~~        void~~
~~);~~
```

~~**Description**~~

~~Hide Rear Camera.~~

~~**Parameter**~~

~~None.~~

~~**Return Value**~~

~~None~~

~~2.2.3 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~~

~~2.2.4 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~~

~~2.2.5 NXDA_StartBackGearDetectService()~~

~~int32_t NXDA_StartBackGearDetectService(~~

~~int32_t nGpio,~~

~~int32_t nChkDelay~~

~~);~~

~~**Description**~~

~~Start back gear detection service.~~

~~**Parameter**~~

~~. nGpio : GPIO port number.~~

~~. nChkDelay : GPIO check delay (mSec)~~

~~**Return Value**~~

~~Zero is returned.~~

~~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

-. ~~Apply to change audio focus scenario.~~