Demo Application UI Guide (QuickRearCam)

Version 0.7.0

Display Audio

Solution Team

Release information

The following changes have been make to this document.

Change History

Date	Change
15 Nov. 2018	Applying single application framework and Using configuration file for display for v0.7.0
04 Dec. 2017	First release for v0.6.0

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.

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 1
1.4 Configuration File 2

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

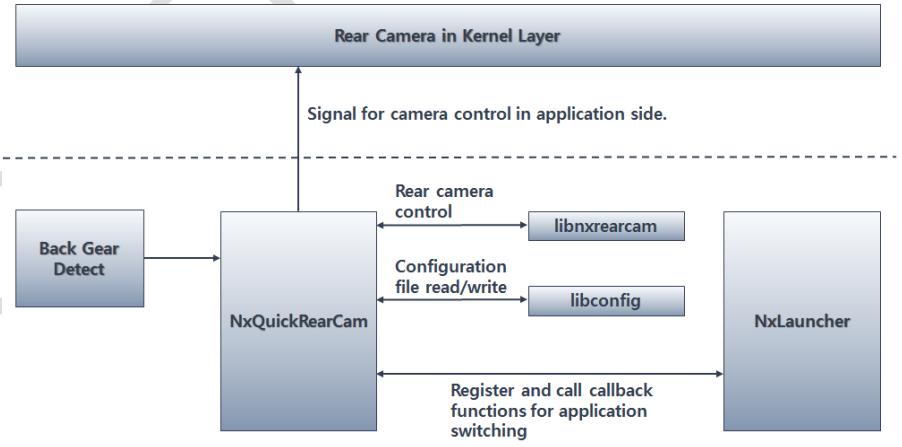
1.1 Overview

이 문서는 Display Audio 의 Demo Application 인 NxQuickRearCam 에 대해서 설명한 문서이다. NxQuickRearCam 은 Kernel Layer 의 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 실행되도록 구성되어있다.

1.2 Block Diagram

NxQuickRearCam 은 아래와 같이 구성되어있다. Application 시작시에 kernel layer 의 Camera Application 으로부터 제어권을 얻어 오기 위한 signal 을 발생시킨다. 이로부터 application 은 자체적으로 back gear 를 detection 하며 NxQuickRearCam 이 실행되도록 구성되어있다.



1.3 Application UI

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

1.4 Configuration File

Configuration file(config.xml)은 display 를 위한 configuration 값들이 포함되어 있다. Configuration file 은 "/nexell/daudio/NxQuickRearCam/"에 위치한다 파일이 존재하지 않을 경우, application 은 default value 들로 실행되며 application 이 종료되면 default 값으로 setting 된 configuration file 이 생성된다. Configuration 값은 display 를 위한 crtc index 와 layer index 를 갖는다. 첫 번째 crtc 의 video layer 에 display 를 원할 경우 crtc index 는 "0", layer index 는 "0"으로 setting 하면 된다.

Configuration file 의 형식은 다음과 같다.

[config.xml]

```
<?xml version="1.0" encoding="UTF-8"?>
<map>
    <string name="ctrc_idx">0</string>
    <string name="layer_idx">0</string>
</map>
```

Chap 2. Camera Library

2.1 Overview

Camera 의 동작을 제어하기 위한 libnxrearcam 이 제공되며, 사용방법은 아래와 같다.

2.2 APIs

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

-. Audio Focus 전환 시나리오 적용.