Porting Guide (NxQuickRearCam)

Version 1.0.0

Display Audio

Solution Team

Release information

The following changes have been make to this document.

Change History

Date	Change
28 Feb. 2019	First release for v1.0.0

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

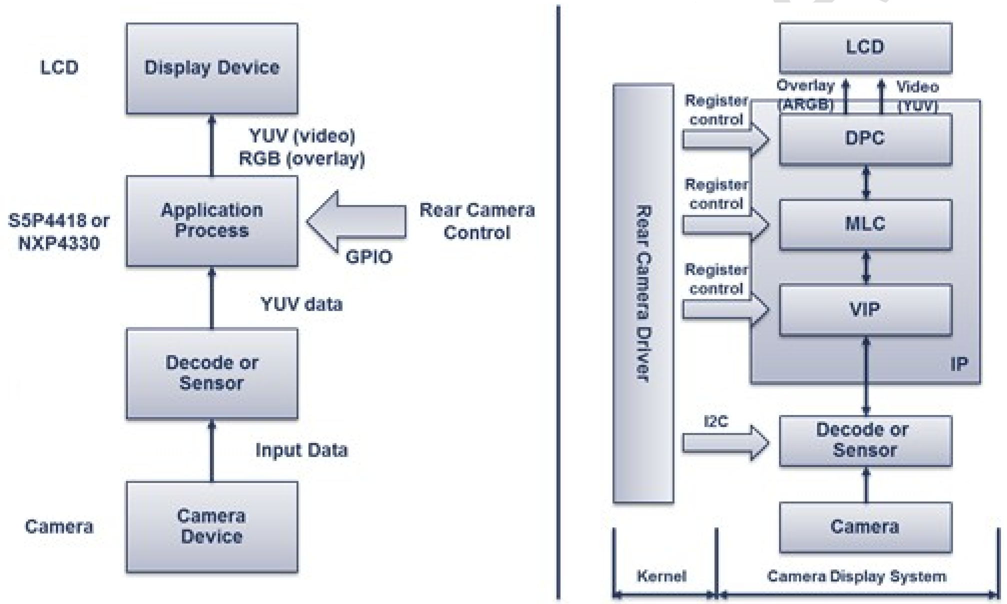
Contents

Chap 1. Overview

1.1 OverView

NxQuickRearCam can rendering the image received via the external decoder or sensor to the LCD according to the state of the GPIO (high or low) that controls whether the camera is operating or not. .

It also includes the software deinterlace engine and drawing parking guide line.



Hardware Structure Software Structure

Chap 2. Simple Guide

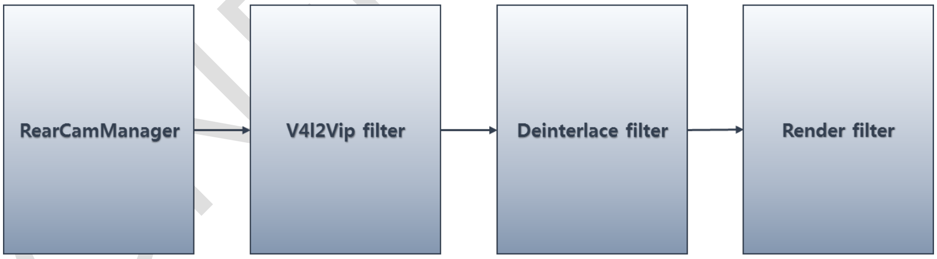
2.1 Directory Structure

The directory structure of released solution.

```
├── apps
│      └── NxQuickRearCam   : Console Application for Linux and android
├── include                 : Include Files of Libraries
├── lib
│      └── linux            : libraries for linux application
└── src
       ├── common           : source code for mutex and debug log
       └── librearcam       : source code for libnxrearcam
```

2.2 Structure

This solution consists of Manager and tree filters.



- RearCam Manager : NxQuickRearCam control

- V4l2Vip filter : getting video data from camera sensor

- Deinterlace filter : deinterlacing interlaced video data

- Render filter : displaying video data on screen

2.3 Build

This solution is compiled at static build environment for Quick Boot. build.sh script file is used for build.

# ./build_linux.sh -t	Build all libraries and application
# ./build_linux.sh -a	Build application
# ./build_linux.sh -r	Build libnxrearcam.a
# ./build_linux.sh -p	Build libnxv4l2.a and libnxdeinterlace.a
# ./build_linux.sh -c	clean

2.4 Execute

This application is executed with following options.

-m : module index

-i : using interlace camera

0 - not use interlace camera

1 - use interlace camera

-g : index of GPIO that is mapped as backgear

-b : backgear detection enable/disable

0 - disable

1 - enable

-c : crtc ID

-v : plane ID of video layer to display of camera images.

-p : plane ID of rgb layer for parking guide line

-r : resolution of images

-d : selection of deinterlace engine

0 : none

1: nexell deinterlace engine

2: 3rd party deinterlace engine

-l : debug log level

-s : motion sensitivity parameter for 3rd party deinterlace engine

-D : display start position(x,y) for camera images

-R : display size for camera images

-P : enable/disable drawing paring guide line

0 - disable

1 – enable

-L : screen size

[Example]

```
#NxQuickRearCam -m 1 -i 1 -g 43 -b 1 -c 26 -v 27 -r 960x480 -d 2 -s 3 -D 420,0 -R 1080x720 -L
1920x720
```

Chap 3. APIs

3.1 Overview

This part descripts APIs of NxQuickRearCam and an example of APIs use.

3.2 APIs

3.2.1 NX_RegisterBackGearEventCallBack()

`void NX_RegisterBackGearEventCallBack(void (*callback)( int32_t))`
Description Register callback function for action when backgear is detected.
Parameter void (*callback)(int32_t) : callback function
Return Value None

3.2.2 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

3.2.3 NX_StopBackGearDetectService()

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

3.2.4 NX_QuickRearCamInit()

```
int32_t NX_QuickRearCamInit
    (NX_REARCAM_INFO *p_VipInfo, DISPLAY_INFO* p_dspInfo, DEINTERLACE_INFO *p_deinterInfo)
```

Description

Initialize NxQuickRearCam

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 iConnectorId;      //drm connector ID
        int32_t iPlaneId;           //drm plane ID
        int32_t iCrtcId;            //drm crtc ID
        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
} 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;
```

3.2.5 NX_QuickRearCamStart()

`int32_t NX_QuickRearCamStart()`

Description

Start quick rear cam

Parameter

None

Return Value

Zero is successful, -1 is failed.

3.2.6 NX_QuickRearCamGetStatus

`int32_t NX_QuickRearCamGetStatus()`

Description

Get status

Parameter

None

Return Value

0 : stop

1 : init

2 : running

3.2.7 NX_QuickRearCamGetVersion

`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)

3.3 Simple example of APIs Use

```
static int32_t backgear_status;
static int32_t change_backgear_status;
static void cbBackGearStatus( int32_t iStatus )                    //backgear callback function
{
            backgear_status    = iStatus ? NX_BACKGEAR_NOTDETECTED : NX_BACKGEAR_DETECTED;

            change_backgear_status = 1;

            if(backgear_status)
                printf("[QuickRearCam] : BackGear Detected\n");
            else
                printf("[QuickRearCam] : BackGear Released\n");
}

int32_t main(void)
{
            int32_t iModule = 0;
            int32_t gpioIdx = 43;
            int32_t backgear_enable = 1;
            int32_t crtcId = 26;
            int32_t videoPlaneId = 27;
            int32_t rgbPlaneId = 18;
            int32_t cam_width = 960;
            int32_t cam_height = 576;
            int32_t deinter_engine = THUNDER_DEINTERLACER;
            int32_t dbg_level;
            int32_t corr = 3;

            NX_REARCAM_INFO vip_info;
            DISPLAY_INFO dsp_info;
            DEINTERLACE_INFO deinter_info;
```

```
memset( &vip_info, 0x00, sizeof(vip_info) );
memset( &dsp_info, 0x00, sizeof(dsp_info) );

vip_info.iType                          = CAM_TYPE_VIP;
vip_info.iModule                        = iModule;
vip_info.iSensor                        = nx_sensor_subdev;
vip_info.iClipper                       = nx_clipper_subdev;
vip_info.bUseMipi                       = false;
vip_info.bUseInterCam                   = true;
vip_info.iWidth                         = cam_width;
vip_info.iHeight                        = cam_height;
vip_info.iCropX                         = 0;
vip_info.iCropY                         = 0;
vip_info.iCropWidth                     = vip_info.iWidth;
vip_info.iCropHeight                    = vip_info.iHeight;
vip_info.iOutWidth                      = vip_info.iWidth;
vip_info.iOutHeight                     = vip_info.iHeight;

dsp_info.iPlaneId                       = videoPlaneId;
dsp_info.iCrtcId                        = crtcId;
dsp_info.uDrmFormat                     = DRM_FORMAT_YUV420;
dsp_info.iSrcWidth                      = vip_info.iWidth;
dsp_info.iSrcHeight                     = vip_info.iHeight;
dsp_info.iCropX                         = 0;
dsp_info.iCropY                         = 0;
dsp_info.iCropWidth                     = vip_info.iWidth;
dsp_info.iCropHeight                    = vip_info.iHeight;
dsp_info.iDspX                          = 0;
dsp_info.iDspY                          = 0;
dsp_info.iDspWidth                      = 1024;
dsp_info.iDspHeight                     = 600;

dsp_info.iPlaneId_PGL                   = rgbPlaneId;
dsp_info.uDrmFormat_PGL                 = DRM_FORMAT_ARGB8888;

deinter_info.iWidth                     = cam_width;
deinter_info.iHeight                    = cam_height;

if(vip_info.bUseInterCam == false)
{
        deinter_info.iEngineSel = NON_DEINTERLACER;
}else
{
        deinter_info.iEngineSel = deinter_engine;
}

deinter_info.iCorr                      = corr;

backgear_status = NX_BACKGEAR_NOTDETECTED;
change_backgear_status = 0;
```

```
        NX_RegisterBackGearEventCallBack( cbBackGearStatus );                  //Register backgear callback function

        NX_StartBackGearDetectService(gpioIdx, 100);                           //Backgear detecting service start

        if( 0 > NX_QuickRearCamInit( &vip_info, &dsp_info, &deinter_info ) )   //NxQuickRearCam Initialize
                return -1;

        while(1){
                if(change_backgear_status == 1)
                {
                        if(backgear_status == NX_BACKGEAR_DETECTED)
                        {
                                NX_QuickRearCamInit( &vip_info, &dsp_info, &deinter_info); //NxQuickRearCam Initialize
                                NX_QuickRearCamStart();      //NxQuickRearCam Start

                        }else
                        {
                                NX_QuickRearCamDeInit();     //NxQuickRearCam DeInit
                                usleep(100000);
                        }
                        change_backgear_status = 0;
                }
                usleep(1000);
        }
        NX_StopBackGearDetectService();             //Backgear detecting service stop

        NX_QuickRearCamDeInit();                    //NxQuickRearCam DeInit

        return 0;
}
```

Chap 4. Porting of NxQuickRearCam

4.1 Copy NxQuickRearCam binary file

Copy NxQuickRearCam binary file to yocto/meta-nexell/meta-nexell-distro/recipes-extended/nexell-init/files/nx_init

4.2 Modify nx_init.cpp file

Modify the line to yocto/meta-nexell/meta-nexell-distro/recipes-extended/nexell-init/files/nx_init/nx_init.cpp

For navi-ref board

```
[modify line]


execl("/sbin/NxQuickRearCam",  "NxQuickRearCam",  "-m1",  "-b1",  “-g163”,  "-c26",  “-
r704x480", “-D0,0”, “-R1024x600”, “-L1024x600”, NULL);
```

For convergence board

```
[modify line]


execl("/sbin/NxQuickRearCam",  "NxQuickRearCam",  "-m1",  "-b1",  “-g163”,  "-c26",  “-
r704x480", NULL);
```