AN13712

OS08A20 ISP Sensor on i.MX 8M Plus EVK

Rev. 3 — 9 October 2024 Application note

Document information

Information	Content
Keywords	AN13712, OS08A20 Sensor, i.MX 8M Plus EVK, i.MX 8M Plus Image Signal Processor (ISP), i.MX Yocto SDK
Abstract	This document describes the usage of the OS08A20 sensor on the i.MX 8M Plus EVK board. It provides detailed steps to enable the i.MX 8M ISP and camera on the board. The OS08A20 sensor supports image sizes of 4K, 2K, 1080p, and 720p.

1 Introduction

This document describes the steps for using the OS08A20 sensor on the i.MX 8M Plus ISP module. It describes the steps for building the NXP kernel and related files and enabling the ISP and camera module on the i.MX 8M Plus EVK board.

2 Module and board

This section describes the i.MX 8M Plus EVK board, OS08A20 sensor module, and their hardware connection.

2.1 i.MX 8M Plus EVK

Figure 1 shows the i.MX 8M Plus EVK board.

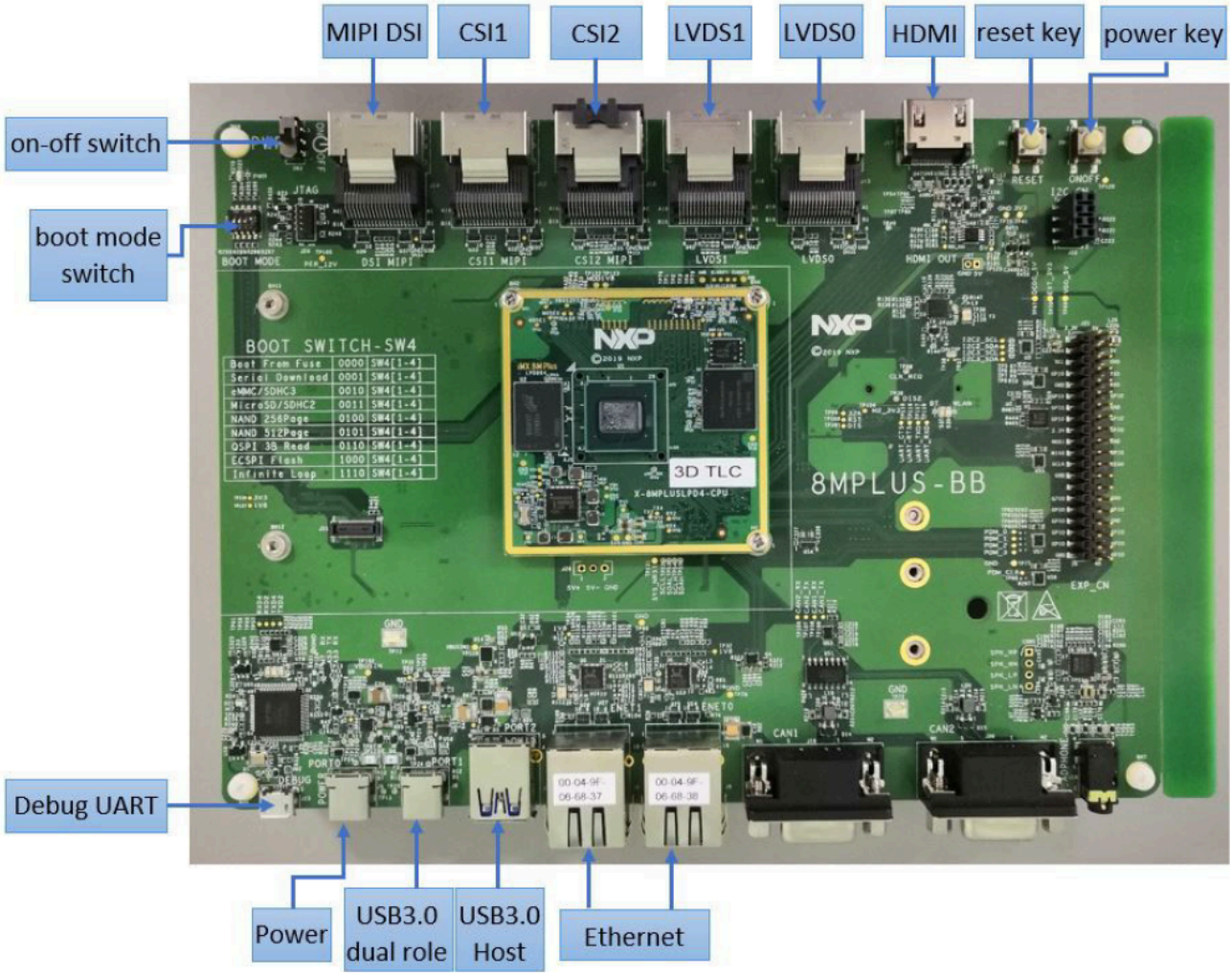

Figure 1. i.MX 8M Plus EVK board top picture

Note: *For more information about the i.MX 8M Plus EVK board, refer to the URL: Evaluation Kit for the i.MX 8M Plus Applications Processor.*

2.2 OS08A20 sensor module

Figure 2 shows the OS08A20 sensor module.

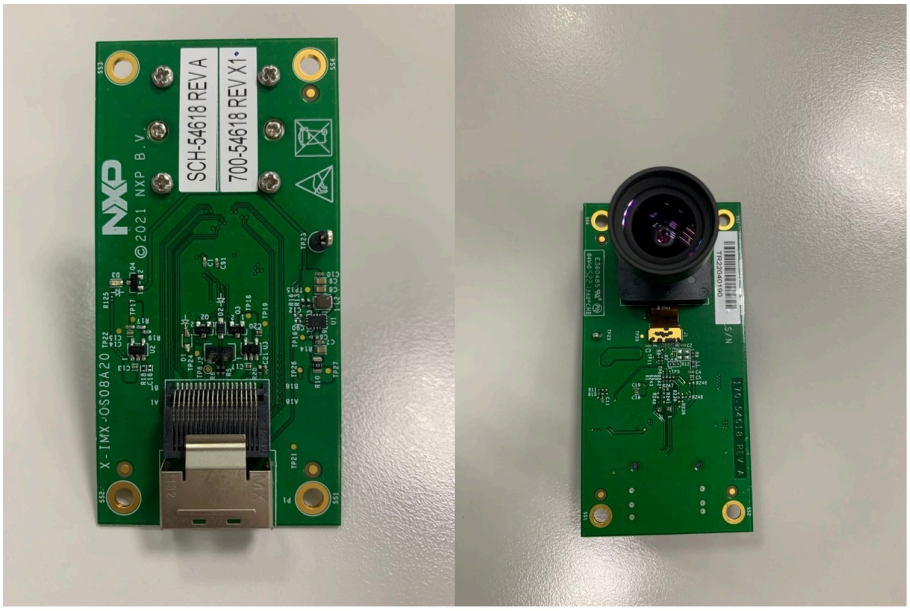

Figure 2. OS08A20 sensor module

The OS08A20 sensor supports image sizes of 4K, 2K, 1080p, and 720p. The output format is 12-bit / 10-bit RAW RGB. The sensor has 2-exposure staggered HDR and it supports the frame start input. The i.MX 8M Plus ISP has a demosaicing sensor (also known as color reconstruction sensor) for raw data processing and generating the YUV output format. The ISP also has the denoise, sharpen, and gamma modules to improve the sensor image quality.

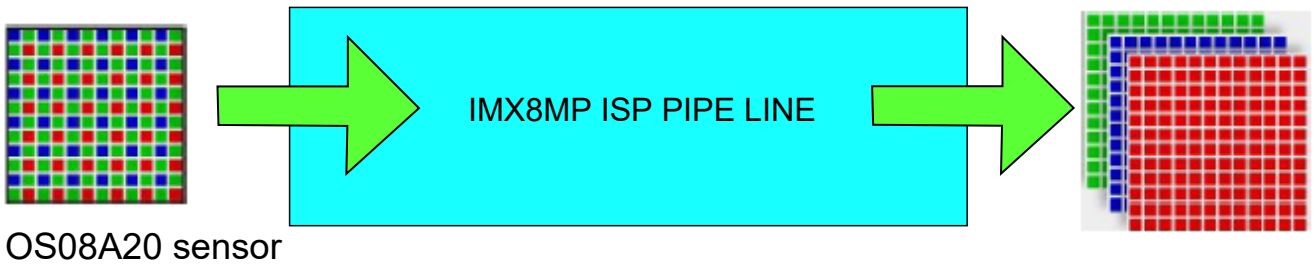

Figure 3. OS08A20 sensor

Following are the features of the OS08A20 sensor:

- Supports dual OS08A20 modules
- Support three ISP output formats:
  - YUV422
  - NV16
  - NV12
- Supports four sensor modes:
  - 1920x1080 10-bit linear mode
  - 1920x1080 10-bit HDR mode
  - 3840x2160 12-bit linear mode
  - 3840x2160 10-bit HDR mode

Note: *For more information about the IMX-OS08A20 sensor module, refer to the URL: https://www.nxp.com/part/IMX-OS08A20*

2.3 Hardware connection

The sensor is connected to the switch board. The switch board uses the MiniSas cable to connect to the CSI1 or CSI2 connectors on the I.MX 8M Plus EVK board.

For details about the hardware connection, refer to the *Application Note AN13736 "Using IMX-OS08A20 Camera Module on i.MX 8M Plus Evaluation Kit"*.

3 Configuring the software

This section describes how to configure the software for the sensor module.

3.1 OS08A20 SDK HAL source code

The OS08A20 SDK HAL source code is shown below:

```
├── OS08a20
│   ├── calib
│   │   ├── OS08a20_8M_10_1080p_linear.xml
│   │   ├── OS08a20_8M_10_1080p_hdr.xml
│   │   ├── OS08a20_8M_10_4k_linear.xml
│   │   └── OS08a20_8M_10_4k_hdr.xml
│   ├── source
│   │   └── OS08a20.c
│   ├── dewarp
│   │   ├── sensor_dwe_os08a20_1080P_config.json
│   │   └── sensor_dwe_os08a20_4K_config.json
```

3.2 OS08A20 kernel driver source code

The OS08A20 kernel driver source code is as follows:

```
├── os08a20
│   ├── os08a20_mipi_v3.c
│   ├── os08a20_regs_1080p.h
│   ├── os08a20_regs_1080p_hdr.h
│   ├── os08a20_regs_4k.h
│   └── os08a20_regs_4k_hdr.h
```

3.3 Sensor mode table

Table 1 describes the sensor modes.

Table 1. Sensor mode table

Mode	Index	Data format
1080P_linear	0	RAW10
1080P_hdr0	1	RAW10
4K_linear	2	RAW12
4K_hdr	3	RAW10

The "`/opt/imx8-isp/bin/start_isp.sh`" file has a mode-select parameter.

4 Building and testing

This section describes building and testing the demo using the hardware setup described earlier.

4.1 Creating the i.MX Yocto SDK and installing the toolchain

This section describes how to create the i.MX Yocto SDK and install the toolchain on the I.MX 8M Plus EVK board.

4.1.1 Downloading the repository (if required)

Use the below commands to download the repository (in case the bin folder does not exist already):

```
$ mkdir ~/bin (this step may not be needed if the bin folder already exists)
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
$ export PATH=~/bin:$PATH
```

4.1.2 Setting up Git (if required)

Use the below commands to setup the git repository (if not already done):

```
$ git config --global user.name "Your Name"
$ git config --global user.email "Your Email"
$ git config –list
```

4.1.3 Creating the Yocto build environment

Use the commands below to create the Yocto build environment:

```
$ mkdir imx-yocto-bsp
$ cd imx-yocto-bsp
$ repo init -u https://github.com/nxp-imx/imx-manifest -b imx-linux-nanbield -m
 imx-6.6.3-1.0.0.xml
$ repo sync
$ DISTRO=fsl-imx-xwayland MACHINE=imx8mp-lpddr4-evk source imx-setup-release.sh
 -b build
$ bitbake imx-image-full -c populate_sdk
```

4.1.4 Installing the toolchain

To install the toolchain, perform the following steps:

1. From the "`build`" folder, run the "`./tmp/deploy/sdk/fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx8mp-lpddr4-evk-toolchain-6.6-nanbield.sh`" file.
2. The default directory of the toolchain is "`/opt/fsl-imx-xwayland/6.6-nanbield.sh`". If you install the toolchain in another location, replace the default path in the following sessions with your own path.

4.2 Building the NXP kernel

This section describes how to build the NXP kernel.

4.2.1 Downloading the latest version of NXP kernel

Download the latest version of the NXP kernel using the command below:

```
$ git clone https://github.com/nxp-imx/linux-imx.git -b lf-6.6.3-1.0.0
```

4.2.2 Building the kernel

Use the commands below to build the kernel:

```
$ source /opt/6.6-nanbield/environment-setup-armv8a-poky-linux
$ make mrproper
$ make ARCH=arm64 imx_v8_defconfig O=./build_v8
$ cd build_v8/
$ make ARCH=arm64 -j8
```

4.3 Building the isp-imx

This section describes how to build the isp-imx.

4.3.1 Downloading the latest version of isp-imx

Download the latest version of `isp-imx` using the commands below:

```
$ wget https://www.nxp.com/lgfiles/NMG/MAD/YOCTO/isp-imx-4.2.2.24.1.bin
$ chmod +x isp-imx-4.2.2.24.1.bin
$ ./isp-imx-4.2.2.24.1.bin
```

In Yocto, "`isp-imx`" is located in the path: "`tmp/work/aarch64-mx8mp-poky-linux/isp-imx`".

4.3.2 Building the SDK

Build the SDK using the commands below:

```
$ source /opt/6.6-nanbield/environment-setup-armv8a-poky-linux
$ ./build-all-isp.sh release partial
```

4.4 Building the isp-vvcam

This section describes how to build the `isp-vvcam`.

4.4.1 Downloading the latest version of `isp-vvcam`

Download the latest version of `isp-vvcam` using the commands below:

```
$ git clone https://github.com/nxp-imx/isp-vvcam.git -b lf-6.6.3-1.0.0
```

In Yocto, "`isp-vvcam`" is located in the directory "`build-wayland-8mp/tmp/work/imx8mpevk-poky-linux/kernel-module-isp-vvcam`".

4.4.2 Building the vvcam

Build the `vvcam` file using the commands below:

```
$ source /opt/6.6-nanbield/environment-setup-armv8a-poky-linux
$ export KERNEL_SOURCE_DIR = [the build path of the NXP kernel]
$ ./build-all-vvcam.sh
```

4.5 Storing useful files

This section describes how to store useful files.

4.5.1 Copying useful files to the output directory

Perform the following steps to copy useful files to the output directory:

1. Copy the kernel files to the build-out directory:

```
$ cp linux-imx/build_v8/arch/arm64/boot/dts/freescale/imx8mp-evk-*.dtb [your
 build-out directory]/boot
$ cp linux-imx/build_v8/arch/arm64/boot/Image [the build-out directory]/boot
$ cp linux-imx/build_v8/drivers/staging/media/imx/imx8-media-dev.ko [the
 build-out directory]/sdk
```

2. Copy the isp-imx files to the build-out directory:

```
$ cp -r ./isp-imx-4.2.2.24.1/build_output_release_partial/blob/* [the build-
out directory]/sdk
```

3. Copy the isp-vvcam files to the build-out directory:

```
$ cp ./isp-vvcam/modules/* [the build-out directory]/sdk
```

4.5.2 Sending a file to a board

```
$ scp -r [the build out directory]/sdk/* root@$EVK_IP_Address:/home/root/[your
 test directory in root]
$ scp [the build out directory]/boot/* root@$EVK_IP_Address:/run/media/boot-
mmcblk1p1/
```

The output directory should contain the following files:

```
root@imx8mpevk:~/build-out-guest# ls
OS08a20_8M_10_1080p_hdr.xml              liba3dnr.so              libbufsync_ctrl.so              libdewarp_hal.so     liboslayer.so
OS08a20_8M_10_1080p_linear.xml           libadpcc.so              libcam_calibdb.so               libebase.so          libsom_ctrl.so
OS08a20_8M_10_4k_hdr.xml                 libadpf.so               libcam_device.so                libfpga.so           libversion.so
OS08a20_8M_10_4k_linear.xml              libaec.so                libcam_engine.so                libhal.so            libvom_ctrl.so
Sensor0_Entry.cfg                        libaee.so                libcameric_drv.so               libi2c_drv.so        libvvdisplay_shared.so
Sensor0_Entry_os08a20.cfg                libaf.so                 libcameric_reg_drv.so           libibd.so            os08a20.drv
Sensor1_Entry.cfg                        libaflt.so               libcim_ctrl.so                  libisi.so            os08a20.ko
Sensor1_Entry_os08a20.cfg                libahdr.so               libcommon.so                    libjsoncpp.so        ov2775.ko
VSI_Monitor.cfg                          libappshell_ebase.so     libcppnetlib-client-connections.so  libjsoncpp.so.1.9.0  run.sh
basler-camera-driver-vvcam.ko            libappshell_hal.so       libcppnetlib-client-connections.so.0  libjsoncpp.so.21   start_isp.sh
dewarp_config                            libappshell_ibd.so       libcppnetlib-client-connections.so.0.13.0  libmedia_server.so  tuningext
imx8-media-dev.ko                        libappshell_oslayer.so   libcppnetlib-server-parsers.so  libmim_ctrl.so       video_test
imx8mp-evk-revA3-8mic-revE.dtb           libavs.so                libcppnetlib-server-parsers.so.0  libmipi_drv.so     vvcam-dwe.ko
imx8mp-evk-revb4-hifiberry-dacplusadc.dtb  libawb.so              libcppnetlib-server-parsers.so.0.13.0  libmom_ctrl.so  vvcam-isp.ko
imx8mp-evk-rpmsg.dtb                     libawdr3.so              libcppnetlib-uri.so             libos08a20.so        vvcam-video.ko
isp_media_server                         libbase64.so             libcppnetlib-uri.so.0           libos08a20.so.1      vvext
liba2dnr.so                              libbufferpool.so         libcppnetlib-uri.so.0.13.0      libos08a20.so.1.0.0
```

Figure 4. Output directory

4.6 Selecting a device tree

To select a device tree, perform the following steps:

1. "`imx8mp-evk-os08a20.dtb`" - # single os08a20, connect to CSI1.
2. "`imx8mp-evk-dual-os08a20.dtb`" - # dual os08a20, connect to CSI1 and CSI2.
3. "`imx8mp-evk-os08a20-ov5640.dtb`" - # ov5640 and os08a20 (os08a20 -> CSI1, ov5640 -> CSI2).

4.7 Editing the sensor-configuration file and selecting the correct mode

The below example shows how to edit a sensor-configuration file and select the correct mode.

Sensor0_Entry.cfg (example):

```
name="os08a20" drv = "os08a20.drv"
mode= 2
[mode.0]
xml = "OS08a20_8M_10_1080p_linear.xml"
dwe = "dewarp_config/sensor_dwe_os08a20_1080P_config.json"
[mode.1]
xml = "OS08a20_8M_10_1080p_hdr.xml"
dwe = "dewarp_config/sensor_dwe_os08a20_1080P_config.json"
[mode.2]
xml = " OS08a20_8M_10_4k_linear.xml"
dwe = "dewarp_config/sensor_dwe_os08a20_4K_config.json"
[mode.3]
xml = " OS08a20_8M_10_4k_hdr.xml"
dwe = "dewarp_config/sensor_dwe_os08a20_4K_config.json""
```

4.8 Enabling the ISP and camera on the board

This section describes how to enable the ISP and camera module on the board.

4.8.1 Adding to a path

Use the below command to add a path:

```
$ export LD_LIBRARY_PATH=$pwd:$LD_LIBRARY_PATH
```

4.8.2 Stopping the default ISP

Use the below command to

```
$ systemctl stop imx8-isp.service
```

4.8.3 Removing existing modules

Use the below command to remove existing modules.

```
$ rmmod vvcam-dwe
$ rmmod vvcam-isp
$ rmmod vvcam-video
$ rmmod imx8-media-dev.ko
$ rmmod os08a20.ko
```

4.8.4 Installing modules

Use the below command to install modules:

```
$ insmod vvcam-dwe
$ insmod vvcam-isp
$ insmod vvcam-video
$ insmod imx8-media-dev.ko
$ insmod os08a20.ko
```

4.8.5 Starting the ISP media server

To start the ISP media server (**Single sensor**), use the below command:

```
$ ./isp_media_server CAMERA0&
```

To start the ISP media server (**Dual sensor**), use the below command:

```
$ ./isp_media_server DUAL_CAMERA&
```

4.9 OS08A20 test cases

The following section describes the OS08A20 sample test cases for single sensor and dual sensor modes.

4.9.1 Single sensor

***Mode 0: 1080p linear**:*

- Change "`Sensor0_Entry.cfg`" to mode 0:

```
$ gst-launch-1.0 -v v4l2src device=/dev/video3 ! "video/x-
raw,format=YUY2,width=1920,height=1080" ! queue ! waylandsink
```

***Mode 1: 1080p HDR**:*

- Change "`Sensor0_Entry.cfg`" to mode 1:

```
$ gst-launch-1.0 -v v4l2src device=/dev/video3 ! "video/x-
raw,format=YUY2,width=1920,height=1080" ! queue ! waylandsink
```

***Mode 2: 4K linear**:*

- Change "`Sensor0_Entry.cfg`" to mode 2:

```
$ gst-launch-1.0 -v v4l2src device=/dev/video3 ! "video/x-
raw,format=YUY2,width=3840,height=2160" ! queue ! waylandsink
```

***Mode 3: 4K linear**:*

- Change "`Sensor0_Entry.cfg`" to mode 3:

```
$ gst-launch-1.0 -v v4l2src device=/dev/video3 ! "video/x-
raw,format=YUY2,width=3840,height=2160" ! queue ! waylandsink
```

4.9.2 Dual sensor

Mode 0: 1080p linear:

- Change "`Sensor0_Entry.cfg`" and “`Sensor1_Entry.cfg`" to mode 0:

```
$ gst-launch-1.0 -v v4l2src device=/dev/video3 ! "video/x-
raw,format=YUY2,width=1920,height=1080" ! queue ! waylandsink window-width=640
 window-height=480 --no-position &
$ gst-launch-1.0 -v v4l2src device=/dev/video4 ! "video/x-
raw,format=YUY2,width=1920,height=1080" ! queue ! waylandsink window-width=640
 window-height=480 --no-position &
```

Mode 1: 1080p HDR:

- Change "`Sensor0_Entry.cfg`" and “`Sensor1_Entry.cfg`" to mode 1:

```
$ gst-launch-1.0 -v v4l2src device=/dev/video3 ! "video/x-
raw,format=YUY2,width=1920,height=1080" ! queue ! waylandsink window-width=640
 window-height=480 --no-position &
$ gst-launch-1.0 -v v4l2src device=/dev/video4 ! "video/x-
raw,format=YUY2,width=1920,height=1080" ! queue ! waylandsink window-width=640
 window-height=480 --no-position &
```

Mode 2: 4K linear:

- Change "`Sensor0_Entry.cfg`" and "`Sensor1_Entry.cfg`" to mode 2:

```
$ gst-launch-1.0 -v v4l2src device=/dev/video3 ! "video/x-
raw,format=YUY2,width=3840,height=2160" ! queue ! waylandsink window-width=640
 window-height=480 --no-position &
$ gst-launch-1.0 -v v4l2src device=/dev/video4 ! "video/x-
raw,format=YUY2,width=3840,height=2160" ! queue ! waylandsink window-width=640
 window-height=480 --no-position &
```

Mode 3: 4K linear:

- Change "`Sensor0_Entry.cfg`" and "`Sensor1_Entry.cfg`" to mode 3:

```
$ gst-launch-1.0 -v v4l2src device=/dev/video3 ! "video/x-
raw,format=YUY2,width=3840,height=2160" ! queue ! waylandsink window-width=640
 window-height=480 --no-position &
$ gst-launch-1.0 -v v4l2src device=/dev/video4 ! "video/x-
raw,format=YUY2,width=3840,height=2160" ! queue ! waylandsink window-width=640
 window-height=480 --no-position &
```

4.10 Disabling or bypassing dewarp functionality

To bypass the dewarp configuration, set the "`dewarp bypass`" parameter in the dewarp configuration file to "`true`".

```
{
        "dewarpConfigArray" :[
                {
                        "source_image":{
                                "width"  : 1920,
                                "height" : 1080
                        },

                        "?dewarpType": "LENS_CORRECTION, FISHEYE_EXPAND, SPLIT_SCREEN",
                        "dewarpType": "FISHEYE_DEWARP",

                        "scale": {
                                "roix"   : 0,
                                "roiy"   : 0,
                                "factor" : 1.0
                        },

                        "split": {
                                "horizon_line"      : 540,
                                "vertical_line_up"  : 960,
                                "vertical_line_down": 960
                        },

                        "bypass" : true,
                        "hflip"  : false,
                        "vflip"  : false,

                        "camera_matrix"   : [1.9584556270377586e+003,0.0, 9.6819933899253533e+00
                        "distortion_coeff": [-1.2839656060464022e-001, 1.4121087523973114e-001,
```

Figure 5. Disabling or bypassing the `dewarp` parameter

To disable the dewarp functionality, run the following command after starting "`isp_media_server`", and before running the "`gstream`" command:

```
$ v4l2-ctl -d 2 -c viv_ext_ctrl='{<id>:<pipeline.s.dwe.onoff>;<enable>:false}'
```

5 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2022-2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6 Revision history

Table 2 summarizes the changes done to this document.

Table 2. Revision history

Document ID	Release date	Description
AN13712 v.3.0	9 October 2024	• Updated the kernel version and resolution of the sensor modes. • Added dual sensor mode test cases in Section 4.9.2.
AN13712 v.2	4 September 2023	Updated the Linux kernel version to 6.1.22.
AN13712 v.1	29 November 2022	Updated Section 1.
AN13712 v.0	24 August 2022	Initial release.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Intel, the Intel logo, Intel Core, OpenVINO, and the OpenVINO logo — are trademarks of Intel Corporation or its subsidiaries.

Contents