AN13713

8MP ISP New Camera Porting

Rev. 0 — 24 August 2022 Application note

Document information

Information	Content
Keywords	
Abstract	

1 Building ISP module, drivers, and libraries

To run a camera sensor on the i.MX 8MP board, build the ISP module, drivers, and libraries (including <cam>.ko, <cam>.so, and <cam>.drv) for your sensor. In addition, get the camera calibration files (.xml), dewarp calibration files (.json), and sensor-configuration files (.cfg) ready to set correct ISP parameters and stream mode.

1. If you use NXP supported sensors (such as OS08A20), get the drivers and libraries from "yocto imx-isp" and "yocto vvcam" after you build the whole ISP module.
2. Find detailed instructions for writing code or building new drivers and libraries in "iMX8MP_CAMERA_DISPLAY_GUIDE.pdf", available at the NXP website. To generate sensor-calibration files, calibrate the new camera sensor in a lab and then generate the ".xml" files using the camera calibration tool and the ".json" files using the dewarp calibration tool. Sensor-configuration files can be easily created by editing the existing files. An example of the OS08A20 sensor-configuration file is as follows:

```
Sensor0_Entry.cfg:
name="os08a20"
drv = "os08a20.drv"
mode= 2
[mode.0]
xml = "OS08a20_8M_02_1080p.xml"
dwe = "dewarp_config/sensor_dwe_1080P_config.json"
[mode.1]
xml = "OS08a20_8M_02_1080p.xml"
dwe = "dewarp_config/sensor_dwe_1080P_config.json"
[mode.2]
xml = " OS08a20_8M_02_4k.xml"
dwe = "dewarp_config/sensor_dwe_4K_config.json"
[mode.3]
xml = " OS08a20_8M_02_4k.xml"
dwe = "dewarp_config/sensor_dwe_4K_config.json"
```

Note: *If you use a flat len sensor and decide not to calibrate the new dewarp configuration files, bypass the ISP's dewarp feature. Use the dewarp configuration files of other sensors with the same resolution and set the "dewarp bypass" parameter to "true".*

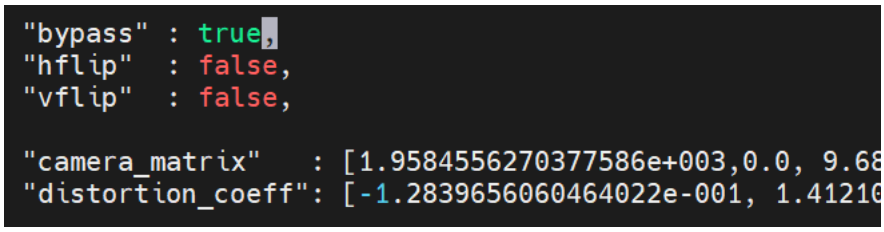

Figure 1. Bypass parameter

2 Building the ISP module

To build the ISP module, get the correct versions of toolchain, linux kernel, "yocto imx-isp" and "yocto vvcam" on your local machine.

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

1. Download the repository (if needed):

```
$ 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
```

2. Set up Git (if needed):

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

3. Create the Yocto build environment:

```
$ mkdir imx-yocto-bsp
$ cd imx-yocto-bsp
$ repo init -u https://source.codeaurora.org/
external/imx/imx-manifest  -b imx-linux-kirkstone -m
 imx-5.15.32-2.2.0.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. From the "build" folder, run the "./tmp/deploy/sdk/fsl-imx-xwayland-glibc-x86_64-imx-image-full-armv8a-imx8mp-lpddr4-evk-toolchain-5.15-kirkstone.sh" file.
5. The toolchain path is "/opt/my-tool-chain-path".

2.2 Building the NXP kernel

1. Download the correct version of the NXP kernel from https://source.codeaurora.org/external/imx/linux-imx.
2. Build the kernel using the following command:

```
$ source [/opt/my-tool-chain-path]/environment-setup-
[depend_on_the_toolchain]
$ make mrproper
$ make ARCH=arm64 imx_v8_defconfig O=./build_v8
$ cd build_v8/
$ make ARCH=arm64 -j8
```

Note: *The build path of the Linux kernel should be the path of the "build_v8" folder.*

3 Building the "yocto imx-isp" (details are in "build-all-isp.sh")

1. Download the correct version of "yocto imx-isp" as follows:

```
$ wget https://www.nxp.com/lgfiles/NMG/MAD/YOCTO/isp-
imx-4.2.2.18.0.bin
$ chmod a+x isp-imx-4.2.2.18.0.bin
$ ./isp-imx-4.2.2.18.0.bin
```

In Yocto, "imx-isp" is in the "tmp/work/cortexa53-crypto-mx8mp-poky-linux/isp-imx" folder.

2. Build "yocto imx-isp" as follows:

```
$ source [/opt/my-tool-chain-path]/environment-setup-
[depend_on_the_toolchain]
```

Then:

```
$ ./build-all-isp.sh release partial V4L2
```

Or:

```
$ rm -rf appshell/build
$ mkdir -p appshell/build
$ cd appshell/build
$ cmake -DCMAKE_BUILD_TYPE=release -
DISP_VERSION=ISP8000NANO_V1802 -DPLATFORM=ARM64 -
DAPPMODE=V4L2 -DQTLESS=1 -DFULL_SRC_COMPILE=1 -DWITH_DWE=1 -
DWITH_DRM=1 -DSERVER_LESS=1 -DSUBDEV_V4L2=1 -DENABLE_IRQ=1-
DPARTITION_BUILD=0 -D3A_SRC_BUILD=0 -DIMX_G2D=ON -
DSDKTARGETSYSROOT=$SDKTARGETSYSROOT -Wno-dev ..
$ make -j8
```

4 Building "yocto vvcam" (details are in "build-all-vvcam.sh")

1. Download the correct version of "yocto vvcam" from https://source.codeaurora.org/external/imx/isp-vvcam.
   In Yocto, "vvcam" is in the "build-wayland-8mp/tmp/work/imx8mpevk-poky-linux/kernel-module-isp-vvcam" folder.
2. Build "vvcam" using the following command (you can also use the "build-all-vvcam.sh" file):

```
$ source [/opt/my-tool-chain-path]/environment-setup-
[depend_on_the_toolchain]
```

Then:

```
$ export KERNEL_SOURCE_DIR = [the build path of the NXP
 kernel]
$ ./build-all-vvcam.sh
```

Or:

```
$ cd vvcam/v4l2
$ make KERNEL_SRC=[the build path of the NXP kernel]
 ENABLE_IRQ=yes
```

5 Copying useful files listed below to a build-out directory

Kernel:

- ".dtb" files (build_v8/arch/arm64/boot/dts/freescale/imx8mp-evk-*.dtb)
- Image (build_v8/arch/arm64/boot/Image)
- "imx-media-dev.ko" (build_v8/drivers/stagging/media/imx/imx-media-dev.ko)

Imx-isp:

- Libraries (isp-imx-4.2.2.18.0/appshell/build/generated/[build mode]/lib/*.so*)

- Sensor-related files (e.g. "isp-imx-4.2.2.18.0/appshell/build/generated/[build mode]/bin/OS08a20.*" and "isp-imx-4.2.2.18.0/appshell/build/generated/[build mode]/bin/os08a20.*")
- "isp_media_server" (isp-imx-4.2.2.18.0/appshell/build/generated/[build mode]/bin/isp_media_server)
- Test files (isp-imx-4.2.2.18.0/appshell/build/generated/[build mode]/bin/test_*)
- Sensor-configuration files (isp-imx-4.2.2.18.0/appshell/build/generated/[build mode]/bin/Sensor*_Entry.cfg)
- "vvext" (isp-imx-4.2.2.18.0/appshell/build/generated/[build mode]/bin/vvext)
- Dewarp configurations (isp-imx-4.2.2.18.0/dewarp/dewarp_config/*)
- Scripts to run the ISP ("isp-imx-4.2.2.18.0/imx/start_isp.sh" and "isp-imx-4.2.2.18.0/imx/run.sh")

Vvcam:

- Sensor driver (isp-vvcam/vvcam/v4l2/sensor/os08a20/os08a20.ko)
- "vvcam-dwe" (isp-vvcam/vvcam/v4l2/video/vvcam-dwe.ko)
- "vvcam-isp" (isp-vvcam/vvcam/v4l2/video/vvcam-isp.ko)
- "vvcam-video" (isp-vvcam/vvcam/v4l2/video/vvcam-video.ko)

Note: *When building the ISP SDK, keep the Linux and SDK versions the same as the Yocto release.*

6 Using SCP to remotely send the files in the build-out directory to EVK

```
$ scp -r [the build out directory] root@$EVK_IP_Address:/home/
root/[the build out directory]
$ scp imx8mp-evk-*.dtb root@$EVK_IP_Address:/run/media/
mmcblk1p1
$ scp Image root@$EVK_IP_Address:/run/media/mmcblk1p1
```

Note: *"libtinyxml2.so" is required for the build. However, it may not be included in all SDK configurations. You can build this package separately if you have the Yocto environment or download pre-built package from a package distribution repository (such as packages.debian.org; choose "arm64" with the correct version).*

7 Setting the environment when booting

```
>setenv fdtfile imx8mp-evk-xxx.dtb
>saveenv
>boot
```

After generating and copying the necessary files into a folder, the ISP folder can look like Figure 2.

```
root@imx8mpevk:~/build_output_release# ls
AR1335.xml                      isp_media_server        libcam_device.so                        libmipi_drv.so
AR1335_12MP.xml                 isp_test                libcam_engine.so                        libmom_ctrl.so
AR1335_4K.xml                   isp_tpg_test            libcameric_drv.so                       libos08a20.so
ISP8000NANO_V1802               kill_app.sh             libcameric_reg_drv.so                   libos08a20.so.1
OS08a20_4k.xml                  liba2dnr.so             libcim_ctrl.so                          libos08a20.so.1.0.0
OS08a20_8M_02_1080p.xml         liba3dnr.so             libcommon.so                            liboslayer.so
OS08a20_8M_02_1080p_10bit.xml   libadpcc.so             libcppnetlib-client-connections.so      libov2775.so
OS08a20_8M_02_4k.xml            libadpf.so              libcppnetlib-client-connections.so.0    libov2775.so.1
OS08a20_8M_02_720p.xml          libaec.so               libcppnetlib-client-connections.so.0.13.0 libov2775.so.1.0.0
OS08a20_fisheye.xml             libaee.so               libcppnetlib-server-parsers.so          libsom_ctrl.so
OS08a20_pentax_04.xml           libaf.so                libcppnetlib-server-parsers.so.0        libtuning_ctrl.so
OS08a20_pentaxcombined_01.xml   libaflt.so              libcppnetlib-server-parsers.so.0.13.0   libversion.so
OV2775.xml                      libahdr.so              libcppnetlib-uri.so                     libvom_ctrl.so
OV2775_8M_02_1080p.xml          libappshell_ebase.so    libcppnetlib-uri.so.0                   libvvdisplay_shared.so
OV2775_8M_02_720p.xml           libappshell_hal.so      libcppnetlib-uri.so.0.13.0              os08a20.drv
OV2775_fisheye.xml              libappshell_ibd.so      libdewarp_hal.so                        os08a20.ko
OV2775_pentax_04.xml            libappshell_oslayer.so  libebase.so                             ov2775.drv
OV2775_pentaxcombined_01.xml    libar1335.so            libfpga.so                              ov2775.ko
Sensor0_Entry.cfg               libar1335.so.1          libhal.so                               run-os08a20.sh
Sensor1_Entry.cfg               libar1335.so.1.0.0      libi2c_drv.so                           video_test
ar1335.drv                      libavs.so               libibd.so                               vvcam-dwe.ko
basler-camera-driver-vvcam.ko   libawb.so               libisi.so                               vvcam-isp.ko
case                            libawdr3.so             libjsoncpp.so                           vvcam-video.ko
dewarp_config                   libbase64.so            libjsoncpp.so.1.9.0                     vvext
drm_test                        libbufferpool.so        libjsoncpp.so.21
gstshark_2021-03-24_10:28:06    libbufsync_ctrl.so      libmedia_server.so
imx8-media-dev.ko               libcam_calibdb.so       libmim_ctrl.so
```

Figure 2. ISP folder

8 Running ISP and sensor (e.g. OS08A20) on 8MP EVK

1. Add it to the following path:

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

2. Because you build the ISP module in a new folder, the Yocto default ISP starts automatically. Therefore, stop the default ISP as follows:

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

3. Remove the existing modules:

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

4. Install modules:

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

5. Start the ISP:

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

6. Run the stream capture using the "gstream" command:

```
$ gst-launch-1.0 -v v4l2src device=/dev/video2 ! "video/x-
raw,format=YUY2,width=1920,height=1080" ! queue ! waylandsink
```

Reference: "build-all-isp.sh", "build-all-vvcam.sh"

9 Porting a new camera to the tuning tool

1. Download the tuning server and client released by NXP.

2. To enable other sensor tuning, you do not have to rebuild "isp-imx" and "vvcam". After porting your sensor according to the porting guide and building it, you can get the related sensor files in the "build_output_release" directory. Just push the sensor-related files to the EVK and then you can run the tuning tool for your own sensor:

```
$ scp build_output_release/* root@$EVK_IP_Address:/home/root/
build_output_release/
$ scp build_output_release/modules/[your camera sensor].ko
 root@$EVK_IP_Address:/home/root/build_output_release/
modules/
```

3. On the EVK board, execute the following steps in the same order (or) run the "exec_isp_ts_[your camera sensor].sh" script:

```
$ cd /home/root/build_output_release/
$ export LD_LIBRARY_PATH=$PWD:$LD_LIBRARY_PATH$ systemctl
 stop imx8-isp.service
$ rmmod vvcam-dwe
$ rmmod vvcam-isp
$ rmmod vvcam-video
$ rmmod imx8-media-dev.ko
$ rmmod ar1335.ko
$ insmod modules/ar1335.ko moto_distance=100
$ insmod modules/vvcam-isp.ko
$ insmod modules/vvcam-dwe.ko
$ insmod modules/vvcam-video.ko
$ insmod modules/imx8-media-dev.ko
$ systemctl stop weston*
$ ./tuning-server
```

10 Revision history

Table 1. Revision history

Revision number	Date	
0	24 August 2022	Initial release

11 Legal information

11.1 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.

11.2 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.

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.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

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.

11.3 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.

Contents

For more information, please visit: http://www.nxp.com

Date of release: 24 August 2022
Document identifier: AN13713