AN14376

i.MX Camera Software Pack

Rev. 2.0 — 28 March 2025 Application note

Document information

Information	Content
Keywords	AN14376, ISP, BSP, i.MX 8M Plus, camera software pack
Abstract	i.MX camera software pack helps customers quickly enable off-the-shelf camera modules on i.MX 8M Plus-EVK using the ISP.

1 Introduction

i.MX camera software pack includes camera drivers, libraries, calibration files, and Yocto recipes to enable customers to use selected off-the-shelf camera modules out of the box while using the i.MX 8M Plus internal image signal processor (ISP).

This document outlines the contents of the i.MX camera software pack for the i.MX 8M Plus. It describes steps for using supported camera modules, with the i.MX 8M Plus ISP for the following supported BSPs:

- LF6.6.3-1.0.0
- LF6.6.52-2.2.0

2 Hardware and software requirements

Table 1 describes the hardware and software requirements for using the software pack.

Table 1. Hardware and software details

Category	Description
Hardware	• NXP i.MX 8M Plus Evaluation Kit (EVK) • Supported camera modules • XRPI-CAM-MINISAS adaptor board • 22 pin/15 pin camera connector FPC cable • Display output (Monitor) and HDMI cable • Personal computer
Software	• Ubuntu 22.04 LTS (If building an image using Yocto) • Serial Terminal: TeraTerm setup for Windows OS

2.1 XRPI-CAM-MINISAS adaptor board

This adaptor board connects cameras using Expansion Interface 22 pin 0.5 mm pitch FPC camera cable and mini-SAS connector for MIPI-CSI connection to the i.MX 8M Plus EVK board. It is going to be launched soon and available for purchase on the NXP website.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

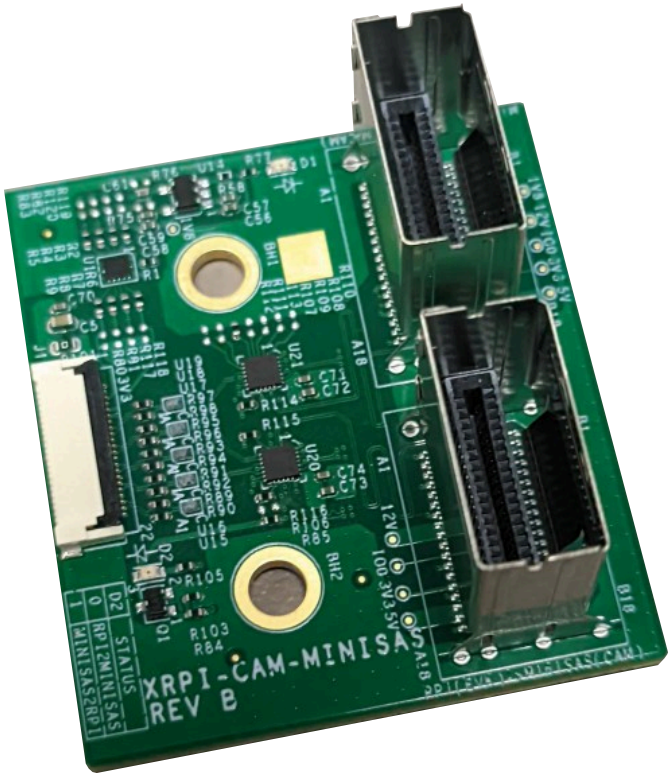


Figure 1. XRPI-CAM-MINISAS

2.2 Supported camera modules

The currently supported camera modules include a wide variety of features from Camera Expansion Interface compatible modules, which can be purchased on any online store. The modules come with the following sensors and lens combinations, along with the online links to purchase them:

- PiBiger Sony IMX 219:
  - 8 MP (3280 × 2464 active resolution)
  - Current enabled resolution 1920 x 1080 30 fps

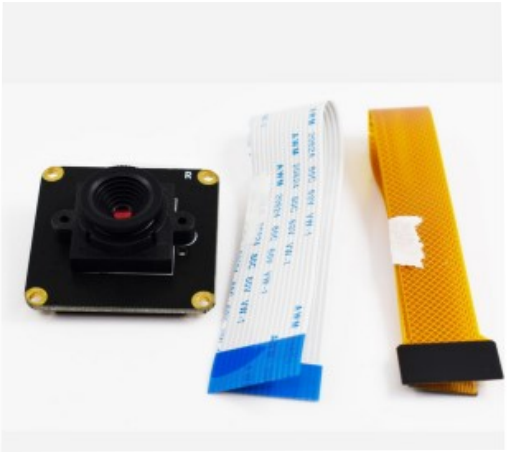


Figure 2. PiBiger Sony IMX 219

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- InnoMaker OmniVision OV5647
  - 5MP (2592 × 1944, 1920 x 1080)
  - Currently enabled resolution
  - 1920 x 1080 30 fps BGGR10

Figure 3. InnoMaker OmniVision OV5647

- ONSemiconductor AR0144:
  - 1MP (1280 x 800)
  - Currently enabled as a monochrome sensor for i.MX 8M Plus ISP at 1280 x 800

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Figure 4. ONSemiconductor AR0144

- ONSemiconductor AR0830
  - 8MP
  - Active resolution 3840x2160 at 30 FPS
  - OnSemi AR0830 PRISM module is used with PRISM/IAS to NXP i.MX 8MP EVK adapter board

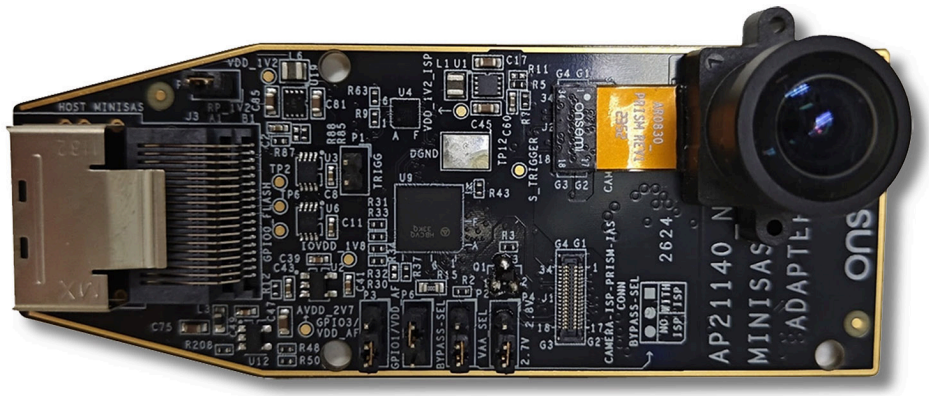


Figure 5. AR0830 PRISM module attached on PRISM/IAS to NXP i.MX 8MP EVK adapter board

- Sony IMX 258
  - 12MP
  - Active resolution 3840x2160 at 30 FPS, 1920x1080 at 30 FPS

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

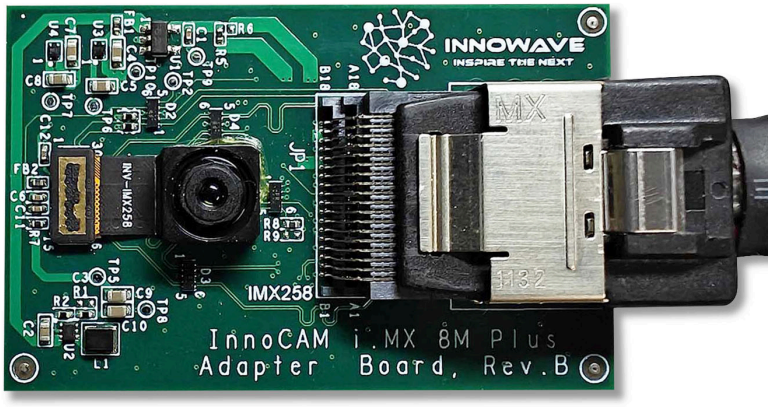


Figure 6. Sony IMX 258 sensor on Innowave InnoCAM i.MX 8M Plus adapter board

2.3 Camera FPC cable

The 22 pin (0.5 mm pitch) to 15 pin (1 mm pitch) FPC cable comes included with camera modules supporting the 22-pin camera connector interface.

Figure 7. 22 pin (0.5 mm pitch) to 15 pin (1 mm pitch) FPC cable

AR0144 camera module board connects to the i.MX 8M Plus EVK board through the 22 pin (0.5 mm pitch) FPC cable.

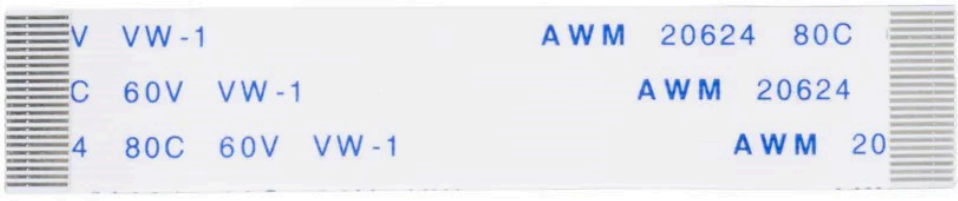


Figure 8. 22 pin (0.5 mm) to 22 pin (0.5 mm)

3 Using the software pack

This section explains the two different methods for enabling the camera module using the software pack and describes the prerequisite setup for both methods:

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

3.1 Local build

The software pack includes patches in the form of Yocto recipes for the ISP module, drivers, and calibration files necessary to enable each sensor camera module. It allows users to build a flashable BSP image locally using Yocto.

The Yocto recipes can be obtained from the link: https://github.com/nxp-imx-support/imx-camera-sw-pack

3.1.1 Prerequisite

If compiling recipes locally using the camera software pack, these steps are required. These steps set up the base Yocto build environment.

3.1.2 Set up build environment for Yocto project

To build the Yocto image for the i.MX 8M Plus, perform the following steps:

1. Install the essential Yocto project host packages as follows:

```
$ sudo apt install gawk wget git diffstat unzip texinfo gcc build-essential \
chrpath socat cpio python3 python3-pip python3-pexpect xz-utils debianutils \
iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev \
python3-subunit mesa-common-dev zstd liblz4-tool file locales -y
$ sudo locale-gen en_US.UTF-8
```

2. To set up repo utility, perform the following steps:
   a. Create a bin folder in the home directory, as follows:

```
$ 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
```

   b. To ensure that the `~/bin` folder is in your `PATH` variable, add the following command line to the `.bashrc` file:

```
$ export PATH=~/bin:$PATH
```

3. Set up the Git as follows:

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

4. Set up Yocto for required BSP as follows:

```
$ mkdir imx-yocto-bsp
$ cd imx-yocto-bsp
// For Setting up 6.6.3-1.0.0 BSP
$: repo init -u https://github.com/nxp-imx/imx-manifest -b imx-linux-
scarthgap -m imx-6.6.36-2.1.0.xml

//OR

// For Setting up 6.6.52-2.2.0 BSP
$: repo init -u https://github.com/nxp-imx/imx-manifest -b imx-linux-
scarthgap -m imx-6.6.52-2.2.0.xml
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

```
$ repo sync
```

Note:
`imx-yocto-bsp` directory is referred as the `Yocto_Directory`.
Yocto directory contains a "sources" directory, containing various recipes used to build one or more build directories.

5. Image configuration:

```
DISTRO=fsl-imx-xwayland MACHINE=imx8mp-lpddr4-evk source imx-setup-release.sh
 -b build
```

Where:

- `DISTRO=<distro name> fsl-imx-xwayland` is the distro configuration
- `MACHINE=<machine name> imx8mp-lpddr4-evk` is the board configuration
- `-b <build_directory>` specifies the name of the build directory

Note: *The "build" directory is referred as the `build_directory` in the next section.*

6. Build the image using the following command:

```
$ bitbake imx-image-full
```

Now, your Yocto build environment is set up, allowing you to add and compile the recipes for individual camera modules into the final image build.

For more information on Yocto build setup instructions, refer to the *i.MX Yocto Project User's Guide* (document IMXLXYOCTOUG) on the IMXLINUX webpage. See Figure 9.

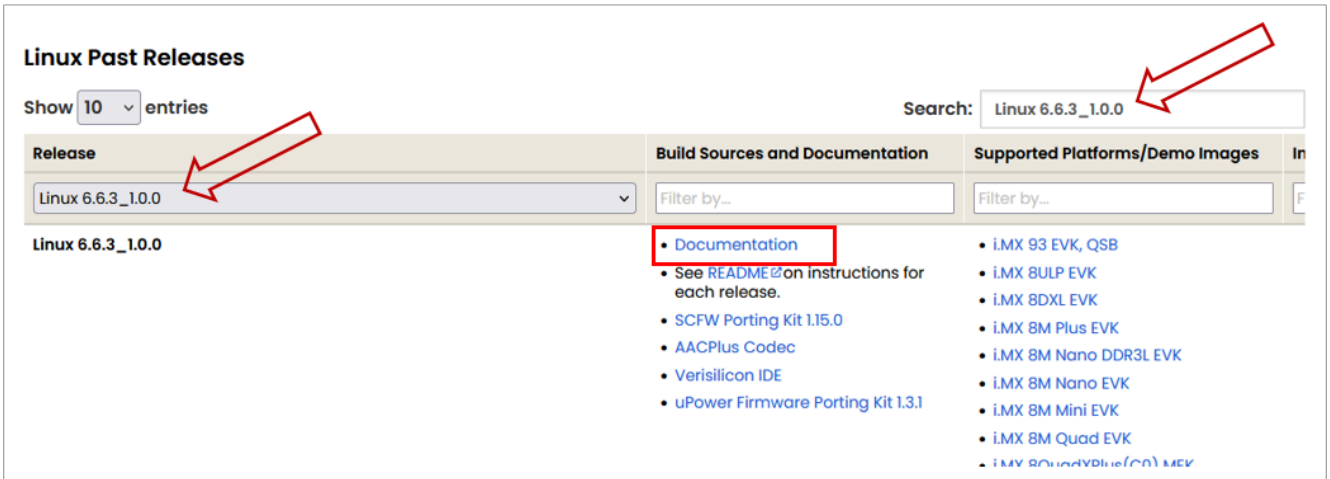


Figure 9. Downloading the Yocto documentation for LF6.6.3_1.0.0

3.2 Use precompiled binaries

The software pack also includes precompiled ISP modules, drivers, and libraries for each sensor. This content can be directly copied onto the supported i.MX 8M Plus Linux BSP image.

The binaries directory for each camera sensor contains precompiled drivers, kernel modules, and libraries required to enable the camera sensor to use the i.MX 8M Plus ISP. It also contains the camera calibration files (.xml), dewarp calibration files (.json), and sensor-configuration files (.cfg) to get the ISP read and set the required parameters for streaming camera data. Binaries can be downloaded using the following links:

- LF6.6.3_P24.1-Binaries

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- LF6.6.52_P24.4-Binaries

Table 2 lists the contents of the binaries directory.

Table 2. Contents of binaries directory

Directory	Content
`Kernel`	Device tree file to enable sensor
`imx-isp`	• Compiled libraries • Sensor calibration and dewarp files • Modified scripts to run ISP on board bootup
`isp-vvcam`	Sensor driver kernel module
`copy_binaries.sh`	Script to copy binaries to the right location on the board

3.2.1 Prerequisite

This section includes steps to use the precompiled binaries with pre-built image for i.MX 8M Plus EVK.

Download a pre-built image from IMXLINUX.

3.2.2 Use the precompiled binaries with prebuilt image

1. Select and download the right image package for i.MX 8M Plus EVK marked for the required BSP version:
   - Linux 6.6.3-1.0.0
   - Linux 6.6.52-2.2.0

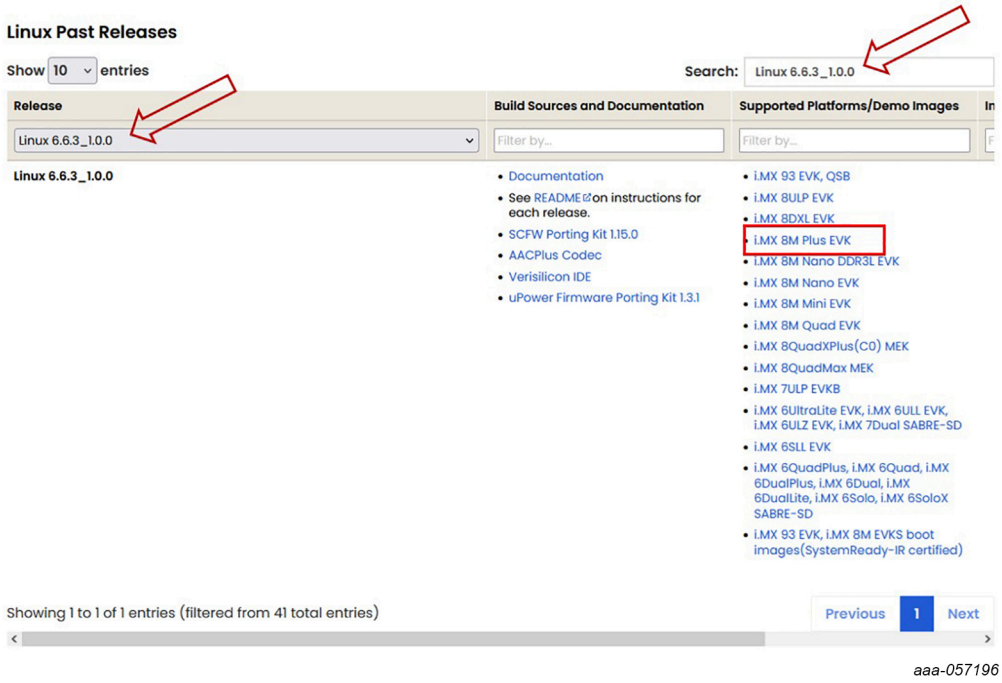


Figure 10. Example for downloading Linux 6.6.3_1.0.0 i.MX 8M Plus image

2. Extract the contents of the downloaded BSP zip file, taking Linux 6.6.3_1.0.0 as an example `LF_v6.6.3-1.0.0_images_IMX8MPEVK.zip`.
3. After extracting, select the `imx-image-full-imx8mpevk.wic` image file and flash it onto an SD card using the `Rufus.exe` application on the Windows PC.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

If using Ubuntu, run `sudo dd if=imx-image-full-imx8mpevk.wic of=/dev/sd<partition> bs=1M conv=fsync`.

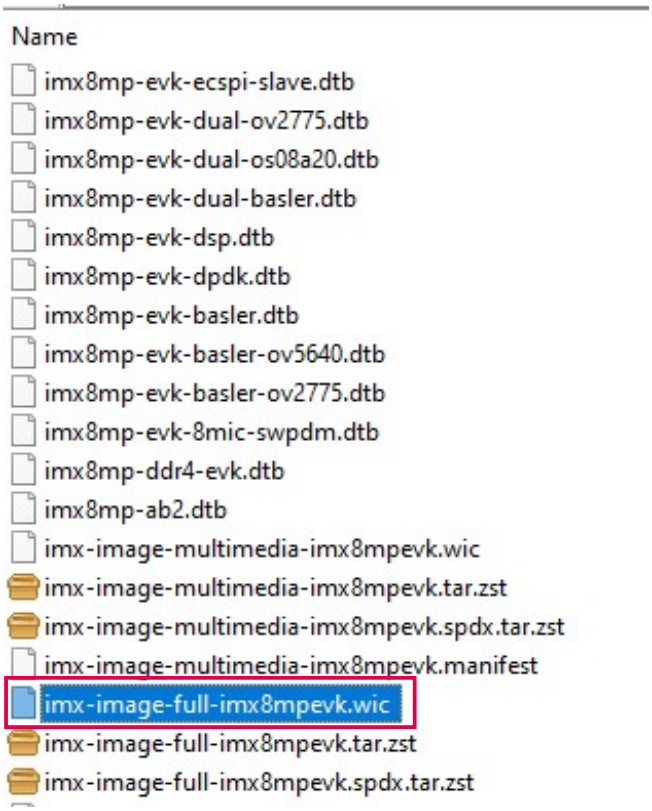


Figure 11. Full image for i.MX 8M Plus

4. Boot up the i.MX 8M Plus EVK with the flashed SD card while connected to the serial console terminal on your PC.

4 Enable Sony IMX 219 camera module

This section describes the two ways to enable the IMX 219 camera module and steps to connect it with the i.MX 8M Plus EVK board via the XRPI-CAM-MINISAS adaptor board:

- Build an i.MX 8M Plus EVK Yocto Image with Sony IMX 219 camera module enabled
- Enable IMX 219 using pre-compiled binaries

4.1 Build an i.MX 8M Plus EVK Yocto image with Sony IMX 219 camera module enabled

1. Perform the Prerequisite for Local build.
2. To set up software pack repo, perform the following steps:
   a. Clone and checkout required BSP branch:
      - LF6.6.3_P24.1
      - LF6.6.52_P24.4

```
$ git clone https://github.com/nxp-imx-support/imx-camera-sw-pack.git
$ git checkout LF6.6.3_P24.1
//Or
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

```
$ git checkout LF6.6.52_P24.4
```

b. Copy only `meta-imx8mp-isp-imx219` into the `sources` directory in your `Yocto_Directory`:

```
$ cp -r  imx-camera-sw-pack/sony_imx219_sensor/i.MX8MPLUS/meta-imx8mp-isp-imx219  imx-yocto-bsp/sources/
```

3. To build and flash the image enabled with the IMX 219 camera module, perform the following steps:
   a. Navigate to the `Yocto_Directory`:

```
$ cd  imx-yocto-bsp
```

   b. Source the setup script to configure Yocto build system to enable the IMX 219 camera module and build the image:

```
$ source sources/meta-imx8mp-isp-imx219/setup/setup-env-imx8mp-imx219 -b
 build
$ bitbake imx-image-full
```

   c. After the build process is completed, the image file `imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst` is available at the following location:

```
$ imx-yocto-bsp/build/tmp/deploy/images/imx8mp-lpddr4-evk/ imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst
```

   d. Flash this image on to the SD card:

```
$ zstdcat imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst | sudo dd of=/dev/sd<partition> bs=1M conv=fsync
```

4. The following set of commands are executed on the i.MX 8M Plus EVK board using a serial console terminal:
   a. Now, boot up the i.MX 8M Plus EVK board with the serial console terminal open.
   While the board is booting up, press any key on the keyboard when you see the prompt shown in Figure 12 on your terminal to pause U-Boot execution.

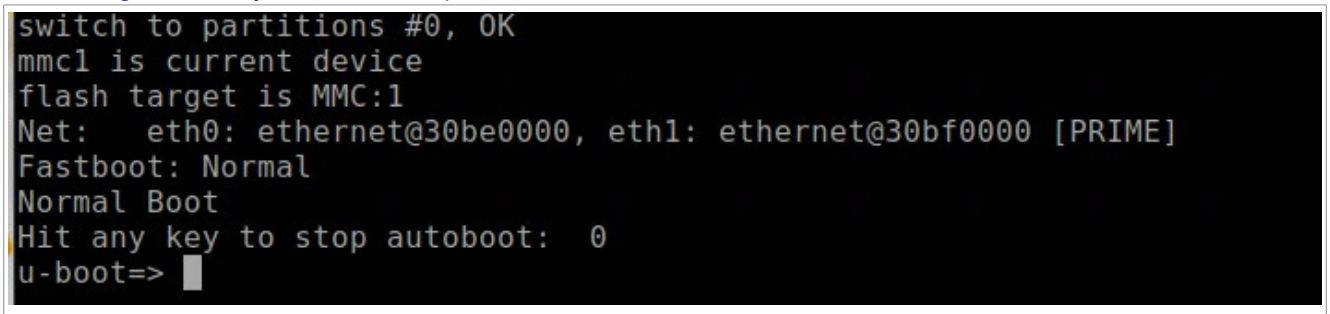


Figure 12. U-Boot prompt

   b. Set the U-Boot environment variable to set the device tree that enables IMX 219 `imx8mp-evk-imx219.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-imx219.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

   As a result, the board is rebooted with the IMX 219 sensor enabled. At this point, the board is ready to run the selected camera module.

For information on connecting the IMX 219 camera module to i.MX 8M Plus EVK board via the XRPI-CAM-MINISAS adaptor board, see Section 4.3.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

4.2 Enable IMX 219 using precompiled binaries

1. Download the software pack binaries for the required BSP and extract its contents:
   - LF6.6.3_P24.1-Binaries
   - LF6.6.52_P24.4-Binaries
2. Perform Prerequistie for Use precompiled binaries.
3. Copy the binaries directory for the selected camera module onto the i.MX 8M Plus EVK board.
4. The following set of commands are executed on the i.MX 8M Plus board using a serial console terminal:
   a. Once the binaries directory has been copied onto the board, give execution permission to the `copy_binaries.sh` script and run the script. This script copies the binaries to the required locations on the board.

```
root@imx8mp-lpddr4-evk:~# cd sony_imx219_sensor\i.MX8MPLUS\Binaries\
root@imx8mp-lpddr4-evk:~# chmod +x copy_binaries.sh
root@imx8mp-lpddr4-evk:~#  ./copy_binaries.sh
```

   b. Reboot the board and press any key on your keyboard when you see the prompt shown in Figure 13 on your terminal to pause U-Boot execution.

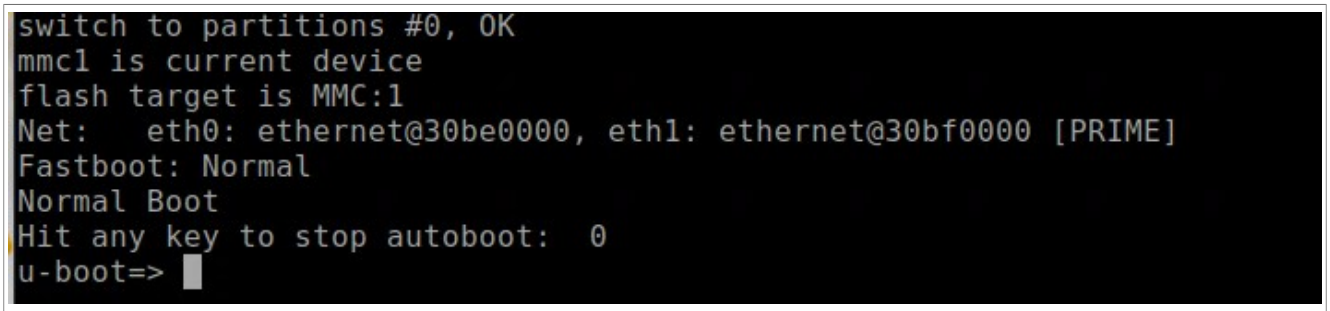


Figure 13. U-Boot prompt

   c. Set the U-Boot environment variable to set the device tree to enable IMX 219 `imx8mp-evk-imx219.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-imx219.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

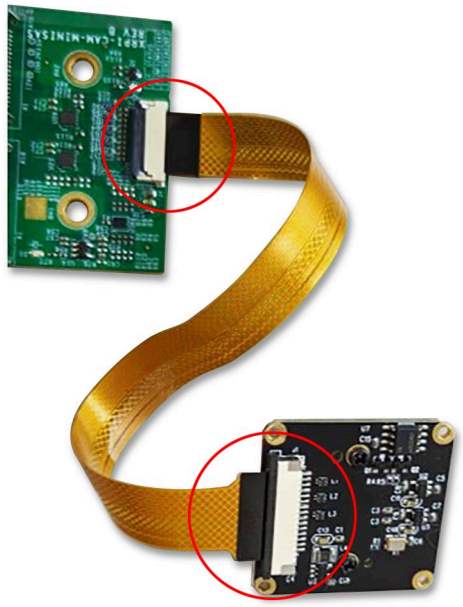
   As a result, the board is rebooted with the selected sensor enabled. At this point, the board is fully enabled to run the selected camera module.

4.3 Hardware connections for IMX 219 camera module

1. Power off the i.MX 8M Plus EVK board before connecting the camera module.
2. To connect the XRPI-CAM-MINISAS to the IMX 219 camera connector, use the 22 pin (0.5 mm pitch) to 15 pin (1 mm pitch) FPC cable. Match the side of the cable marked as black, as shown in Figure 14.

All information provided in this document is subject to legal disclaimers.

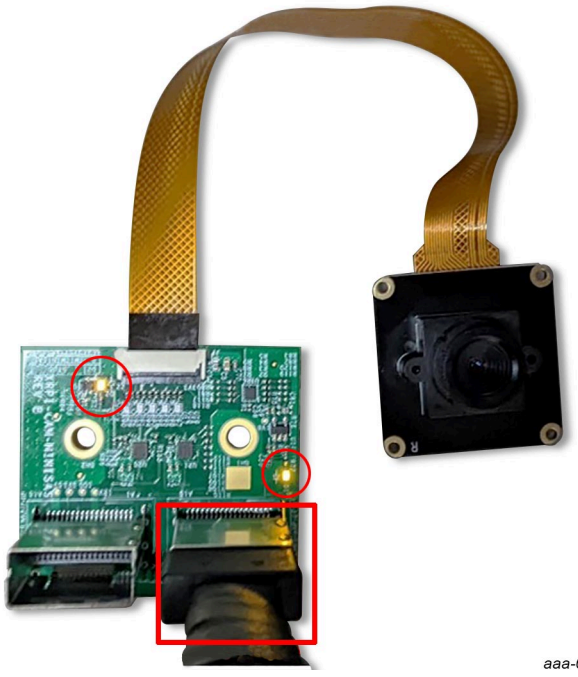
© 2025 NXP B.V. All rights reserved.



aaa-057197

Figure 14. IMX 219 connection with XRPI-CAM-MINISAS/RPI-CAM-MIPI board

3. Currently, only the highlighted MINISAS connector on the XRPI-CAM-MINISAS adaptor board shown in Figure 15 can be used on the XRPI-CAM-MINISAS adaptor.



aaa-057198

Figure 15. XRPI-CAM-MINISAS adaptor board powered on with IMX 219

4. Use the highlighted MIPI-CSI 1 port to connect the i.MX 8M Plus EVK board with the XRPI-CAM-MINISAS adaptor board, as shown in Figure 16.
5. Connect using the highlighted HDMI connector on the i.MX 8M Plus EVK board to connect to an external display.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6. Power on the board. When powered on, the two highlighted LEDs must turn on ensuring proper connection with the camera module.

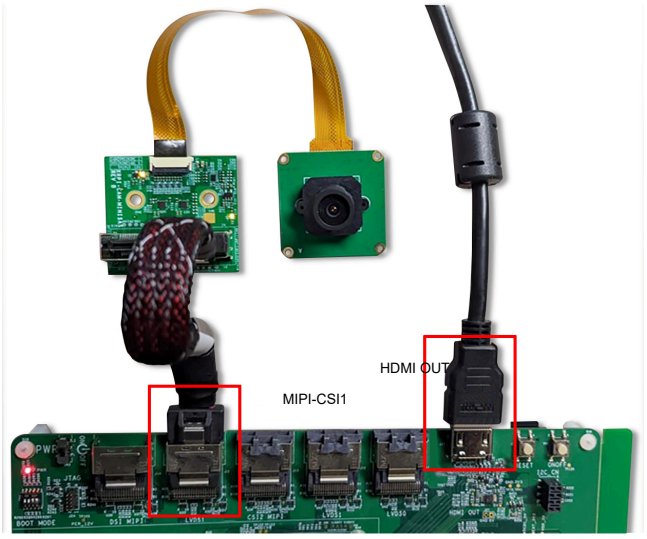


Figure 16. IMX 219 XRPI-CAM-MINISAS connection to MIPI-CSI 1 on i.MX 8M Plus EVK board

4.4 Test IMX 219 camera module

After connecting the camera module, power on the i.MX 8M Plus EVK board.

To test camera output on the external display, run the following commands:

- To test the IMX 219 camera module, run the following command:

```
root@imx8mp-lpddr4-evk:~# media-ctl -p
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

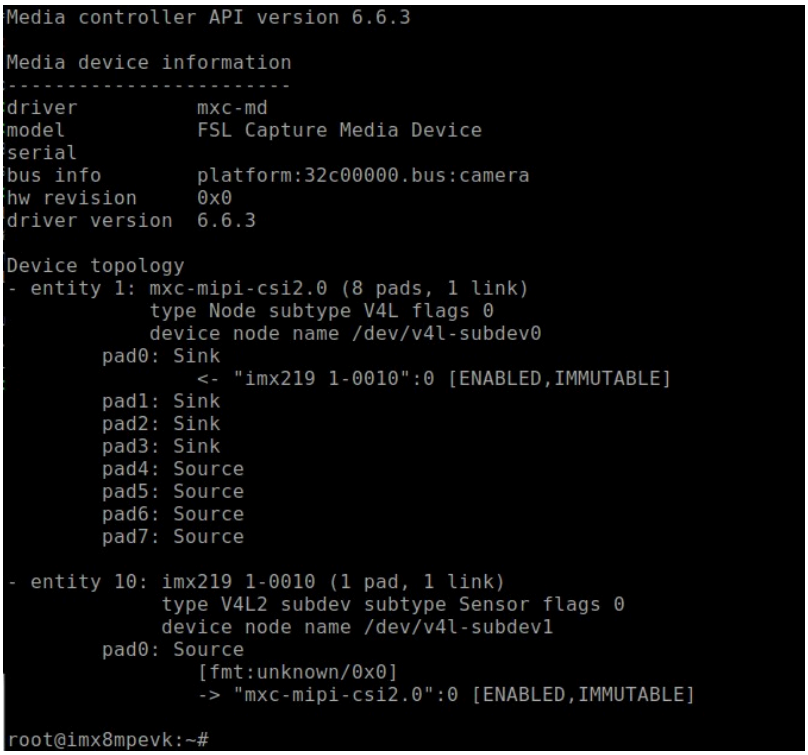


Figure 17. IMX 219 camera sensor is detected

- Identify v4l2 capture device ID for `VIV (platform:viv0)`:

```
root@imx8mp-lpddr4-evk:~# v4l2-ctl --list-device
```

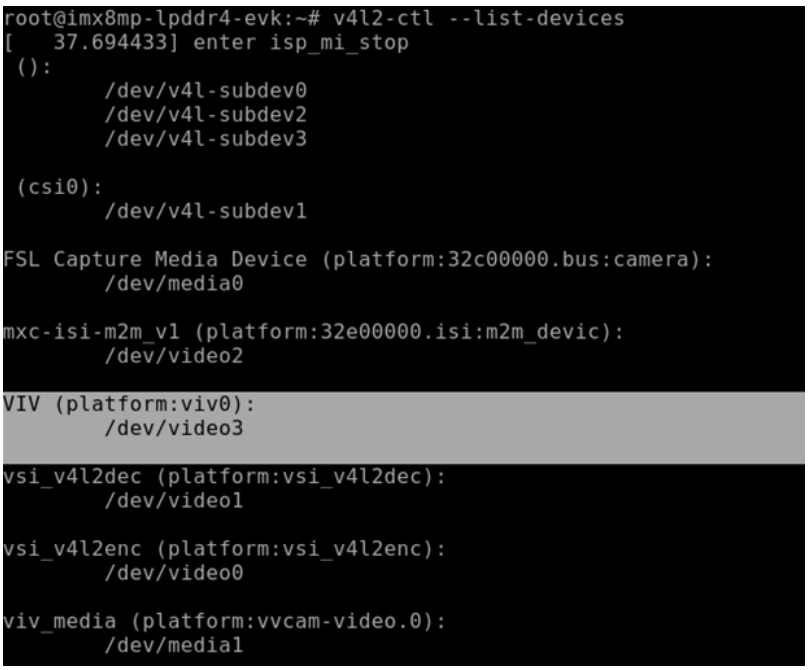


Figure 18. IMX 219 v4l2 capture device

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- Use `v4l2src device=</dev/video#>` as the identified v4l2 capture device ID:

```
root@imx8mp-lpddr4-evk:~# gst-launch-1.0 v4l2src device=/dev/video3 ! "video/x-raw ,format=YUY2, width=1920, height=1080" ! waylandsink
```

- Running the above Gstreamer command sets up pipeline to use the v4l2 API to use the camera device on `/dev/video3`, with YUV2 format at a resolution of 1920 x 1080 and use waylandsink as the display output.

Figure 19. IMX 219 capture output

5 Enable OmniVision OV5647 camera module

This section describes the two ways to enable the OmniVision OV5647 camera module and the steps to connect it with the i.MX 8M Plus EVK board via the XRPI-CAM-MINISAS adaptor board:

- Build an i.MX 8M Plus EVK Yocto image with OmniVision OV5647 camera module enabled
- Enable OV5647 using pre-compiled binaries

5.1 Build an i.MX 8M Plus EVK Yocto image with OmniVision OV5647 camera module enabled

1. Perform Prerequisite for Local build.
2. To set up a software pack repo, perform the following steps:
   a. Clone and checkout the required BSP branch:
      - LF6.6.3_P24.1
      - LF6.6.52_P24.4

```
$ git clone https://github.com/nxp-imx-support/imx-camera-sw-pack.git
$ git checkout LF6.6.3_P24.1
//Or
$ git checkout LF6.6.52_P24.4
```

   b. Copy only `meta-imx8mp-isp-ov5647` into the `sources` directory in your `Yocto_Directory`:

```
$ cp -r  imx-camera-sw-pack/ov_ov5647_sensor/i.MX8MPLUS/meta-imx8mp-isp-ov5647  imx-yocto-bsp/sources/
```

3. To build and flash the image enabled with the OV5647 camera module, perform the following steps:
   a. Navigate to the `Yocto_Directory`:

```
$ cd  imx-yocto-bsp
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

b. Source the setup script to configure Yocto build system to enable the OV5647 camera module and build the image:

```
$ source sources/meta-imx8mp-isp-ov5647/setup/setup-env-imx8mp-ov5647 -b
 build
$ bitbake imx-image-full
```

c. After the build process is completed, the image file `imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst` is available at the following location:

```
$ imx-yocto-bsp/build/tmp/deploy/images/imx8mp-lpddr4-evk/ imx-image-full-
imx8mp-lpddr4-evk.rootfs.wic.zst
```

d. Flash this image to the SD card:

```
$ zstdcat imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst | sudo dd of=/
dev/sd<partition> bs=1M conv=fsync
```

4. The following set of commands are executed on the i.MX 8M Plus EVK board using a serial console terminal:
   a. Now, boot up the i.MX 8M Plus EVK board with the serial console terminal open.
   While the board is booting up, press any key on the keyboard when you see the prompt shown in Figure 20 on your terminal to pause U-Boot execution.

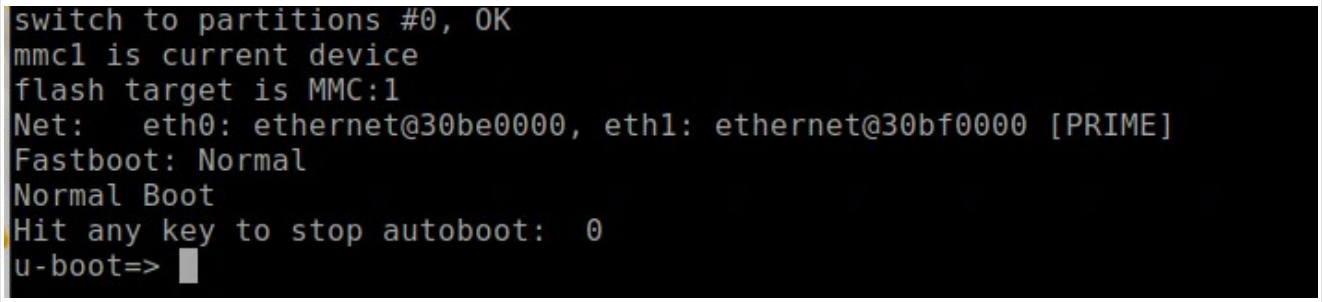


Figure 20. U-Boot prompt

   b. Set the U-Boot environment variable to set the device tree that enables OV5647 `imx8mp-evk-ov5647.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-ov5647.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

   As a result, the board with the OV5647 sensor enabled is rebooted. At this point, the board is fully enabled to run the selected camera module.

For information on connecting the OV5647 camera module to i.MX 8M Plus EVK board via the xRPI-CAM-MINISAS adaptor board, see Section 5.3.

5.2 Enable OV5647 using precompiled binaries

1. Download the software pack binaries for the required BSP and extract its contents:
   - LF6.6.3_P24.1-Binaries
   - LF6.6.52_P24.4-Binaries
2. Perform Prerequistie for Use precompiled binaries.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

3. Copy the binaries directory for the selected camera module onto the i.MX 8M Plus EVK board.
4. The following set of commands are executed on the i.MX 8M Plus EVK board using a serial console terminal:
   a. Once the binaries directory has been copied on the board, give execution permission to the `copy_binaries.sh` script and run the script. This script copies the binaries to the required locations on the board.

```
root@imx8mp-lpddr4-evk:~# cd ov_ov5647_sensor\i.MX8MPLUS\Binaries\
root@imx8mp-lpddr4-evk:~#  chmod +x copy_binaries.sh
root@imx8mp-lpddr4-evk:~#  ./copy_binaries.sh
```

   b. Reboot the board and press any key on your keyboard when you see the prompt shown in Figure 21 on your terminal to pause U-Boot execution.

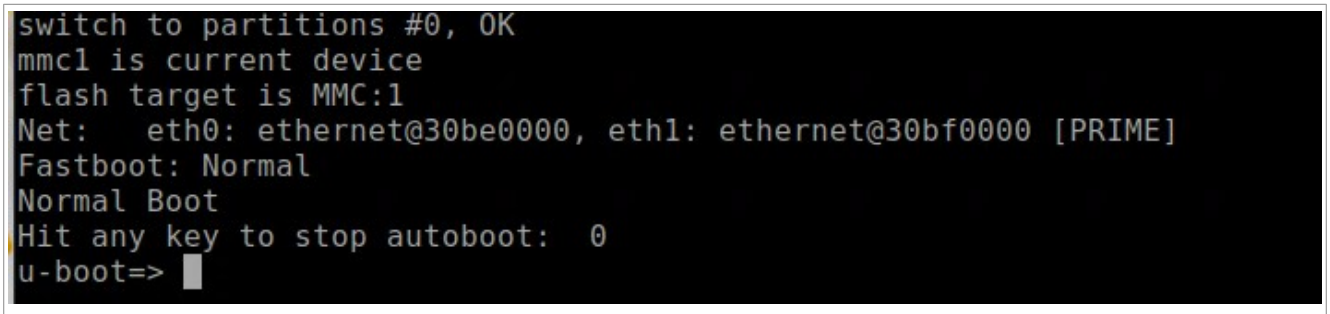


Figure 21. U-Boot prompt

   c. Set the U-Boot environment variable to set the device tree to enable OV5647 `imx8mp-evk-ov5647.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-ov5647.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

   As a result, the board is rebooted with the selected sensor enabled. At this point, the board is fully enabled to run the selected camera module.

5.3 Hardware connections for OV5647 camera module

1. Power off the i.MX 8M Plus EVK board before connecting the camera module.
2. To connect the XRPI-CAM-MINISAS to the OV5647 camera connector, use the 22 pin (0.5 mm pitch) to 15 pin (1 mm pitch) FPC cable. Match the side of the cable marked as black, as shown in Figure 22.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

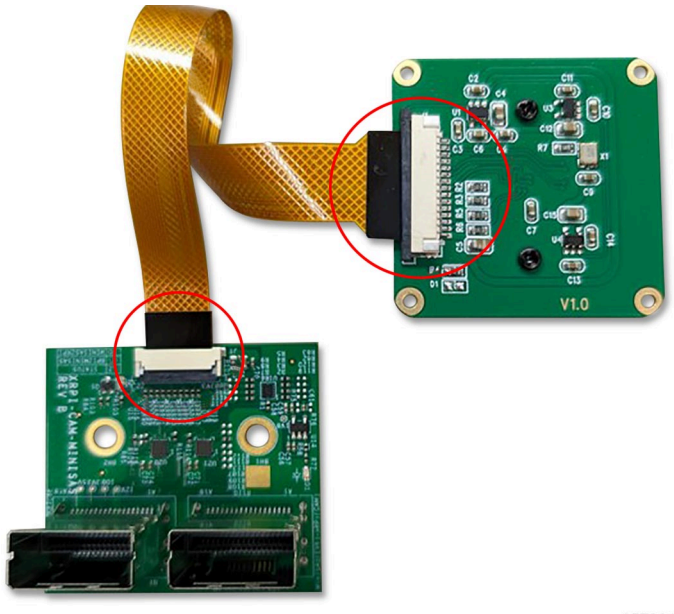


Figure 22. OV5647 connection with XRPI-CAM-MINISAS/RPI-CAM-MIPI board

3. Currently, only the highlighted MINISAS connector on the XRPI-CAM-MINISAS adaptor board shown in Figure 23 can be used on the XRPI-CAM-MINISAS adaptor.

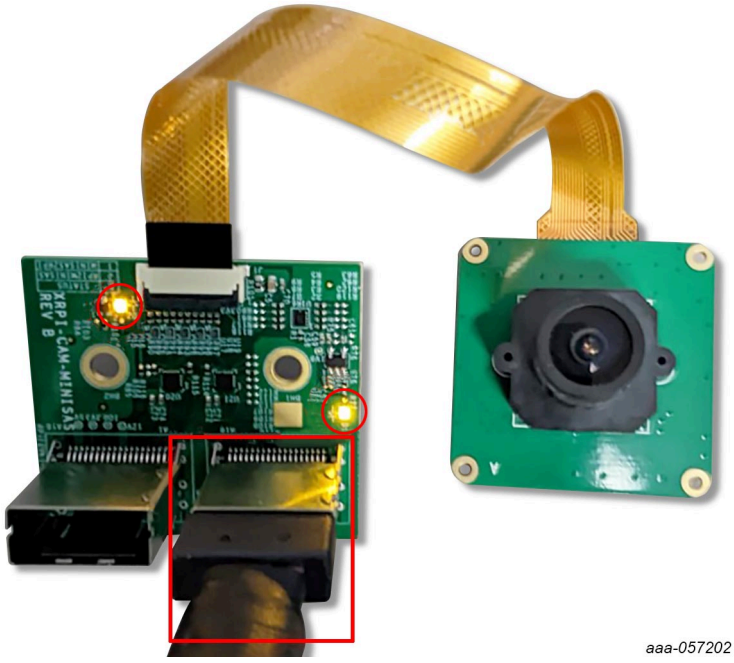


Figure 23. XRPI-CAM-MINISAS adaptor board powered on with OV5647

4. Use the highlighted MIPI-CSI 1 port to connect the i.MX 8M Plus EVK board with the XRPI-CAM-MINISAS adaptor board.
5. Connect using the highlighted HDMI connector on the i.MX 8M Plus EVK board to connect to an external display.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6. Power on the board. When powered on, the two highlighted LEDs must turn on ensuring proper connection with the camera module.

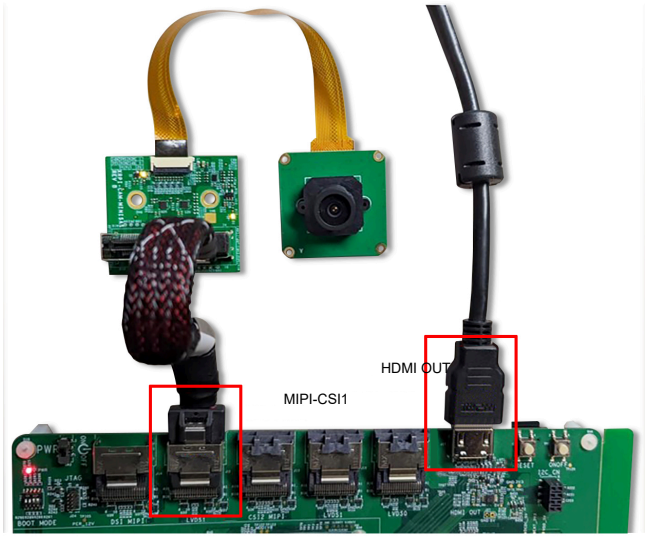


Figure 24. OV5647 XRPI-CAM-MINISAS connection to MIPI-CSI 1 on i.MX 8M Plus EVK board

5.4 Test OV5647 camera module

After connecting the camera module, power on the i.MX 8M Plus EVK board.

To test camera output on the external display, run the following commands:

- To test the OV5647 camera module, run the following command:

```
root@imx8mp-lpddr4-evk:~# media-ctl -p
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

```
root@imx8mpevk:~# media-ctl -p
Media controller API version 6.6.3

Media device information
------------------------
driver          mxc-md
model           FSL Capture Media Device
serial
bus info        platform:32c00000.bus:camera
hw revision     0x0
driver version  6.6.3

Device topology
- entity 1: mxc-mipi-csi2.0 (8 pads, 1 link)
            type Node subtype V4L flags 0
            device node name /dev/v4l-subdev0
        pad0: Sink
                <- "ov5647 1-0036":0 [ENABLED,IMMUTABLE]
        pad1: Sink
        pad2: Sink
        pad3: Sink
        pad4: Source
        pad5: Source
        pad6: Source
        pad7: Source

- entity 10: ov5647 1-0036 (1 pad, 1 link)
             type V4L2 subdev subtype Sensor flags 0
             device node name /dev/v4l-subdev1
        pad0: Source
                [fmt:SBGGR10_1X10/1920x1080 field:none]
                -> "mxc-mipi-csi2.0":0 [ENABLED,IMMUTABLE]
```

Figure 25. OV5647 camera sensor is detected

- Identify v4l2 capture device ID for `VIV (platform:viv0)`:

```
root@imx8mp-lpddr4-evk:~# v4l2-ctl --list-device
```

```
root@imx8mp-lpddr4-evk:~# v4l2-ctl --list-devices
[  450.219247] enter isp_mi_stop
 ():
        /dev/v4l-subdev0
        /dev/v4l-subdev2
        /dev/v4l-subdev3

 (csi0):
        /dev/v4l-subdev1

FSL Capture Media Device (platform:32c00000.bus:camera):
        /dev/media0

mxc-isi-m2m_v1 (platform:32e00000.isi:m2m_devic):
        /dev/video2

VIV (platform:viv0):
        /dev/video3

vsi_v4l2dec (platform:vsi_v4l2dec):
        /dev/video1

vsi_v4l2enc (platform:vsi_v4l2enc):
        /dev/video0

viv_media (platform:vvcam-video.0):
        /dev/media1
```

Figure 26. OV5647 v4l2 capture device

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- Use `v4l2src device=</dev/video#>` as the identified v4l2 capture-device ID:

```
root@imx8mp-lpddr4-evk:~# gst-launch-1.0 v4l2src device=/dev/video3 ! "video/x-raw ,format=YUY2, width=1920, height=1080" ! waylandsink
```

- Running the above Gstreamer command sets up pipeline to use the v4l2 API to use the camera device on `/dev/video3`, with YUV2 format at a resolution of 1920 x 1080 and use waylandsink as the display output.

Figure 27. OV5647 capture output

6 Enable ON Semiconductor AR0144 camera module

This section describes the two ways to enable the ON Semiconductor (OnSemi) AR0144 camera module and the steps to connect it with the i.MX 8M Plus EVK board via the XRPI-CAM-MINISAS adaptor board:

- Build an i.MX 8M Plus EVK Yocto image with OnSemi AR0144 camera module enabled
- Enable AR0144 using pre-compiled binaries

6.1 Build an i.MX 8M Plus EVK Yocto image with ONSemi AR0144 camera module enabled

1. Perform the Prerequisite for Local build.
2. To set up a software pack repo, perform the following steps:
   a. Clone and checkout the required BSP branch:
      - LF6.6.3_P24.1
      - LF6.6.52_P24.4

```
$ git clone https://github.com/nxp-imx-support/imx-camera-sw-pack.git
$ git checkout LF6.6.3_P24.1
//Or
$ git checkout LF6.6.52_P24.4
```

   b. Copy only `meta-imx8mp-isp-ar0144` into the `sources` directory in your `Yocto_Directory`:

```
$ cp -r  camera-sw-pack/onsemi_ar0144_sensor/i.MX8MPLUS/meta-imx8mp-isp-imx219  imx-yocto-bsp/sources
```

3. To build and flash the image enabled with the AR0144 camera module, perform the following steps:
   a. Navigate to the `Yocto_Directory`:

```
$ cd  imx-yocto-bsp
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

b. Source the setup script to configure Yocto build system to enable the AR0144 camera module and build the image:

```
$ source sources/meta-imx8mp-isp-ar0144/setup/setup-env-imx8mp-ar0144 -b
 build
$ bitbake imx-image-full
```

c. After the build process is completed, the image file `imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst` is available at the following location:

```
$ imx-yocto-bsp/build/tmp/deploy/images/imx8mp-lpddr4-evk/ imx-image-full-
imx8mp-lpddr4-evk.rootfs.wic.zst
```

d. Flash this image to the SD card:

```
$ zstdcat imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst | sudo dd of=/
dev/sd<partition> bs=1M conv=fsync
```

4. The following set of commands are executed on the i.MX 8M Plus EVK board using a serial console terminal:
   a. Now, boot up the i.MX 8M Plus EVK board with the serial console terminal open.
   While the board is booting up, press any key on the keyboard when you see the prompt shown in Figure 28 on your terminal to pause U-Boot execution.

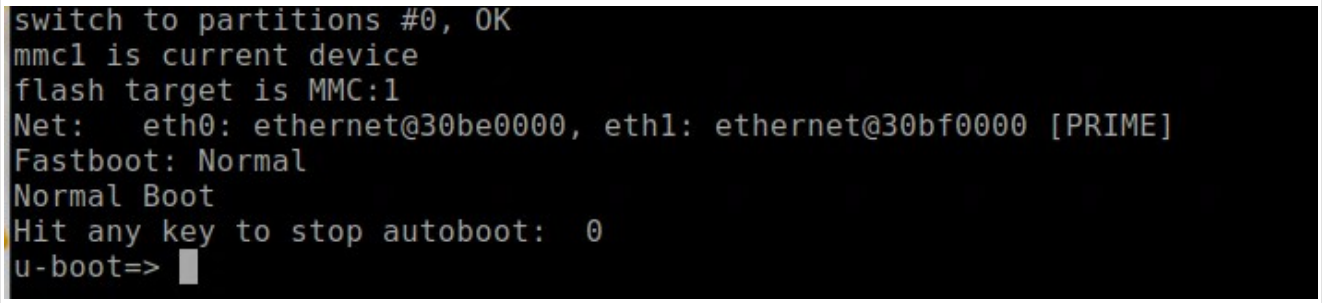


Figure 28. U-Boot prompt

   b. Set the U-Boot environment variable to set the device tree that enables AR0144 `imx8mp-evk-ar0144.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-ar0144.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

   As a result, the board with the AR0144 sensor enabled is rebooted. At this point, the board is fully enabled to run the selected camera module.

For information on connecting the AR0144 camera module to i.MX 8M Plus EVK board via the XRPI-CAM-MINISAS adaptor board, see Section 6.3.

6.2 Enable AR0144 using precompiled binaries

1. Download the software pack binaries for the required BSP and extract its contents:
   - LF6.6.3_P24.1-Binaries
   - LF6.6.52_P24.4-Binaries
2. Perform Prerequistie for Use precompiled binaries.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

3. Copy the binaries directory for the selected camera module onto the i.MX 8M Plus EVK board.
4. The following set of commands are executed on the i.MX 8M Plus EVK board using a serial console terminal:
   a. Once the binaries directory has been copied onto the board, give execution permission to the `copy_binaries.sh` script and run the script. This script copies the binaries to the required locations on the board.

```
root@imx8mp-lpddr4-evk:~# cd onsemi_ar0144_sensor\i.MX8MPLUS\Binaries\
root@imx8mp-lpddr4-evk:~#  chmod +x copy_binaries.sh
root@imx8mp-lpddr4-evk:~#  ./copy_binaries.sh
```

   b. Reboot the board and press any key on your keyboard when you see the prompt shown in Figure 29 on your terminal to pause U-Boot execution.

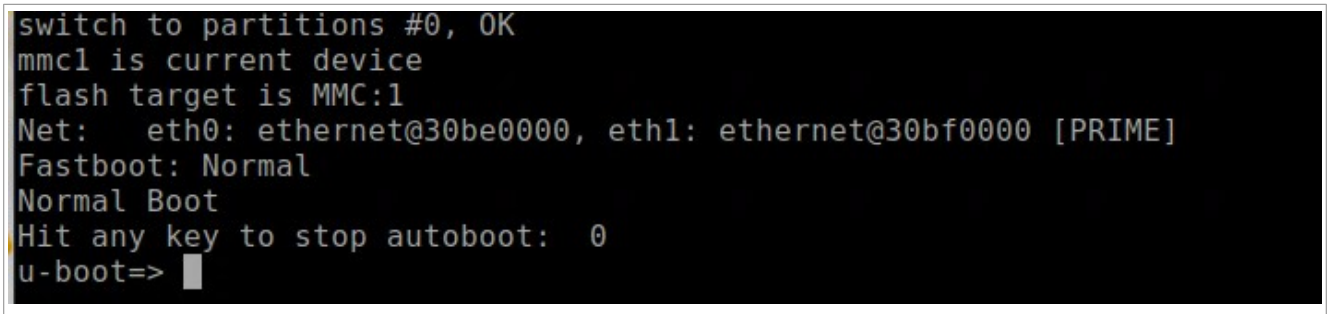


Figure 29. U-Boot prompt

   c. Set the U-Boot environment variable to set the device tree to enable AR0144 `imx8mp-evk-ar0144.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-ar0144.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

   Asa result, the board is rebooted with the selected sensor enabled. At this point, the board is fully enabled to run the selected camera module.

For information on connecting the AR1044 camera module to i.MX 8M Plus EVK Board via the xRPI-CAM-MINISAS adaptor board, see Section 6.3.

6.3 Hardware connections for OnSemi AR0144 camera module

1. Power off the i.MX 8M Plus EVK board before connecting the camera module.
2. To connect the XRPI-CAM-MINISAS to the AR0144 camera connector, use the 22 pin (0.5 mm pitch) to 15 pin (1 mm pitch) FPC cable, as shown in Figure 30.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

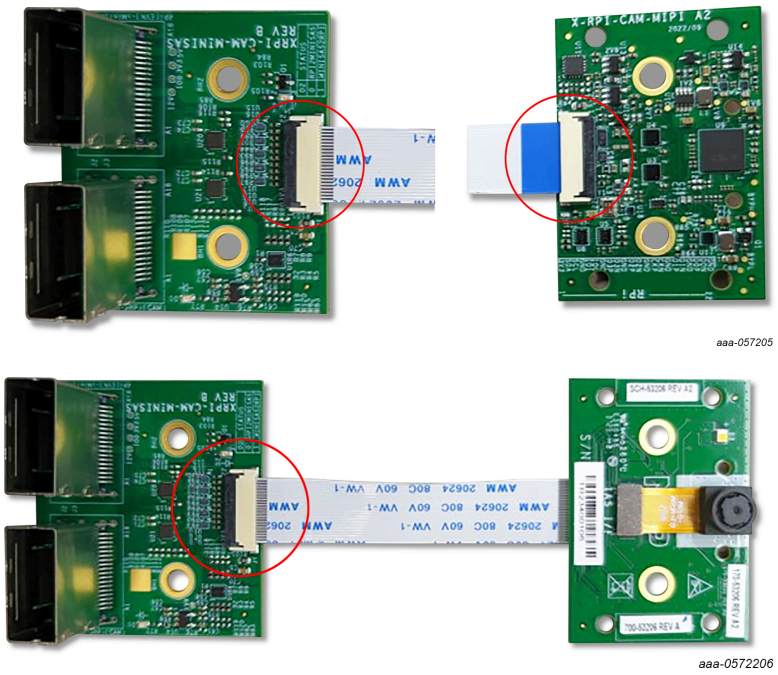


Figure 30. AR0144 connection with XRPI-CAM-MINISAS/RPI-CAM-MIPI board

3. Currently, only the highlighted MINISAS connector on the XRPI-CAM-MINISAS adaptor board shown in Figure 31 can be used on the XRPI-CAM-MINISAS adaptor.

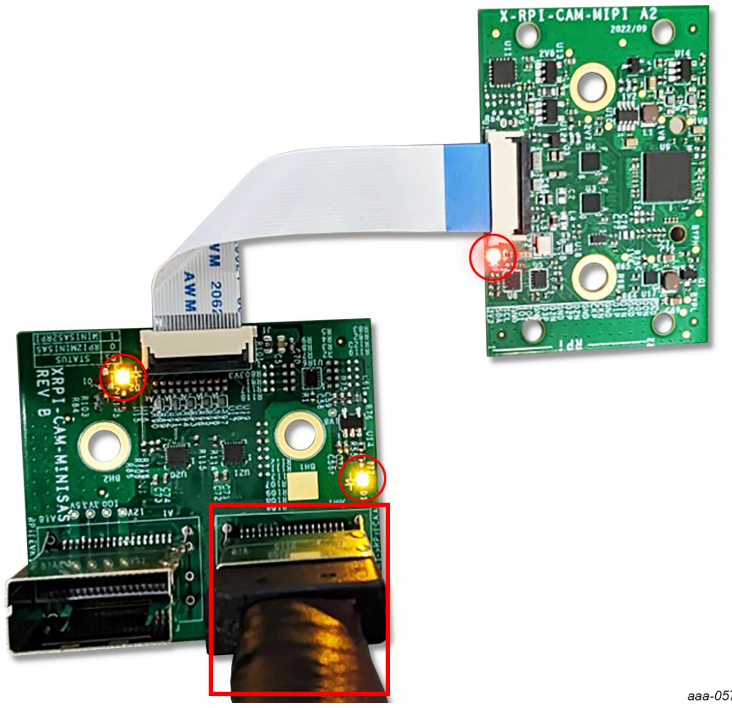


Figure 31. XRPI-CAM-MINISAS adaptor board powered on with AR0144

4. Use the highlighted MIPI-CSI 1 port to connect the i.MX 8M Plus EVK board with the XRPI-CAM-MINISAS adaptor board.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5. Connect using the highlighted HDMI connector on the i.MX 8M Plus EVK board to connect to an external display.
6. Power on the board. When powered on, the two highlighted LEDs must turn on ensuring proper connection with the camera module.

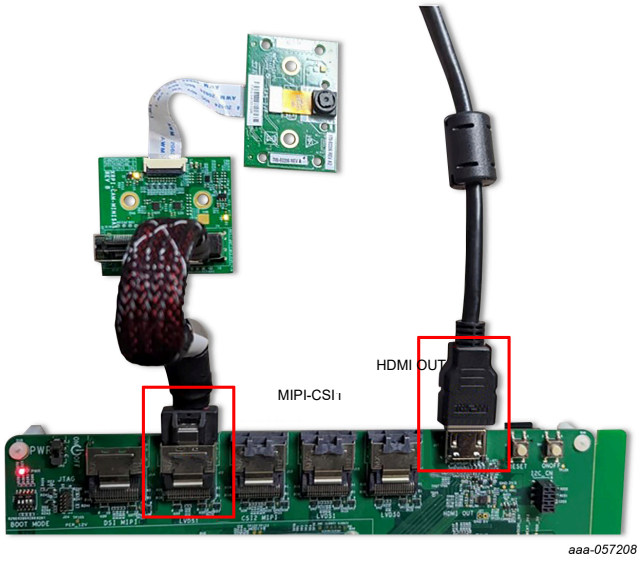


Figure 32. AR0144 XRPI-CAM-MINISAS connection to MIPI-CSI 1 on i.MX 8M Plus EVK board

6.4 Test AR0144 camera module

After connecting the camera module, power on the i.MX 8M Plus EVK board.

To test camera output on the external display, run the following commands:

- To test the AR0144 camera module. run the following command:

```
root@imx8mp-lpddr4-evk:~# media-ctl -p
```

```
root@imx8mpevk:~# media-ctl -p
Media controller API version 6.6.[  481.112937] ar0144 1-0010: get_fmt: code=0x3010, wxh=1280x
800
3

Media device information
------------------------
driver          mxc-md
model           FSL Capture Media Device
serial
bus info        platform:32c00000.bus:camera
hw revision     0x0
driver version  6.6.3

Device topology
- entity 1: mxc-mipi-csi2.0 (8 pads, 1 link)
            type Node subtype V4L flags 0
            device node name /dev/v4l-subdev0
        pad0: Sink
                <- "ar0144 1-0010":0 [ENABLED,IMMUTABLE]
        pad1: Sink
        pad2: Sink
        pad3: Sink
        pad4: Source
        pad5: Source
        pad6: Source
        pad7: Source

- entity 10: ar0144 1-0010 (1 pad, 1 link)
             type V4L2 subdev subtype Sensor flags 0
             device node name /dev/v4l-subdev1
        pad0: Source
                [fmt:SGBRG12_1X12/1280x800 field:none]
                -> "mxc-mipi-csi2.0":0 [ENABLED,IMMUTABLE]
```

Figure 33. AR0144 camera sensor is detected

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- Identify v4l2 capture device ID for `VIV (platform:viv0)`:

```
root@imx8mp-lpddr4-evk:~# v4l2-ctl --list-device
```

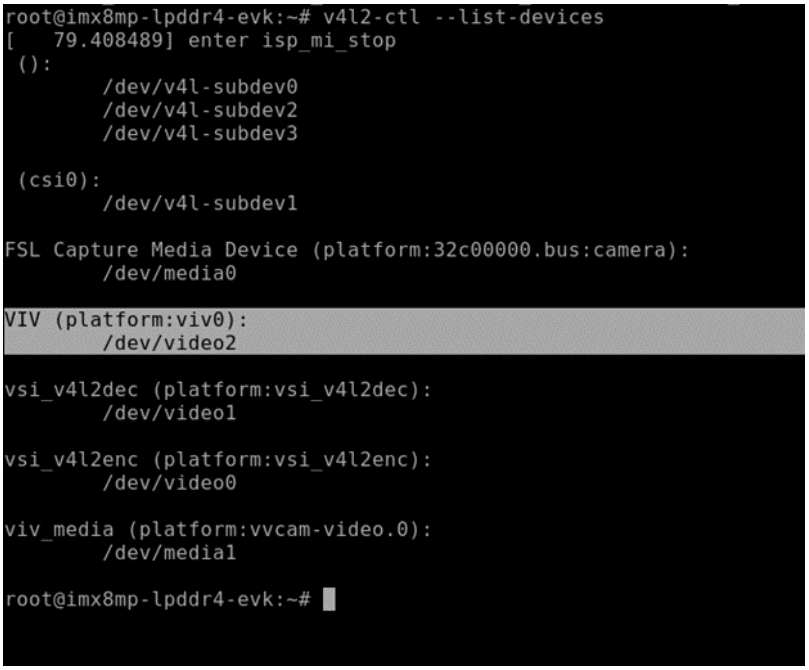


Figure 34. AR0144 v4l2 capture device

- Use `v4l2src device=</dev/video#>` as the identified v4l2 capture-device ID:

```
root@imx8mp-lpddr4-evk:~# gst-launch-1.0 -v v4l2src device=/dev/video2 !
 "video/x-raw,format=YUY2,width=1280,height=800" ! queue ! waylandsink
```

- Running the above Gstreamer command sets up pipeline to use the v4l2 API to use the camera device on `/dev/video2`, with YUV2 format at a resolution of 1920 x 1080 and use waylandsink as the display output.

Figure 35. Monochrome capture output

7 Enable ON Semiconductor AR0830 camera module

This section describes two ways to enable the ON Semiconductor (OnSemi) AR0830 camera module and steps to connect it with the i.MX 8M Plus EVK board:

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

7.1 Build an i.MX 8M Plus EVK Yocto image with ONSemi AR0830 camera module enabled

1. Perform the Prerequisite for Local build.
2. To set up a software pack repo, perform the following steps:
   a. Clone and checkout the required BSP branch:
      - LF6.6.3_P24.1
      - LF6.6.52_P24.4

```
$ git clone https://github.com/nxp-imx-support/imx-camera-sw-pack.git
$ git checkout LF6.6.3_P24.1
//Or
$ git checkout LF6.6.52_P24.4
```

   b. Copy only `meta-imx8mp-isp-ar0830` into the `sources` directory in your `Yocto_Directory`:

```
$ cp -r camera-sw-pack/onsemi_ar0830_sensor/i.MX8MPLUS/meta-imx8mp-isp-
 imx219 imx-yocto-bsp/sources
```

3. To build and flash the image enabled with the AR0830 camera module, perform the following steps:
   a. Navigate to the `Yocto_Directory`:

```
$ cd imx-yocto-bsp
```

   b. Source the setup script to configure Yocto build system to enable the AR0830 camera module and build the image:

```
$ source sources/meta-imx8mp-isp-ar0830/setup/setup-env-imx8mp-ar0144 -b
 build
$ bitbake imx-image-full
```

   c. After the build process is completed, the image file `imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst` is available at the following location:

```
$ imx-yocto-bsp/build/tmp/deploy/images/imx8mp-lpddr4-evk/ imx-image-full-
 imx8mp-lpddr4-evk.rootfs.wic.zst
```

   d. Flash this image to the SD card:

```
$ zstdcat imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst | sudo dd of=/
 dev/sd<partition> bs=1M conv=fsync
```

4. The following set of commands are executed on the i.MX 8M Plus EVK board using a serial console terminal:
   a. Now, boot up the i.MX 8M Plus EVK board with the serial console terminal open. While the board is booting up, press any key on the keyboard when you see the prompt shown in Figure 36 on your terminal to pause U-Boot execution.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

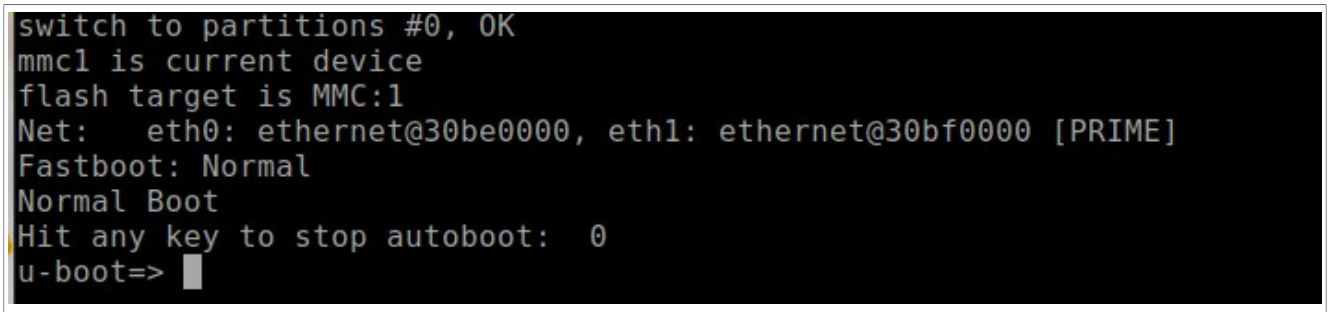


Figure 36. U-Boot prompt

b. Set the U-Boot environment variable to set the device tree that enables AR0830 `imx8mp-evk-ar0830.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-ar0830.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

As a result, the board with the AR0830 sensor enabled is rebooted. At this point, the board is fully enabled to run the selected camera module.

For information on connecting the AR0830 camera module to i.MX 8M Plus EVK board, see Section 7.3.

7.2 Enable AR0830 using precompiled binaries

1. Download the software pack binaries for the required BSP and extract its contents:
   - LF6.6.3_P24.1-Binaries
   - LF6.6.52_P24.4-Binaries
2. Perform Prerequistie for Use precompiled binaries.
3. Copy the binaries directory for the selected camera module onto the i.MX 8M Plus EVK board.
4. The following set of commands are executed on the i.MX 8M Plus EVK board using a serial console terminal:
   a. Once the binaries directory has been copied onto the board, give execution permission to the `copy_binaries.sh` script and run the script. This script copies the binaries to the required locations on the board:

```
root@imx8mp-lpddr4-evk:~# cd onsemi_ar0830_sensor\i.MX8MPLUS\Binaries\
root@imx8mp-lpddr4-evk:~# chmod +x copy_binaries.sh
root@imx8mp-lpddr4-evk:~# ./copy_binaries.sh
```

   b. Reboot the board and press any key on your keyboard when you see the prompt shown in Figure 37 on your terminal to pause U-Boot execution.

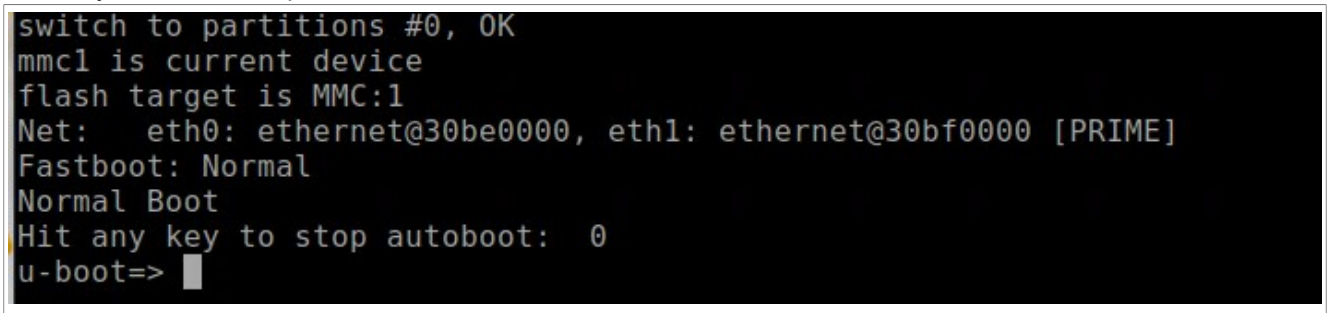


Figure 37. U-Boot prompt

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

c. Set the U-Boot environment variable to set the device tree to enable AR0830 `imx8mp-evk-ar0830.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-ar0830.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

As a result, the board is rebooted with the selected sensor enabled. At this point, the board is fully enabled to run the selected camera module.

7.3 Hardware connections for OnSemi AR0830 camera module

1. Power off the i.MX 8M Plus EVK board before connecting the camera module.
2. Figure 38 shows the AR0830 PRISM module connected on PRISM/IAS to NXP i.MX 8MP EVK adapter board. Set the ISP/Bypass jumpers, as shown in Figure 38.

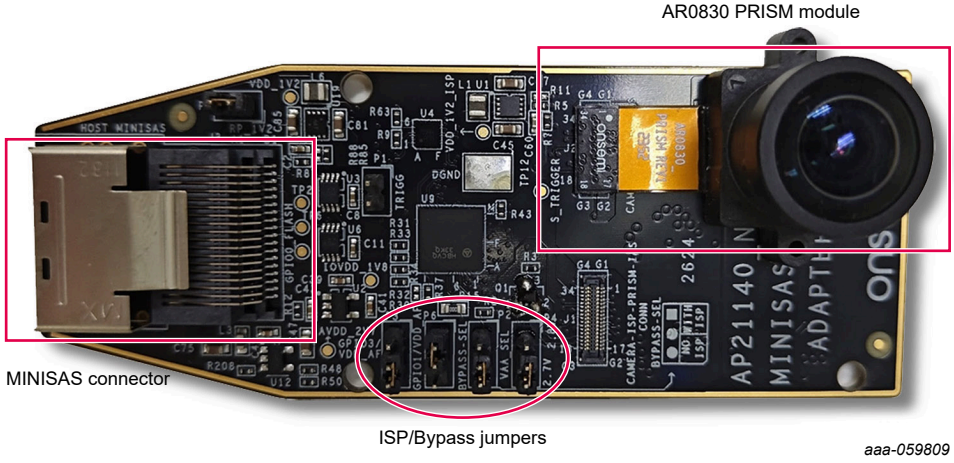


Figure 38. PRISM/IAS to NXP i.MX 8MP EVK adapter board with AR0830 PRISM module attached

3. Use the highlighted MIPI-CSI 1 port to connect the i.MX 8M Plus EVK board with the adapter board in Figure 39.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

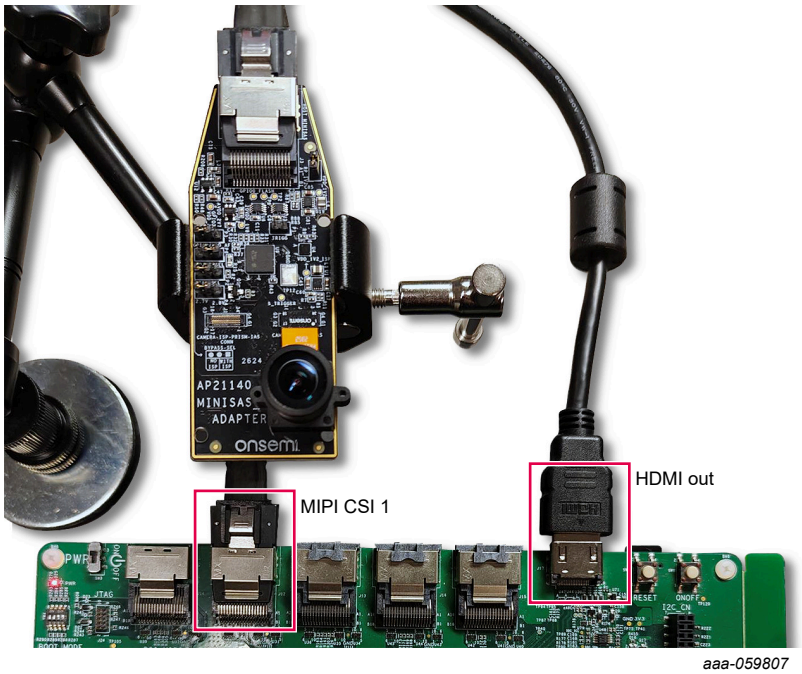


Figure 39. AR0830 adaptor connected on MIPI-CSI 1 on i.MX 8M Plus EVK board

4. Connect using the highlighted HDMI connector on the i.MX 8M Plus EVK board to connect to an external display.
5. Power on the board.

7.4 Test AR0830 camera module

After connecting the camera module, power on the i.MX 8M Plus EVK board.

- To test camera output on the external display, run the following commands:

```
root@imx8mp-lpddr4-evk:~# media-ctl -p
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

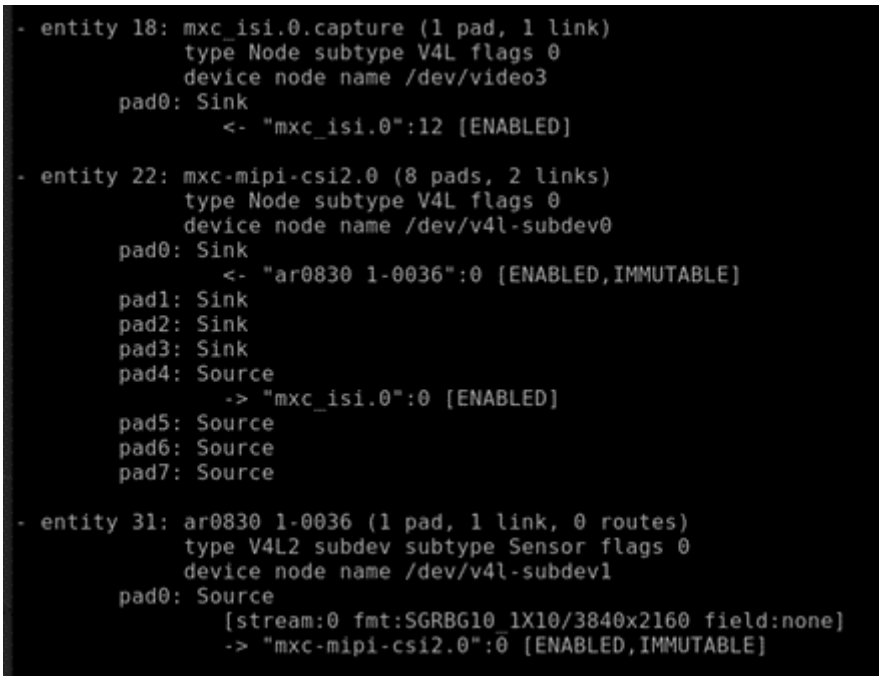


Figure 40. AR0830 camera sensor is detected

- Identify v4l2 capture device ID for `VIV (platform:viv0)`:

```
root@imx8mp-lpddr4-evk:~# v4l2-ctl --list-device
```

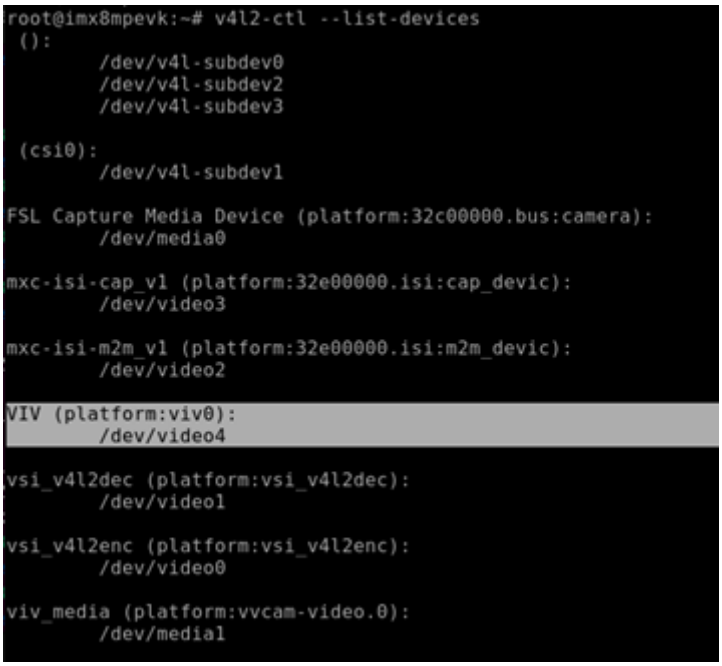


Figure 41. AR0830 v4l2 capture device

- Use `v4l2src device=</dev/video#>` as the identified v4l2 capture-device ID:

```
root@imx8mp-lpddr4-evk:~# gst-launch-1.0 -v v4l2src device=/dev/video4 ! video/
x-raw, format=YUY2, width=3840, height=2160  ! waylandsink
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- Running the above Gstreamer command sets up pipeline to use the v4l2 API to use the camera device on `/dev/video4`, with YUV2 format at a resolution of 3840 x 2160 and use waylandsink as the display output.

aaa-059812

Figure 42. AR0830 camera output

8 Enable Sony IMX 258 camera module

This section describes the two ways to enable the IMX 258 camera module and steps to connect it with the i.MX 8M Plus EVK board:

- Build an i.MX 8M Plus EVK Yocto image with Sony IMX 258 camera module enabled
- Enable IMX 258 using precompiled binaries

8.1 Build an i.MX 8M Plus EVK Yocto image with Sony IMX 258 camera module enabled

1. Perform the Prerequisite for Local build.
2. To set up a software pack repo, perform the following steps:
   a. Clone and checkout the required BSP branch:
      - LF6.6.3_P24.1
      - LF6.6.52_P24.4

```
$ git clone https://github.com/nxp-imx-support/imx-camera-sw-pack.git
$ git checkout LF6.6.3_P24.1
//Or
$ git checkout LF6.6.52_P24.4
```

   b. Copy only `meta-imx8mp-isp-imx258` into the `sources` directory in your `Yocto_Directory`:

```
$ cp -r imx-camera-sw-pack/sony_imx258_sensor/i.MX8MPLUS/meta-imx8mp-isp-
 imx258 imx-yocto-bsp/sources/
```

3. To build and flash the image enabled with the IMX 258 camera module, perform the following steps:
   a. Navigate to the `Yocto_Directory`:

```
$ cd imx-yocto-bsp
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

b. Source the setup script to configure Yocto build system to enable the IMX 258 camera module and build the image:

```
$ source sources/meta-imx8mp-isp-imx258/setup/setup-env-imx8mp-imx258 -b
 build
$ bitbake imx-image-full
```

c. After the build process is completed, the image file `imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst` is available at the following location:

```
$ imx-yocto-bsp/build/tmp/deploy/images/imx8mp-lpddr4-evk/ imx-image-full-
 imx8mp-lpddr4-evk.rootfs.wic.zst
```

d. Flash this image to the SD card:

```
$ zstdcat imx-image-full-imx8mp-lpddr4-evk.rootfs.wic.zst | sudo dd of=/
 dev/sd<partition> bs=1M conv=fsync
```

4. The following set of commands are executed on the i.MX 8M Plus EVK board using a serial console terminal:
   a. Now, boot up the i.MX 8M Plus EVK board with the serial console terminal open. While the board is booting up, press any key on the keyboard when you see the prompt shown in Figure 43 on your terminal to pause U-Boot execution.

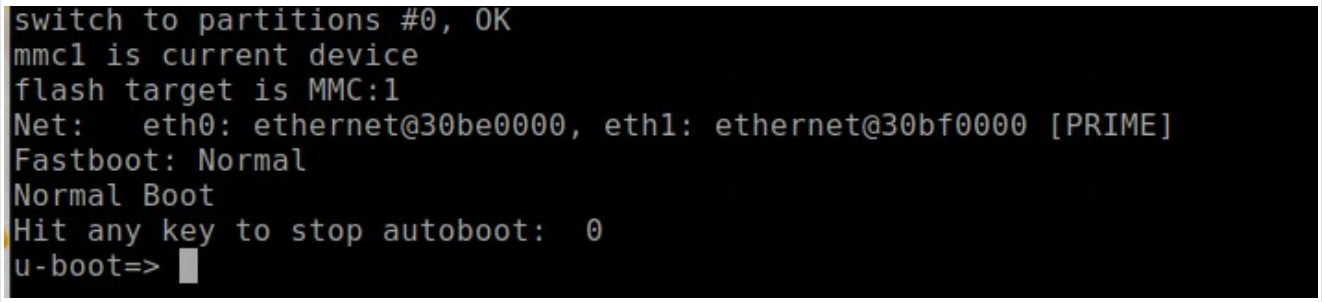


Figure 43. U-Boot prompt

   b. Set the U-Boot environment variable to set the device tree that enables IMX 258 `imx8mp-evk-imx219.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-imx258.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

   As a result, the board is rebooted with the IMX 258 sensor enabled. At this point, the board is ready to run the selected camera module.

For information on connecting the IMX 258 camera module to i.MX 8M Plus EVK board, see Section 8.3.

8.2 Enable IMX 258 using precompiled binaries

1. Download the software pack binaries for the required BSP and extract its contents:
   - LF6.6.3_P24.1-Binaries
   - LF6.6.52_P24.4-Binaries
2. Perform Prerequistie for Use precompiled binaries.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

3. Copy the binaries directory for the selected camera module onto the i.MX 8M Plus EVK board.
4. The following set of commands are executed on the i.MX 8M Plus EVK board using a serial console terminal:
   a. Once the binaries directory has been copied onto the board, give execution permission to the `copy_binaries.sh` script and run the script. This script copies the binaries to the required locations on the board:

```
root@imx8mp-lpddr4-evk:~# cd onsemi_ar0830_sensor\i.MX8MPLUS\Binaries\
root@imx8mp-lpddr4-evk:~# chmod +x copy_binaries.sh
root@imx8mp-lpddr4-evk:~# ./copy_binaries.sh
```

   b. Reboot the board and press any key on your keyboard when you see the prompt shown in Figure 44 on your terminal to pause U-Boot execution.

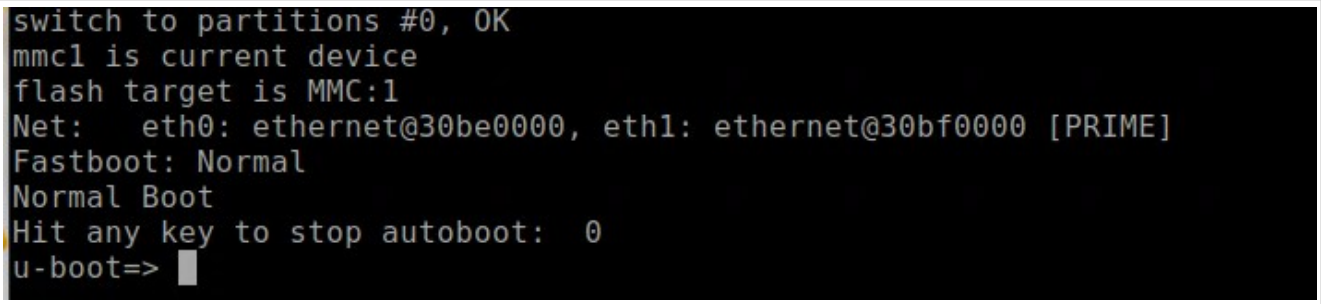


Figure 44. U-Boot prompt

   c. Set the U-Boot environment variable to set the device tree to enable IMX 258 `imx8mp-evk-imx219.dtb`:

```
u-boot=> setenv fdtfile imx8mp-evk-imx258.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

   As a result, the board is rebooted with the selected sensor enabled. At this point, the board is fully enabled to run the selected camera module.

8.3 Hardware connections for IMX 258 camera module

1. Power off the i.MX 8M Plus EVK board before connecting the camera module.
2. Use the highlighted MIPI-CSI 1 port to connect the i.MX 8M Plus EVK board with MINISAS cable to the IMX 258 camera module adapter board shown in Figure 45.
3. Connect using the highlighted HDMI connector on the i.MX 8M Plus EVK board to connect to an external display.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

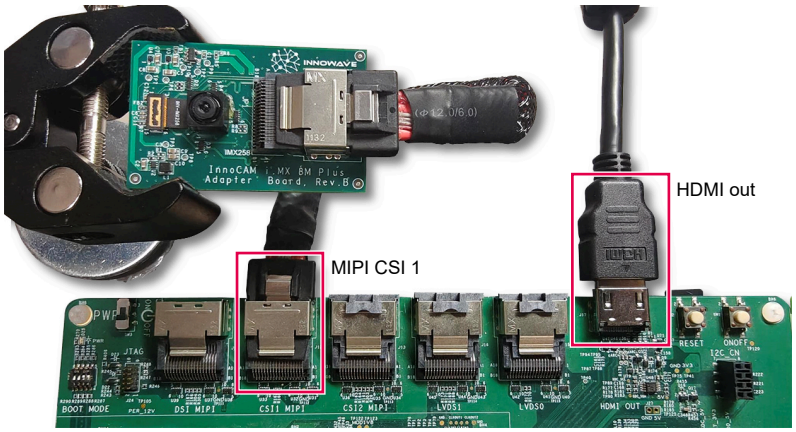


Figure 45. IMX 258 Camera module with InnoCAM i.MX 8M Plus adapter board connected on MIPI-CSI 1 on i.MX 8M Plus EVK board

4. Power on the board. When powered on, the two highlighted LEDs must turn on ensuring proper connection with the camera.

8.4 Test IMX 258 camera module

After connecting the camera module, power on the i.MX 8M Plus EVK board.

To test camera output on the external display, run the following commands:

- To test the IMX 258 camera module, run the following command:

```
root@imx8mp-lpddr4-evk:~# media-ctl -p
```

```
root@imx8mpevk:~# media-ctl -p
Media controller API version 6.6.52

Media device information
------------------------
driver          mxc-md
model           FSL Capture Media Device
serial
bus info        platform:32c00000.bus:camera
hw revision     0x0
driver version  6.6.52

Device topology
- entity 1: mxc-mipi-csi2.0 (8 pads, 1 link)
            type Node subtype V4L flags 0
            device node name /dev/v4l-subdev0
        pad0: Sink
                <- "imx258 1-0010":0 [ENABLED,IMMUTABLE]
        pad1: Sink
        pad2: Sink
        pad3: Sink
        pad4: Source
        pad5: Source
        pad6: Source
        pad7: Source

- entity 10: imx258 1-0010 (1 pad, 1 link, 0 routes)
             type V4L2 subdev subtype Sensor flags 0
             device node name /dev/v4l-subdev1
        pad0: Source
                [stream:0 fmt:SRGGB10_1X10/1920x1080 field:none]
                -> "mxc-mipi-csi2.0":0 [ENABLED,IMMUTABLE]
```

Figure 46. IMX 258 camera sensor is detected

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- Identify v4l2 capture device ID for `VIV (platform:viv0)`:

```
root@imx8mp-lpddr4-evk:~# v4l2-ctl --list-device
```

```
root@imx8mpevk:~# v4l2-ctl --list-devices
 ():
        /dev/v4l-subdev0
        /dev/v4l-subdev2
        /dev/v4l-subdev3

 (csi0):
        /dev/v4l-subdev1

FSL Capture Media Device (platform:32c00000.bus:camera):
        /dev/media0

VIV (platform:viv0):
        /dev/video2

vsi_v4l2dec (platform:vsi_v4l2dec):
        /dev/video1

vsi_v4l2enc (platform:vsi_v4l2enc):
        /dev/video0

viv_media (platform:vvcam-video.0):
        /dev/media1
```

Figure 47. IMX 258 v4l2 capture device

- Use `v4l2src device=</dev/video#>` as the identified v4l2 capture-device ID:

```
root@imx8mp-lpddr4-evk:~# gst-launch-1.0 v4l2src device=/dev/video2 ! "video/x-raw,format=YUY2,width=3840,height=2160" ! waylandsink
```

- Running the above Gstreamer command sets up pipeline to use the v4l2 API to use the camera device on `/dev/video3`, with YUV2 format at a resolution of 1920 x 1080 and use waylandsink as the display output.

Figure 48. IMX 258 capture output

9 Note about the source code in the document

Example code shown in this document has the following copyright and GPL-2.0-only license:

Copyright 2024-2025 NXP

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

10 Revision history

Table 3 summarizes the revisions to this document.

Table 3. Revision history

Document ID	Release date	Description
AN14376 v.2.0	28 March 2025	• Updated the document for: – LF6.6.3-1.0.0 – LF6.6.52-2.2.0 • Updated Section 2 • Updated Section 2.3 • Added ONSemiconductor AR0830 and Sony IMX 258 to Section 2.2 • Updated Section 3.2 • Updated Section 4 • Updated Section 5 • Updated Section 6 • Added Section 7 • Added Section 8
AN14376 v.1.0	11 September 2024	Initial public release

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

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.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2025 NXP B.V. All rights reserved.