AN14853

i.MX Camera Software Pack (i.MX 95 19 mm x 19 mm EVK)

Rev. 1.1 — 28 January 2025

Application note

Document information

Information	Content
Keywords	AN14853, IMX95LPD5EVK-19, camera software pack
Abstract	This document outlines the contents of the i.MX camera software pack for the i.MX 95 19 mm x 19 mm EVK or IMX95LPD5EVK-19.

1 Introduction

The i.MX camera software pack includes camera drivers, libraries, calibration files, and Yocto recipes. These components enable customers to use selected off-the-shelf camera modules out of the box with the IMX95LPD5EVK-19 internal image signal processor (ISP).

It describes steps for using supported camera modules, with the IMX95LPD5EVK-19 ISP for the following supported BSPs:

- LF6.12.20-2.0.0

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 IMX95LPD5EVK-19 • IMX-MIPI-HDMI adaptor Board • RPI-CAM-MINISAS adaptor board • 22-pin/15-pin camera connector FPC cable • Supported camera modules • Display output (Monitor) and HDMI cable • Personal computer
Software	• Ubuntu 22.04 LTS (If building an image using Yocto) • Serial Terminal: Tera Term setup for Windows OS

2.1 NXP IMX95LPD5EVK-19

Figure 1 shows the top view of the IMX95LPD5EVK-19 board.

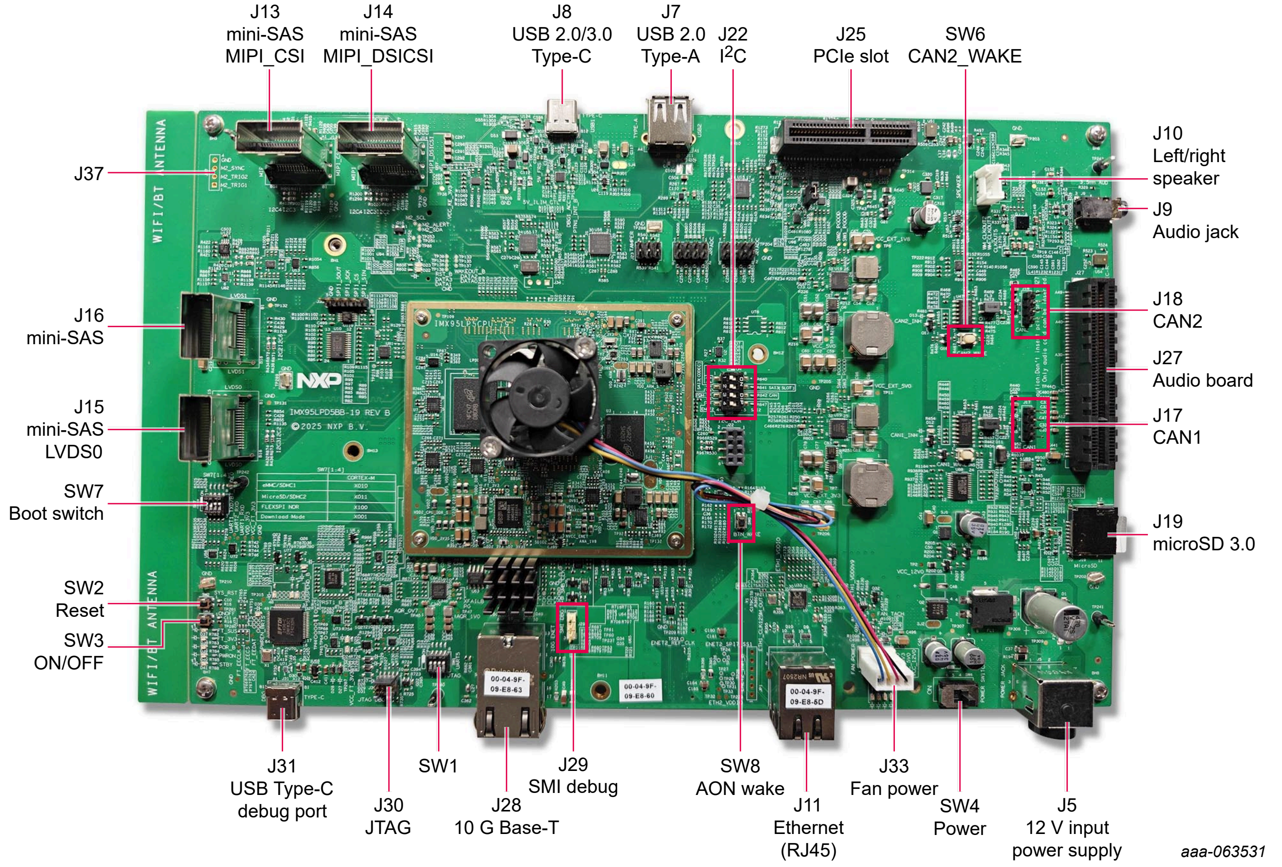


Figure 1. IMX95LPD5EVK-19 board top view

Note: *For more information about the IMX95LPD5EVK-19 board, refer to i.MX95 product page.*

2.2 RPI-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 IMX95LPD5EVK-19 board.

Figure 2 shows the RPI-CAM-MINISAS adaptor board.

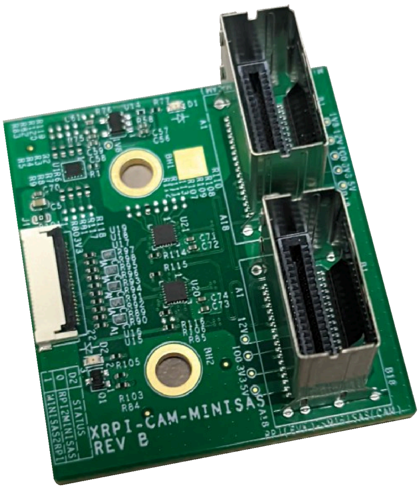


Figure 2. RPI-CAM-MINISAS adaptor board

Note: *For more information about the RPI-CAM-MINISAS adaptor board, refer to RPI-CAM-MINISAS.*

2.3 IMX-MIPI-HDMI adaptor card

Figure 3 shows the IMX-MIPI-HDMI adaptor card.

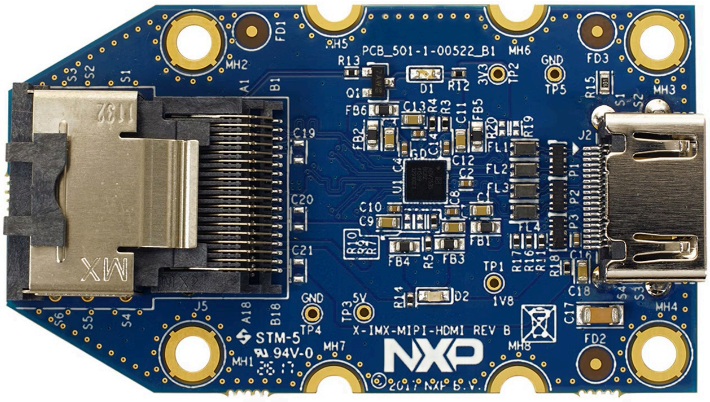


Figure 3. IMX-MIPI-HDMI adaptor card

Note: *For more information about IMX-MIPI-HDMI adaptor board, refer to IMX-MIPI-HDMI.*

2.4 IMX-LVDS-HDMI adaptor card

Figure 4 shows the IMX-LVDS-HDMI adaptor card.

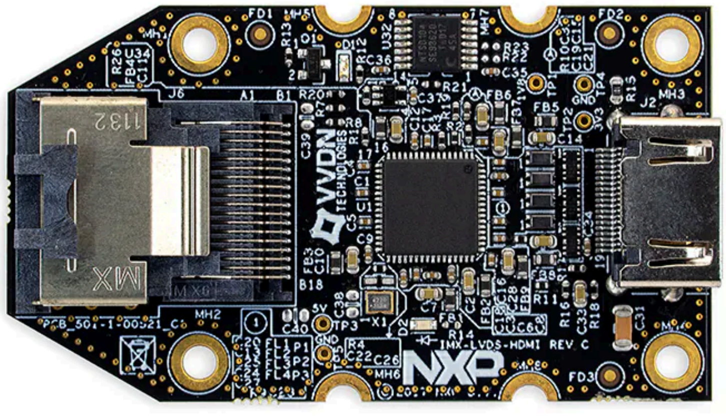


Figure 4. IMX-LVDS-HDMI adaptor card

Note: *For more information about IMX_LVDS-HDMI adaptor board, refer to IMX-LVDS-HDMI.*

2.5 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.

- PiBiger Sony IMX 219 (FOV60 lens):
  - 8 MP (3280 × 2464 active resolution)
  - Current enabled resolutions are:
    - 1920 x 1080 30 fps
    - 3280 x 2464 30 fps

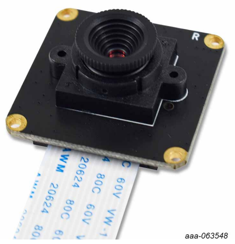


Figure 5. PiBiger Sony IMX 219

- Innowave Sony IMX 258
  - 12 MP
  - Current enabled resolutions are:
    - 3840 x 2160 at 30 FPS
    - 1920 x 1080 at 30 FPS

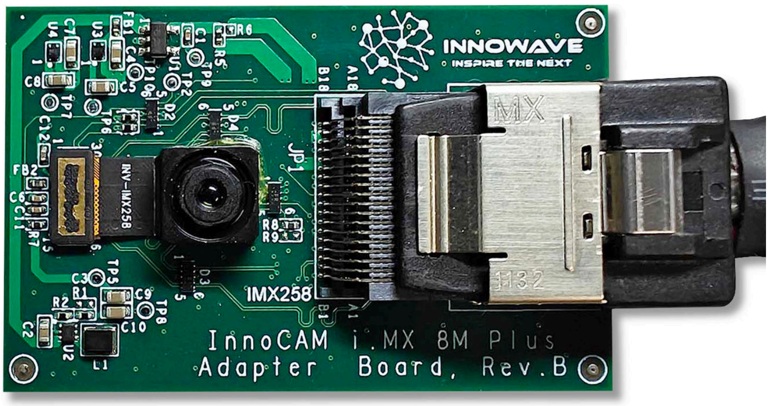


Figure 6. Sony IMX 258 sensor on Innowave InnoCAM i.MX 8M Plus adapter board

- Arducam Sony IMX 477
  - 12 MP
  - Current enabled resolutions are:
    - 3840 x 2160 at 15 FPS

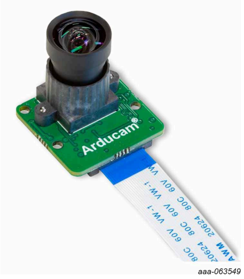


Figure 7. Sony IMX 477 camera module

2.6 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 8. 22 pin (0.5 mm pitch) to 15 pin (1 mm pitch) FPC cable

The Sony IMX477 camera module board connects to the RPI-CAM-MINISAS board through the 22 pin (0.5 mm pitch) FPC cable included with the Camera module. The Innowave Sony IMX258 camera module uses a minisas cable to connect to IMX95LPD5EVK-19 board.

aaa-063530

Figure 9. 22-pin 0.5 mm pitch FPC Flex Cable for DSI CSI

2.7 Flashing image to microSD card on IMX95LPD5EVK-19

Use either of the commands below, as needed on a microSD card with at least 16 GB capacity.

Use open source Flash tool application, for example, RUFUS.exe (https://github.com/pbatard/rufus/releases), balenaEtcher flasher tool (https://github.com/balena-io/etcher/releases) on Windows OS/Ubuntu.

OR

Use the `dd` command on Ubuntu:

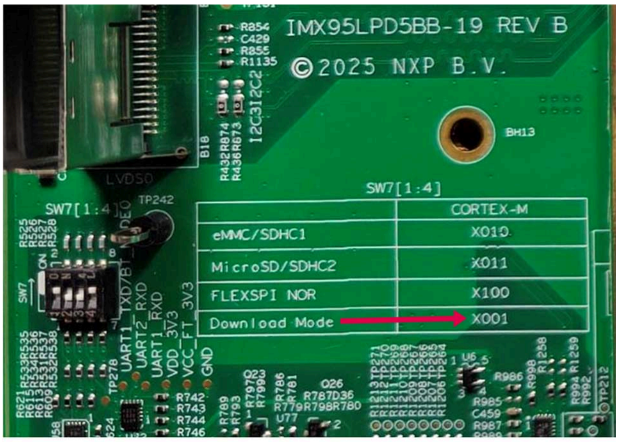
1. Attach a microSD card into a PC running Ubuntu.
2. Change `sd<partition>` below to match the one used by the microSD card. Example commands with full/multimedia BSP images:

```
sudo dd if=imx-image-full-imx95evk.wic of=/dev/sd<partition> bs=1M conv=fsync
zstdcat imx-image-multimedia-imx95-19x19-lpddr5-evk.rootfs-<date-time-stamp>.wic.zst |
 sudo dd of=/dev/sd<partition> bs=1M conv=fsync
```

OR

Use Universal Update Utility (UUU) to flash BSP images to a microSD card attached directly on IMX95LPD5EVK-19:

1. Download UUU (https://github.com/nxp-imx/mfgtools/releases).
2. Place the downloaded “uuu.exe” (for Windows OS) or “uuu” (for Ubuntu) in the same directory as the BSP image to be flashed.
3. Set IMX95LPD5EVK-19 to download mode using boot switch SW7.



aaa-063532

Figure 10. IMX95LPD5EVK-19 boot switch set to Download mode 1001 on SW7

4. Attach USB Type-C cable to USB1 port J1 in Figure 1 on IMX95LPD5EVK-19 board.
5. Insert a microSD card in the microSD card slot on the IMX95LPD5EVK-19 board.
6. Connect USB cable to Windows PC and switch on the IMX95LPD5EVK-19 board.
7. Verify if IMX95LPD5EVK-19 board is detected with UUU.exe:
   For Windows OS open CMD or PowerShell terminal:

```
.\uuu.exe -lsusb
```

   For Ubuntu in terminal:

```
.\uuu -lsusb
```

8. Copy the BSP image to flash in the same location as “UUU.exe” or “uuu” and execute the command to flash the microSD card.
   Example commands on Windows OS for “full” and “multimedia” BSP image:

```
.\uuu.exe -b sd_all .\imx-image-full-imx95evk.wic

.\uuu.exe -b sd_all .\imx-image-multimedia-imx95-19x19-lpddr5-evk.rootfs-<date-time-
stamp>.wic.zst
```

   Example commands on Ubuntu for “full” and “multimedia” BSP image:

```
sudo uuu -b sd_all .\imx-image-full-imx95evk.wic
sudo uuu -b sd_all .\imx-image-multimedia-imx95-19x19-lpddr5-evk.rootfs-<date-time-
stamp>.wic.zst
```

9. After the flash is completed, set the IMX95LPD5EVK-19 board to microSD/SDHC2 boot mode.

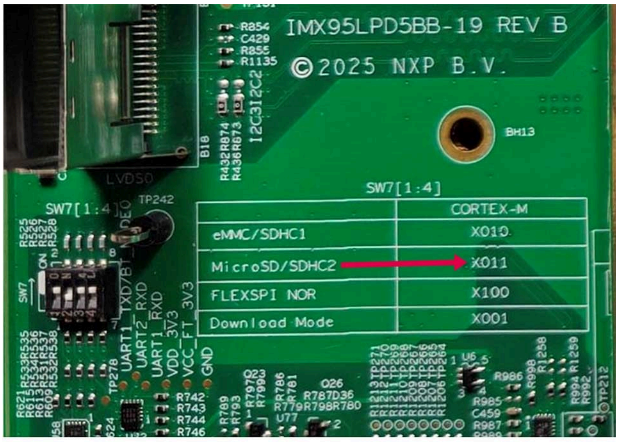


Figure 11. IMX95LPD5EVK-19 boot switch set to microSD/SDHC2 mode 1011 on SW7

2.8 Serial console setup

For Linux OS, the minicom is used to access the serial console of the host platform.

1. Connect the USB-C cable into the debug port (J31 in Figure 1) on IMX95LPD5EVK-19 and the other end to the PC.
2. Open the serial console and login into the device.

```
ubuntu@ubuntu-desktop:/# sudo minicom -s -D /dev/ttyUSBx
```

`ttyUSB0` or `ttyUSB01` are the serial devices. The minicom setup configuration is as follows:

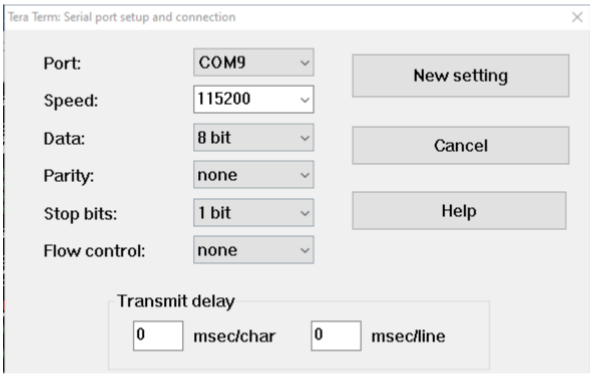
```
A - Serial Device       : /dev/ttyUSBx
E - Bps/Par/Bits        : 115200 8N1
F - Hardware Flow Control : No
```

```
G - Software Flow Control : No
```

3. Save and exit.

For Windows OS, any serial terminal, such as Tera Term or PuTTy is used to access the host platform.

1. Connect the USB-C cable into the debug port (J31 from Figure 1.) on IMX95LPD5EVK-19 and the other end to the PC.
2. Open a serial port console with the baud rate of 115200 and other serial port settings on the PC, as shown in Figure 12.



Tera Term: Serial port setup and connection

Port: COM9

Speed: 115200

Data: 8 bit

Parity: none

Stop bits: 1 bit

Flow control: none

Transmit delay

0 msec/char 0 msec/line

Figure 12. Serial port settings

3. Enter the login information on the console:

```
root@imx95-19x19-lpddr5-evk login: root
```

3 Enable Sony IMX 219 camera module

This section describes two ways to enable the IMX 219 camera module and steps to connect it with IMX95LPD5EVK-19 board using RPI-CAM-MINISAS adaptor board:

- Section 3.1 "Build IMX95LPD5EVK-19 Yocto image with Sony IMX 219 camera module enabled"
- Section 3.2 "Enable IMX 219 using precompiled binaries"

3.1 Build IMX95LPD5EVK-19 Yocto image with Sony IMX 219 camera module enabled

To build iIMX95LPD5EVK-19 Yocto image with Sony IMX 219 camera module enabled, perform the following steps:

1. Perform Prerequisite for Local build.
2. To set up software pack repo, perform the following steps:
   a. Clone and checkout required BSP branch: LF6.12.20_2.0.0

   ```
   $ git clone https://github.com/nxp-imx-support/imx-camera-sw-pack.git
   $ git checkout LF6.12.20_2.0.0
   ```

   b. Copy only *meta-imx95-isp-imx219* into the *sources* directory in your *Yocto_Directory*:

   ```
   $ cp -r  imx-camera-sw-pack/sony_imx219_sensor/i.MX95/meta-imx95-isp-imx219  imx-
   yocto-bsp/sources/
   ```

3. To build and flash the image:
   a. Navigate to the *Yocto_Directory*:

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

b. Source the setup script to configure Yocto build system:

```
$ source sources/meta-imx95-isp-imx219/setup/setup-env-imx95-imx219 -b build
$ bitbake imx-image-multimedia
```

c. After the build process is completed, BSP image file `imx-image-multimedia-imx95-19x19-lpddr5-evk.rootfs.wic.zst` is in this directory:

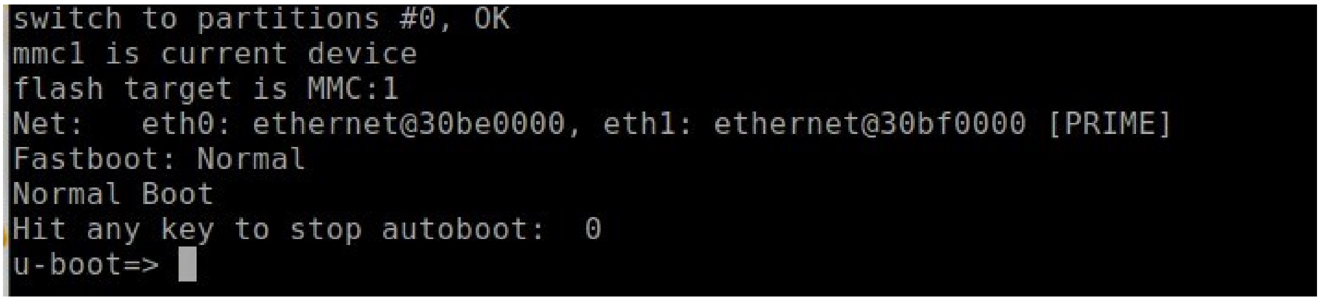
```
$ ls imx-yocto-bsp/build/tmp/deploy/images/imx95-19x19-lpddr5-evk/ imx-image-
multimedia-imx95-19x19-lpddr5-evk.rootfs.wic.zst
```

d. Flash this image on to the SD card or use suitable step from Section 2.7.

```
zstdcat imx-image-multimedia-imx95-19x19-lpddr5-evk.rootfs-<date-time-
stamp>.wic.zst | sudo dd of=/dev/sd<partition> bs=1M conv=fsync
```

4. The following set of commands are executed on the IMX95LPD5EVK-19 board using a serial console terminal:

   a. Boot the IMX95LPD5EVK-19 board with the serial console terminal open.
   While the board is booting, press any key on the keyboard when you see the prompt shown in on your terminal to pause U-Boot execution.



```
switch to partitions #0, OK
mmc1 is current device
flash target is MMC:1
Net:   eth0: ethernet@30be0000, eth1: ethernet@30bf0000 [PRIME]
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot:  0
u-boot=>
```

Figure 13. U-Boot prompt

   b. Set U-Boot environment variable to set the device tree that enables camera sensor IMX 219 and IMX-MIPI-HDMI adaptor (adv7535) *imx95-19x19-evk-isp-imx219-adv7535.dtb*:

```
u-boot=> setenv fdtfile imx95-19x19-evk-isp-imx219-adv7535.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

   At this point, IMX95LPD5EVK-19 board is fully enabled to run the selected camera module and output to HDMI monitor.

For information on connecting the IMX 219 camera module to IMX95LPD5EVK-19 board, via the RPI-CAM-MINISAS adaptor board, see Section 3.3.

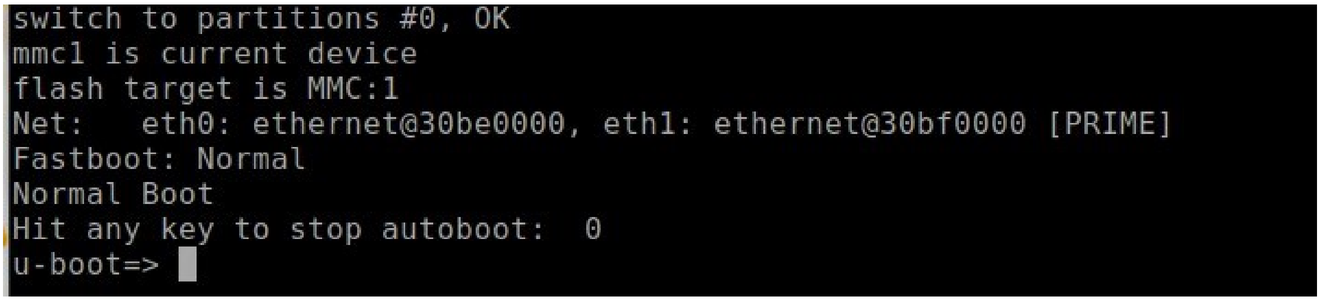
3.2 Enable IMX 219 using precompiled binaries

1. Download the software pack binaries for the required BSP and extract its contents: imx-camera-sw-pack-LF6.12.20_2.0.0-Binaries-i.MX95.zip
2. Perform Prerequistie for Use precompiled binaries.
3. Copy the binaries directory for the selected camera module onto the IMX95LPD5EVK-19 board.
4. The following set of commands are executed on the IMX95LPD5EVK-19 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 binaries to the required locations on the board.

```
root@imx95-19x19-lpddr5-evk:~# cd sony_imx219_sensor\i.MX95\Binaries\
root@imx95-19x19-lpddr5-evk:~# chmod +x copy_binaries.sh
root@imx95-19x19-lpddr5-evk:~#  ./copy_binaries.sh
```

b. Reboot the board and press any key on your keyboard when you see the prompt shown in Figure 14 on your terminal to pause U-Boot execution.



```
switch to partitions #0, OK
mmc1 is current device
flash target is MMC:1
Net:   eth0: ethernet@30be0000, eth1: ethernet@30bf0000 [PRIME]
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot:  0
u-boot=>
```

Figure 14. U-Boot prompt

c. Set the U-Boot environment variable to set the device tree to enable IMX 219 and IMX-MIPI-HDMI adaptor (adv7535) *imx95-19x19-evk-isp-imx219-adv7535.dtb*:

```
u-boot=> setenv fdtfile imx95-19x19-evk-isp-imx219-adv7535.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

At this point, the board is fully enabled to run the selected camera module and output to an HDMI monitor.

3.3 Hardware connections for IMX 219 camera module

1. Power off the IMX95LPD5EVK-19 board before connecting the camera module.
2. Connect RPI-CAM-MINISAS adaptor to IMX 219 camera connector; use the 22 pin (0.5 mm pitch) to 15 pin (1 mm pitch) FPC cable.
   Check the contact direction on both sides of the FPC cable before power on to make sure that the power supply is not reversed.
   Match the circled Black stiffener side as a reference for connections as shown in Figure 15.

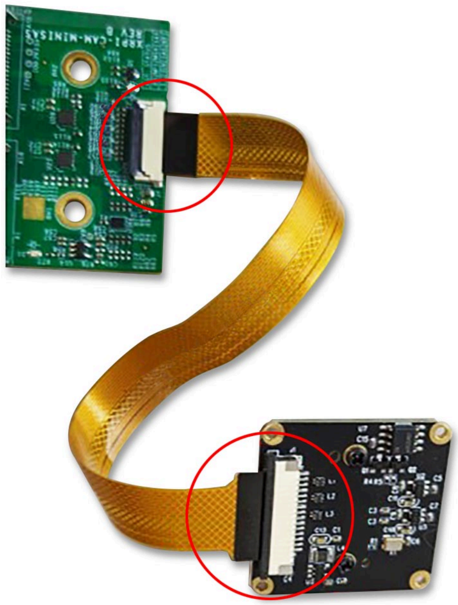


Figure 15. IMX 219 connection with RPI-CAM-MINISAS/RPI-CAM-MIPI board

3. Highlighted MINISAS connector marked [MiniSAS(EVK) -> RPi(CAM)] can be used on XRPI-CAM-MINISAS adaptor board shown in Figure 16.

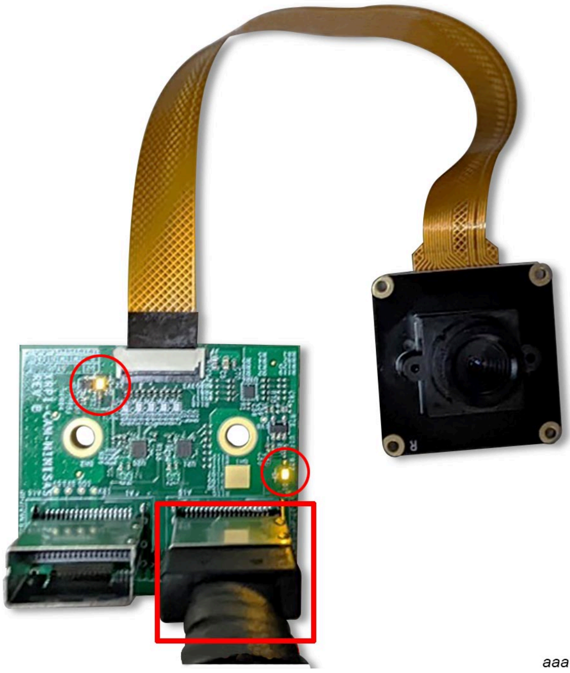


Figure 16. RPI-CAM-MINISAS adaptor board powered ON with IMX 219

4. Use the highlighted MIPI-CSI port to connect IMX95LPD5EVK-19 board with the RPI-CAM-MINISAS adaptor board, as shown in Figure 17.
5. Connect an HDMI monitor using the highlighted MIPI-DSI connector on the IMX95LPD5EVK-19 board with IMX-MIPI-HDMI: MIPI to HDMI adaptor card (mini SAS).

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

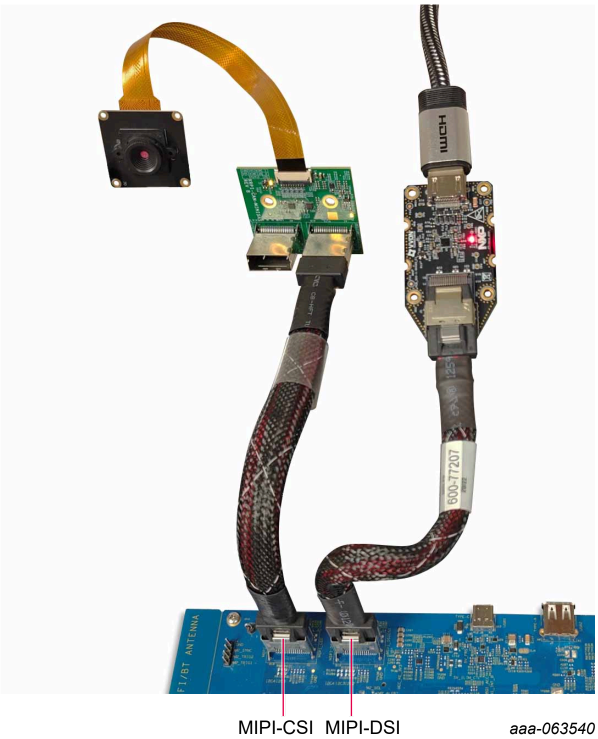


Figure 17. IMX 219 RPI-CAM-MINISAS connection to MIPI-CSI on IMX95LPD5EVK-19 board, IMX-MIPI-HDMI to HDMI monitor

3.4 Test IMX 219 camera module

After connecting the camera module, IMX-MIPI-HDMI adaptor board, power on the IMX95LPD5EVK-19 EVK board.

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

```
root@imx95-19x19-lpddr5-evk:~# cam -l
Available cameras:
1: 'imx219' (/base/soc/bus@42000000/i2c@42530000/imx219@10)
```

- Set the libcamera pipeline handler and uGuzzi IPA:

```
root@imx95-19x19-lpddr5-evk:~# export
LIBCAMERA_IPA_MODULE_PATH="/usr/lib/libcamera/ipa-nxp-neo-uguzzi/"
root@imx95-19x19-lpddr5-evk:~# export
LIBCAMERA_PIPELINES_MATCH_LIST='nxp/neo,imx8-isi,uvc'
```

- To stream camera on HDMI output, run gstreamer command:

```
Set camera to 3280x2464
root@imx95-19x19-lpddr5-evk:~# gst-launch-1.0 libcamerasrc camera-name=/
base/soc/bus@42000000/i2c@42530000/imx219@10 ! video/x-raw , width=3280,
 height=2464, format=YUY2 ! autovideosink

Set camera to 1920x1080
```

```
root@imx95-19x19-lpddr5-evk:~# gst-launch-1.0 libcamerasrc camera-name=/
base/soc/bus@42000000/i2c@42530000/imx219@10 ! video/x-raw , width=1920,
 height=1080, format=YUY2 ! autovideosink
```

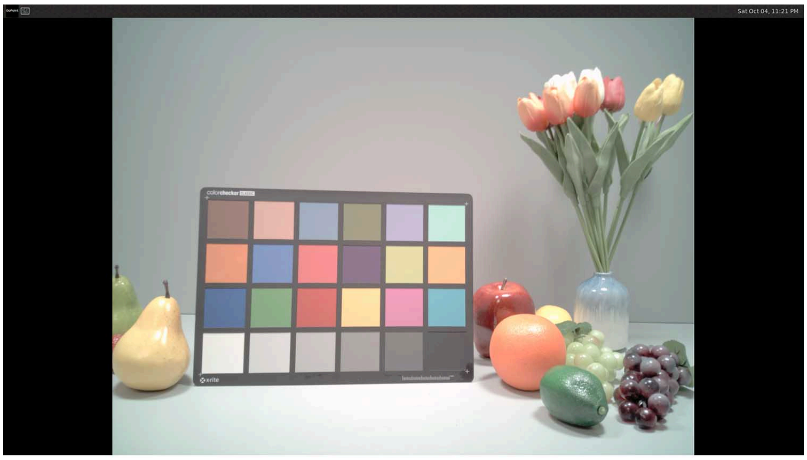

aaa-063541

Figure 18. IMX 219 capture output

4 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 IMX95LPD5EVK-19 board:

- Section 4.1 "Build an IMX95LPD5EVK-19 Yocto image with Sony IMX 258 camera module enabled"
- Section 4.2 "Enable IMX 258 using precompiled binaries"

4.1 Build an IMX95LPD5EVK-19 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: LF6.12.20_2.0.0

   ```
   $ git clone https://github.com/nxp-imx-support/imx-camera-sw-pack.git
   $ git checkout LF6.12.20_2.0.0
   ```

   b. Copy only *meta-imx95-isp-imx258* into the *sources* directory in your *Yocto_Directory*:

   ```
   $ cp -r  imx-camera-sw-pack/sony_imx258_sensor/i.MX95/meta-imx95-isp-imx258  imx-
   yocto-bsp/sources/
   ```

3. To build and flash the image:
   a. Navigate to the *Yocto_Directory*:

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

   b. Source the setup script to configure the Yocto build system and build the image:

   ```
   $ source sources/meta-imx95-isp-imx258/setup/setup-env-imx95-imx258 -b build
   $ bitbake imx-image-multimedia
   ```

c. After the build process is completed, BSP image file *imx-image-multimedia-imx95-19x19-lpddr5-evk.rootfs.wic.zst* is in this directory:

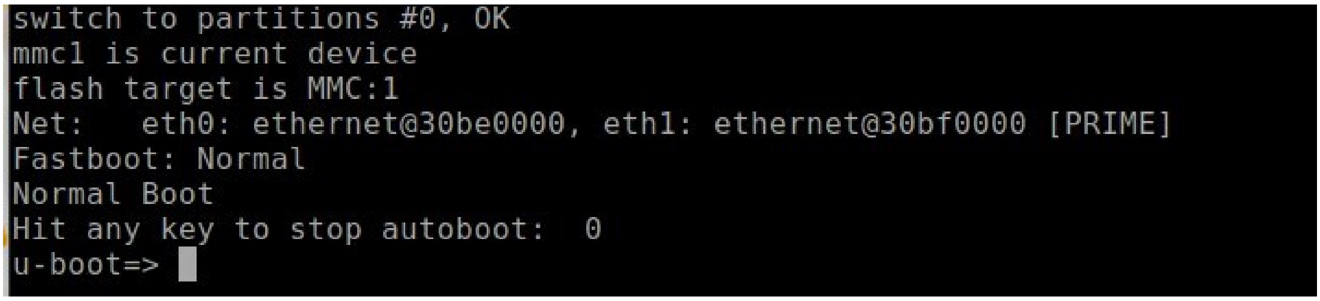
```
$ ls imx-yocto-bsp/build/tmp/deploy/images/imx95-19x19-lpddr5-evk/ imx-image-
multimedia-imx95-19x19-lpddr5-evk.rootfs.wic.zst
```

d. Flash this image on to the SD card or use suitable step from Section 2.7:

```
zstdcat imx-image-multimedia-imx95-19x19-lpddr5-evk.rootfs-<date-time-
stamp>.wic.zst | sudo dd of=/dev/sd<partition> bs=1M conv=fsync
```

4. The following set of commands are executed on the IMX95LPD5EVK-19 board using a serial console terminal:

   a. Now, boot the IMX95LPD5EVK-19 board with the serial console terminal open.
   While the board is booting, press any key on the keyboard when you see the prompt shown in Figure 19 on your terminal to pause U-Boot execution.



```
switch to partitions #0, OK
mmc1 is current device
flash target is MMC:1
Net:   eth0: ethernet@30be0000, eth1: ethernet@30bf0000 [PRIME]
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot:  0
u-boot=>
```

Figure 19. U-Boot prompt

   b. Set the U-Boot environment variable to set the device tree that enables IMX 258 *imx95-19x19-evk-isp-imx258-adv7535.dtb*:

```
u-boot=> setenv fdtfile imx95-19x19-evk-isp-imx258-adv7535.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

   At this point, IMX95LPD5EVK-19 board is ready to run the selected camera module.

For information on connecting the IMX 258 camera module to IMX95LPD5EVK-19 board, see Section 4.3.

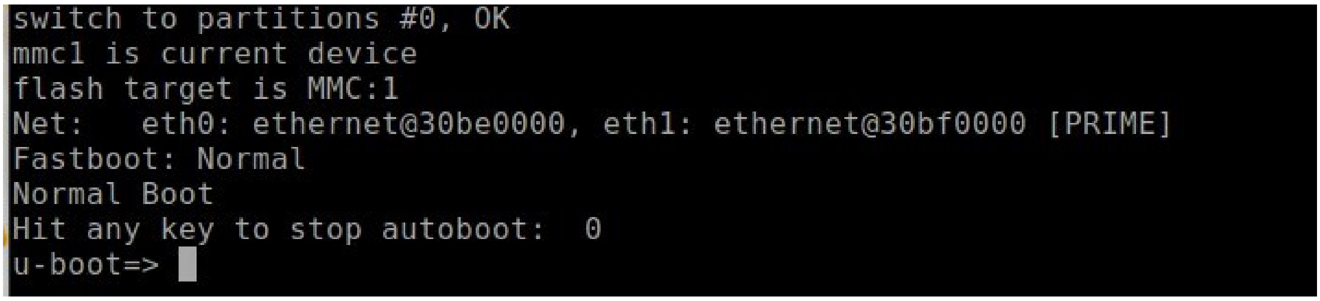
4.2 Enable IMX 258 using precompiled binaries

1. Download the software pack binaries for the required BSP and extract its contents:imx-camera-sw-pack-LF6.12.20_2.0.0-Binaries-i.MX95.zip
2. Perform Prerequistie for Use precompiled binaries.
3. Copy the binaries directory for the selected camera module onto the IMX95LPD5EVK-19 board.
4. The following set of commands are executed on the IMX95LPD5EVK-19 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 binaries to the required locations on the board.

```
root@imx95-19x19-lpddr5-evk:~# cd sony_imx258_sensor\i.MX95\Binaries\
root@imx95-19x19-lpddr5-evk:~# chmod +x copy_binaries.sh
root@imx95-19x19-lpddr5-evk:~#  ./copy_binaries.sh
```

   b. Reboot the board and press any key on your keyboard when you see the prompt shown in Figure 20 on your terminal to pause U-Boot execution.



```
switch to partitions #0, OK
mmc1 is current device
flash target is MMC:1
Net:   eth0: ethernet@30be0000, eth1: ethernet@30bf0000 [PRIME]
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot:  0
u-boot=>
```

Figure 20. U-Boot prompt

c. Set the U-Boot environment variable to set the device tree to enable IMX 258 and IMX-MIPI-HDMI adaptor (adv7535) *imx95-19x19-evk-isp-imx258-adv7535.dtb*:

```
u-boot=> setenv fdtfile imx95-19x19-evk-isp-imx258-adv7535.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

At this point, the board is fully enabled to run the selected camera module and output to an HDMI monitor.

4.3 Hardware connections for IMX 258 camera module

1. Power off the IMX95LPD5EVK-19 board before connecting the camera module.
2. Use the highlighted MIPI-CSI 1 port to connect the IMX95LPD5EVK-19 board with MINISAS cable to the IMX 258 camera module adapter board shown in Figure 21.
3. Connect an HDMI monitor using the highlighted MIPI-DSI connector on the IMX95LPD5EVK-19 board with IMX-MIPI-HDMI: MIPI to HDMI adaptor card (mini SAS).

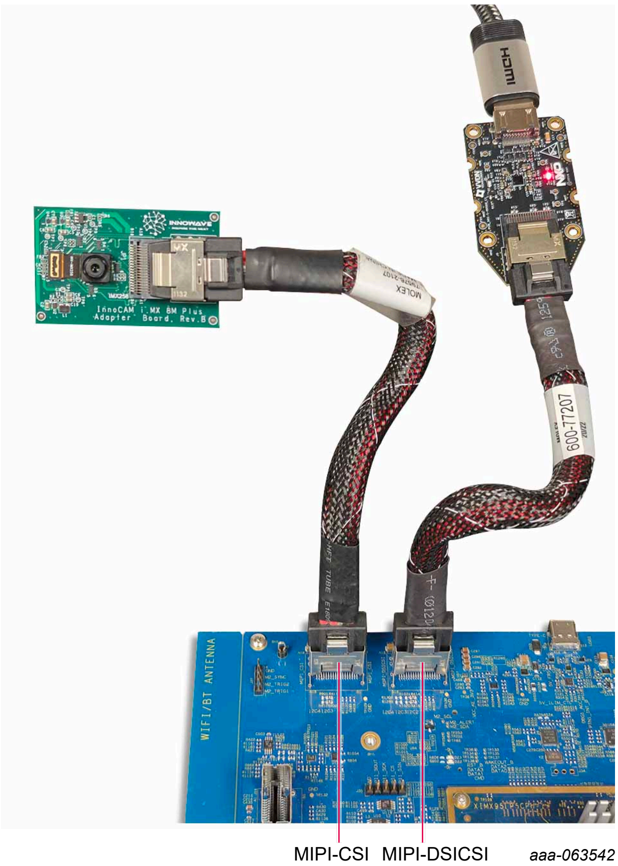


Figure 21. IMX 258 camera module on MIPI-CSI on IMX95LPD5EVK-19

4.4 Test IMX 258 camera module

After connecting the camera module, IMX-MIPI-HDMI adaptor board, power on the IMX95LPD5EVK-19 board.

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

```
root@imx95-19x19-lpddr5-evk:~# cam -l
Available cameras:
1: 'imx258' (/base/soc/bus@42000000/i2c@42530000/imx258_mipi@10)
```

- Set the libcamera pipeline handler and uGuzzi IPA:

```
root@imx95-19x19-lpddr5-evk:~# export
LIBCAMERA_IPA_MODULE_PATH="/usr/lib/libcamera/ipa-nxp-neo-uguzzi/"
root@imx95-19x19-lpddr5-evk:~# export
LIBCAMERA_PIPELINES_MATCH_LIST='nxp/neo,imx8-isi,uvc'
```

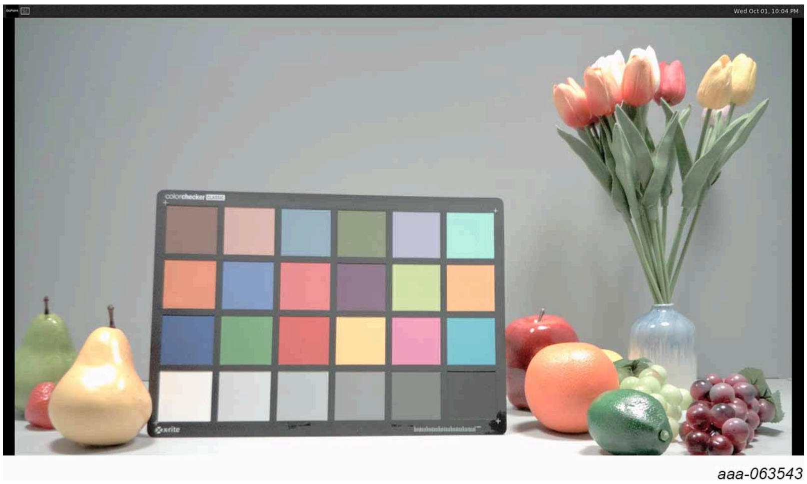
- To stream camera on HDMI output, run gstreamer command:

```
Set Camera to 3840x2160
root@imx95-19x19-lpddr5-evk:~# gst-launch-1.0 libcamerasrc camera-name=/base/
soc/bus@42000000/i2c@42530000/imx258_mipi@10 ! video/x-raw , width=3840,
 height=2160, format=YUY2 ! autovideosink

Set Camera to 1920x1080
```

```
root@imx95-19x19-lpddr5-evk:~# gst-launch-1.0 libcamerasrc camera-name=/base/
soc/bus@42000000/i2c@42530000/imx258_mipi@10 ! video/x-raw , width=1920,
 height=1080, format=YUY2 ! autovideosink
```

- Running the above Gstreamer command sets up IMX 258 to output camera stream at a resolution of 3840 x 2160, or 1920 x 1080 to HDMI output.



aaa-063543

Figure 22. IMX 258 capture output

5 Enable Sony IMX 477 camera module

This section describes two ways to enable IMX 477 camera module and steps to connect it with IMX95LPD5EVK-19 board using RPI-CAM-MINISAS adaptor board:

- Section 5.1 "Build IMX95LPD5EVK-19 Yocto image with Sony IMX 477 camera module enabled"
- Section 5.2 "Enable IMX 477 using precompiled binaries"

5.1 Build IMX95LPD5EVK-19 Yocto image with Sony IMX 477 camera module enabled

1. Perform Prerequisite for Local build.
2. To set up software pack repo, perform the following steps:
   a. Clone and checkout required BSP branch: LF6.12.20_2.0.0

   ```
   $ git clone https://github.com/nxp-imx-support/imx-camera-sw-pack.git
   $ git checkout LF6.12.20_2.0.0
   ```

   b. Copy only *meta-imx95-isp-imx477* into the *sources* directory in your *Yocto_Directory*:

   ```
   $ cp -r  imx-camera-sw-pack/sony_imx477_sensor/i.MX95/meta-imx95-isp-imx477  imx-
   yocto-bsp/sources/
   ```

3. To build and flash the image:
   a. Navigate to the *Yocto_Directory*:

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

   b. Source the setup script to configure Yocto build system:

   ```
   $ source sources/meta-imx95-isp-imx477/setup/setup-env-imx95-imx477 -b build
   $ bitbake imx-image-multimedia
   ```

c. After the build process is completed, BSP image file *imx-image-multimedia-imx95-19x19-lpddr5-evk.rootfs.wic.zst* is in this directory:

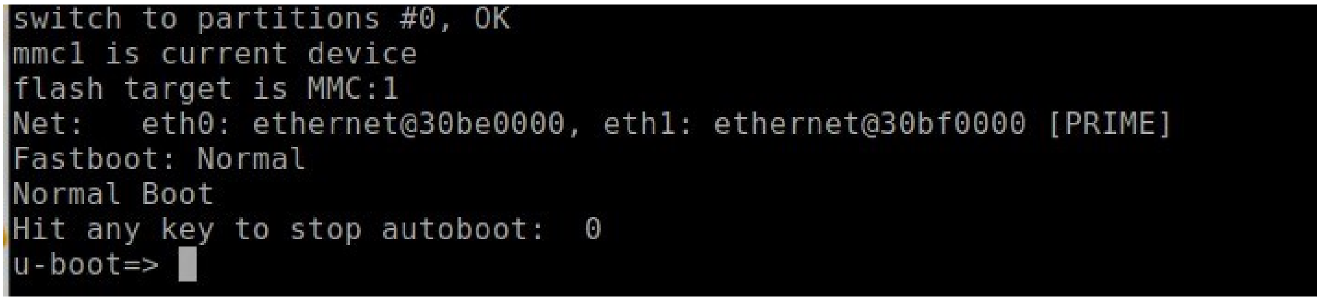
```
$ ls imx-yocto-bsp/build/tmp/deploy/images/imx95-19x19-lpddr5-evk/ imx-image-
multimedia-imx95-19x19-lpddr5-evk.rootfs-<date-time-stamp>.wic.zst
```

d. Flash this image on to the SD card or use suitable step from Section 2.7:

```
zstdcat imx-image-multimedia-imx95-19x19-lpddr5-evk.rootfs-<date-time-
stamp>.wic.zst | sudo dd of=/dev/sd<partition> bs=1M conv=fsync
```

4. The following set of commands are executed on the i.MX 95 EVK board using a serial console terminal:

a. Boot the IMX95LPD5EVK-19 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 23 on your terminal to pause U-Boot execution.



```
switch to partitions #0, OK
mmc1 is current device
flash target is MMC:1
Net:   eth0: ethernet@30be0000, eth1: ethernet@30bf0000 [PRIME]
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot:  0
u-boot=>
```

Figure 23. U-Boot prompt

b. Set U-Boot environment variable to set the device tree that enables camera sensor IMX 477 and IMX-MIPI-HDMI adaptor (adv7535) *imx95-19x19-evk-isp-imx477-adv7535.dtb*:

```
u-boot=> setenv fdtfile imx95-19x19-evk-isp-imx477-adv7535.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

At this point, IMX95LPD5EVK-19 board is fully enabled to run the selected camera module and output to HDMI monitor.

For information on connecting the IMX 477 camera module to IMX95LPD5EVK-19 board, via the RPI-CAM-MINISAS adaptor board, see Section 5.3.

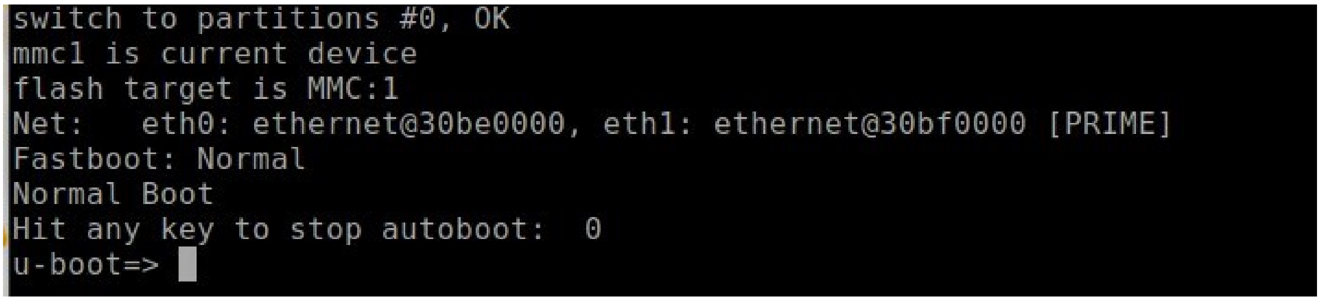
5.2 Enable IMX 477 using precompiled binaries

1. Download the software pack binaries for the required BSP and extract its contents: imx-camera-sw-pack-LF6.12.20_2.0.0-Binaries-i.MX95.zip
2. Perform Prerequistie for Use precompiled binaries.
3. Copy the binaries directory for the selected camera module onto the IMX95LPD5EVK-19 board.
4. The following set of commands are executed on the IMX95LPD5EVK-19 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 binaries to the required locations on the board:

```
root@imx95-19x19-lpddr5-evk:~# cd sony_imx477_sensor\i.MX95\Binaries\
root@imx95-19x19-lpddr5-evk:~# chmod +x copy_binaries.sh
root@imx95-19x19-lpddr5-evk:~#  ./copy_binaries.sh
```

b. Reboot the board and press any key on your keyboard when you see the prompt shown in Figure 24 on your terminal to pause U-Boot execution



```
switch to partitions #0, OK
mmc1 is current device
flash target is MMC:1
Net:   eth0: ethernet@30be0000, eth1: ethernet@30bf0000 [PRIME]
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot:  0
u-boot=>
```

Figure 24. U-Boot prompt

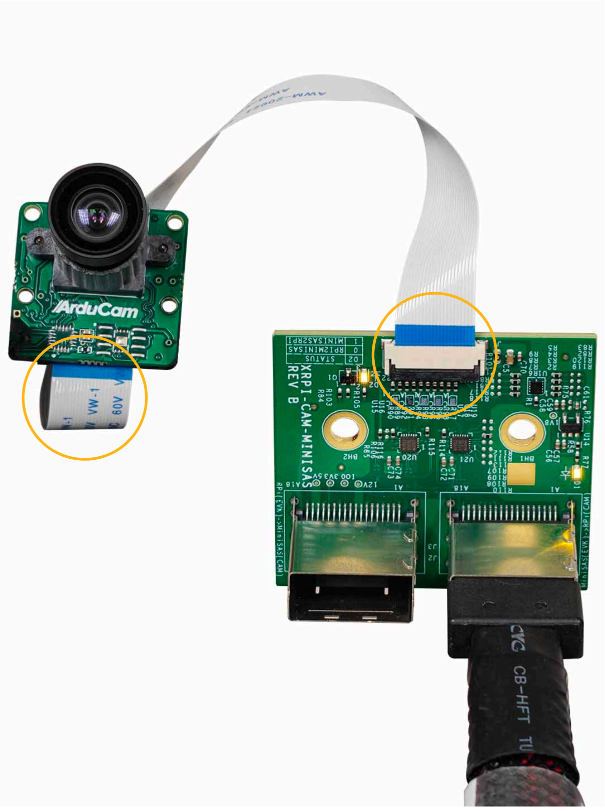
c. Set the U-Boot environment variable to set the device tree to enable IMX 477 and IMX-MIPI-HDMI adaptor (adv7535) *imx95-19x19-evk-isp-imx477-adv7535.dtb*:

```
u-boot=> setenv fdtfile imx95-19x19-evk-isp-imx477-adv7535.dtb
u-boot=> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> boot
```

At this point, the board is fully enabled to run the selected camera module and output to an HDMI monitor.

5.3 Hardware connections for IMX 477 camera module

1. Power off the IMX95LPD5EVK-19 board before connecting the camera module.
2. Connect RPI-CAM-MINISAS adaptor to IMX 477 camera connector; use the 22 pin (0.5 mm pitch) FPC cable included with the camera module.
   Check the contact direction on both sides of the FPC cable before power on to ensure that the power supply is not reversed.
   Match the blue stiffener side and the conductive contact sides as reference for connections circled in Figure 25.



aaa-063544

Figure 25. IMX 477 connection with RPI-CAM-MINISAS board

3. Highlighted MINISAS connector marked [MINISAS(EVK) -> RPI(CAM)] can be used on RPI-CAM-MINISAS adaptor board shown in Figure 26.

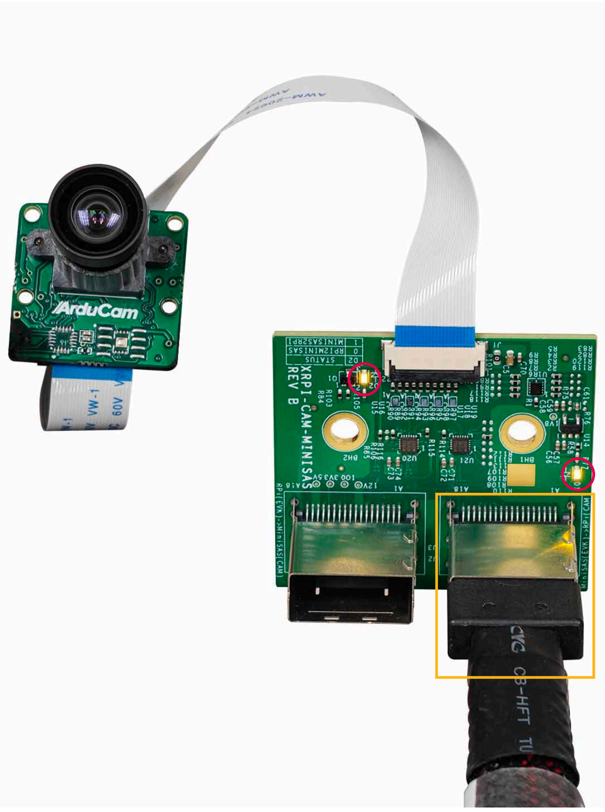


Figure 26. RPI-CAM-MINISAS adaptor board powered on with IMX 477

4. Use the highlighted MIPI-CSI port to connect IMX95LPD5EVK-19 board with the RPI-CAM-MINISAS adaptor board, as shown in Figure 27.
5. Connect an HDMI monitor using the highlighted MIPI-DSI connector on the IMX95LPD5EVK-19 board with IMX-MIPI-HDMI: MIPI to HDMI adaptor card (mini-SAS).
6. Power on the board. When powered on, the two highlighted LEDs must turn on ensuring proper connection with the camera module.

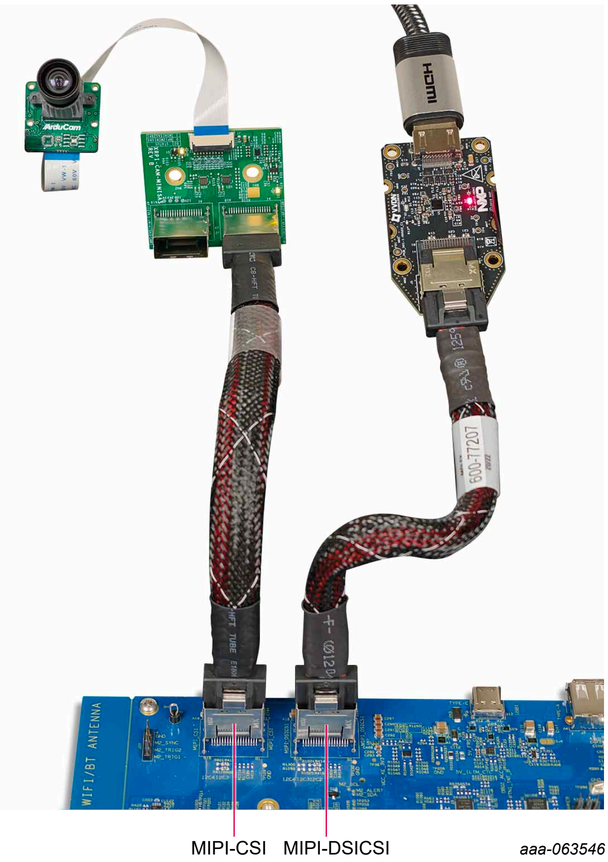


Figure 27. IMX 477 RPI-CAM-MINISAS connection to MIPI-CSI on IMX95LPD5EVK-19 board, IMX-MIPI-HDMI to HDMI monitor

5.4 Test IMX 477 camera module

After connecting the camera module, IMX-MIPI-HDMI adaptor board, power on the IMX95LPD5EVK-19 board.

- To test if IMX 477 camera module is detected, run the following command:

```
root@imx95-19x19-lpddr5-evk:~# cam -l
Available cameras:
1: 'imx477' (/base/soc/bus@42000000/i2c@42530000/imx477@1a)
```

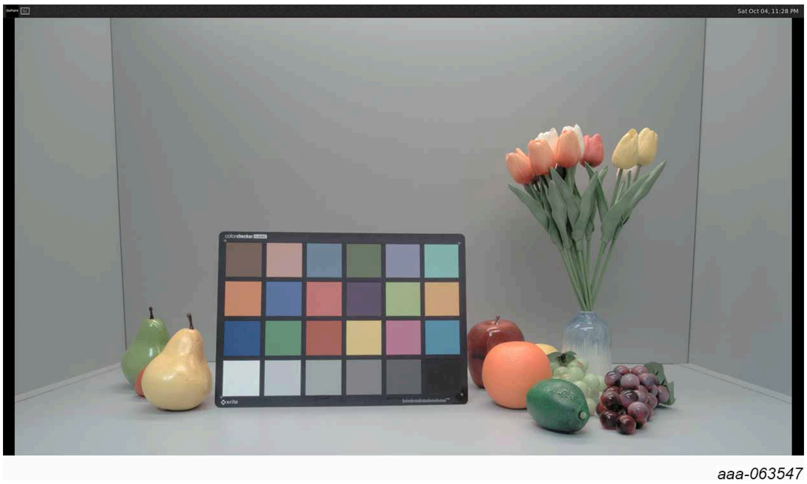
- Set the libcamera pipeline handler and uGuzzi IPA:

```
root@imx95-19x19-lpddr5-evk:~# export
LIBCAMERA_IPA_MODULE_PATH="/usr/lib/libcamera/ipa-nxp-neo-uguzzi/"
root@imx95-19x19-lpddr5-evk:~# export
LIBCAMERA_PIPELINES_MATCH_LIST='nxp/neo,imx8-isi,uvc'
```

- To stream camera on HDMI output, run gstreamer command:

```
root@imx95-19x19-lpddr5-evk:~# gst-launch-1.0 libcamerasrc camera-name=/
base/soc/bus@42000000/i2c@42530000/imx477@1a ! video/x-raw , width=3840,
 height=2160, format=YUY2 ! autovideosink
```

- Running the above Gstreamer command sets up IMX 477 to output camera stream at a resolution of 3840 x 2160 to HDMI output.



aaa-063547

Figure 28. IMX 477 capture output

6 Appendix

6.1 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:

- Section 6.1.1 "Local build"
- Section 6.1.2 "Use precompiled binaries"

6.1.1 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.

6.1.1.1 Prerequisite

Local build prerequisites are required, only if compiling recipes locally using the camera software pack. These steps setup the base Yocto build environment.

6.1.1.1.1 Set up build environment for Yocto project

To build the Yocto image for the IMX95LPD5EVK-19, perform the following steps:

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

```
$ sudo apt-get install build-essential chrpath cpio debianutils diffstat file
 gawk \
  gcc git iputils-ping libacl1 liblz4-tool locales \
  python3 python3-git python3-jinja2 python3-pexpect \
   python3-pip python3-subunit socat texinfo unzip wget xz-utils zstd efitools
```

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 6.12.20-2.0.0 BSP
$ repo init -u https://github.com/nxp-imx/imx-manifest -b imx-linux-walnascar
 -m imx-6.12.20-2.0.0.xml
$ 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=imx95-19x19-lpddr5-evk source imx-setup-
release.sh -b build
```

Where:
- `DISTRO=<distro name> fsl-imx-xwayland` is the distro configuration.
- `MACHINE=<machine name> imx95-19x19-lpddr5-evk` is the board configuration.
- `-b <build_directory>` specifies the name of the build directory.

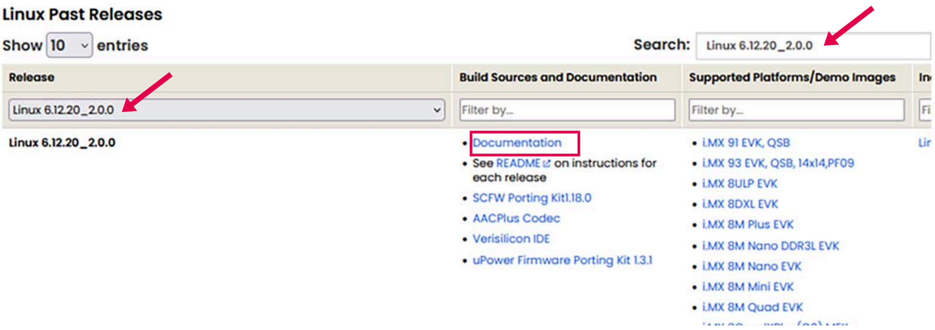
Note: *The "build" directory is referred as the `build_directory` in the next section.*

6. Build the multimedia image using the following command:

```
$ bitbake imx-image-multimedia
```

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 IMXLINUX webpage. For steps to download the right version for the document, see Figure 29.



Linux Past Releases

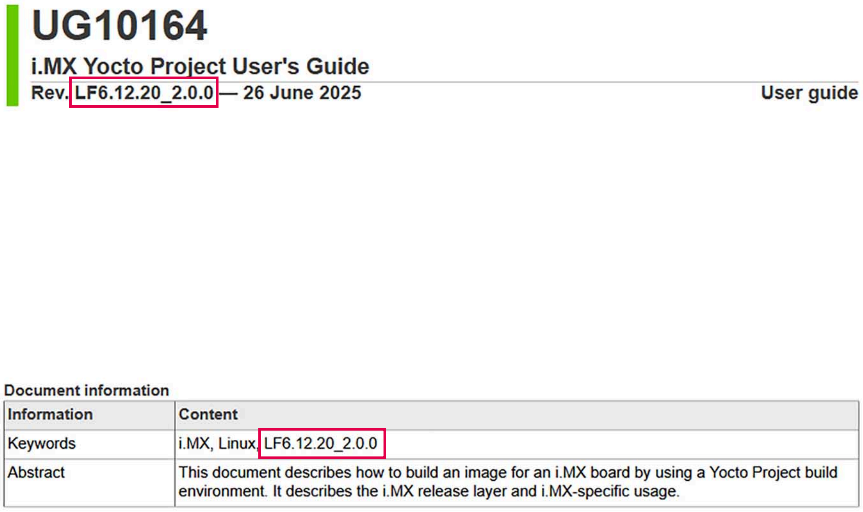
Show 10 entries

Search: Linux 6.12.20_2.0.0

Release	Build Sources and Documentation	Supported Platforms/Demo Images	In
Linux 6.12.20_2.0.0	Filter by...	Filter by...	Fi
Linux 6.12.20_2.0.0	• Documentation • See README on instructions for each release • SCFW Porting Kit1.18.0 • AACPlus Codec • Verisilicon IDE • uPower Firmware Porting Kit 1.3.1	• i.MX 91 EVK, QSB • i.MX 93 EVK, QSB, 14x14,PF09 • i.MX 8ULP EVK • i.MX 8DXL EVK • i.MX 8M Plus EVK • i.MX 8M Nano DDR3L EVK • i.MX 8M Nano EVK • i.MX 8M Mini EVK • i.MX 8M Quad EVK	Lir

aaa-063550

Figure 29. Downloading the Yocto documentation for LF6.12.20-2.0.0



UG10164

i.MX Yocto Project User's Guide

Rev. LF6.12.20_2.0.0 — 26 June 2025 **User guide**

Document information

Information	Content
Keywords	i.MX, Linux, LF6.12.20_2.0.0
Abstract	This document describes how to build an image for an i.MX board by using a Yocto Project build environment. It describes the i.MX release layer and i.MX-specific usage.

aaa-063551

Figure 30. Yocto documentation for LF6.12.20-2.0.0

6.1.2 Use precompiled binaries

The i.MX camera software pack also includes precompiled ISP modules, drivers, and libraries for each sensor. They can be directly copied onto the supported IMX95LPD5EVK-19 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 IMX95LPD5EVK-19 ISP.

It also contains the camera calibration tuning files (database.bin).

Binaries can be downloaded using the link: imx-camera-sw-pack-LF6.12.20_2.0.0-Binaries-i.MX95.zip

Table 2 lists the contents of the binaries directory.

Table 2. Contents of binaries directory

Directory	Content
kernel	Device tree file, overlay, and kernel module to enable sensor
libcamera	Compiled libraries for libcamera

Table 2. Contents of binaries directory*...continued*

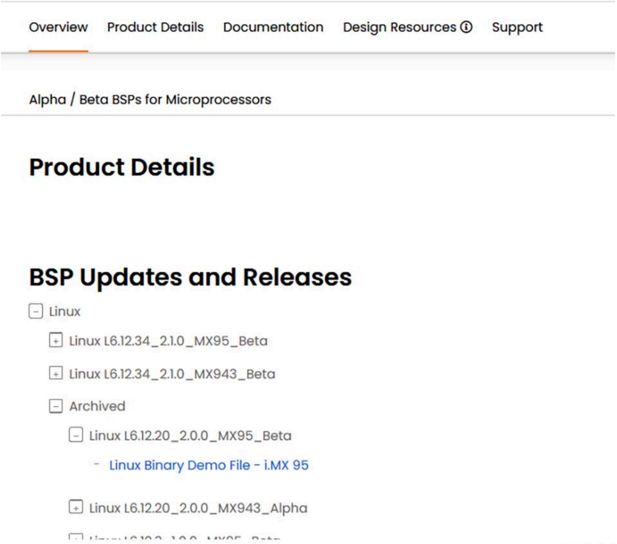
Directory	Content
neo-ipa-uguzzi	Compiled shared libraries for uGuzzi, camera tuning database files
copy_binaries.sh	Script to copy binaries to the right location on the board

6.1.2.1 Prerequisite

This section includes steps to use the precompiled binaries with prebuilt image for IMX95LPD5EVK-19.

6.1.2.2 Use the precompiled binaries with prebuilt image

1. Select and download the right image package for IMX95LPD5EVK-19 marked for the required BSP version: Linux 6.12.20-2.0.0



Alpha / Beta BSPs for Microprocessors

Product Details

BSP Updates and Releases

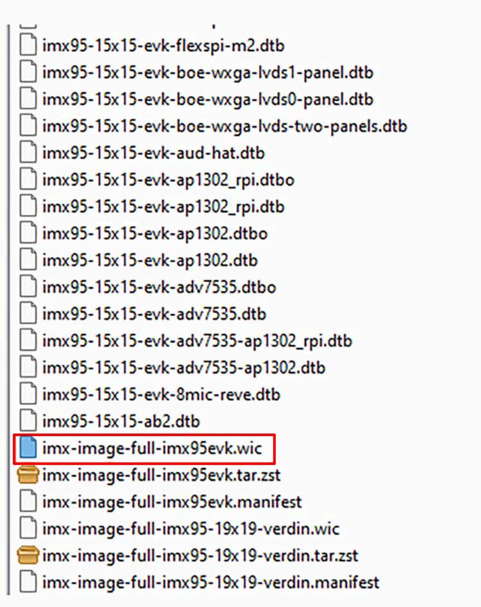
- ⊟ Linux
  - ⊞ Linux L6.12.34_2.1.0_MX95_Beta
  - ⊞ Linux L6.12.34_2.1.0_MX943_Beta
  - ⊟ Archived
    - ⊟ Linux L6.12.20_2.0.0_MX95_Beta
      - Linux Binary Demo File - i.MX 95
    - ⊞ Linux L6.12.20_2.0.0_MX943_Alpha

aaa-063534

Figure 31. Example for downloading Linux 6.12.20_2.0.0 IMX95LPD5EVK-19 image

2. Extract the contents of the downloaded BSP zip file, taking Linux 6.12.20_2.0.0 as an example *LF_v6.12.20-2.0.0_images_IMX95.zip*.
3. After extracting, select the `imx-image-full-imx95evk.wic` image file and flash it onto an SD card using suitable flash method from Section 2.7.
   If using Ubuntu, run the following command:

```
sudo dd if=imx-image-full-imx95evk.wic of=/dev/sd<partition> bs=1M conv=fsync
```



```
imx95-15x15-evk-flexspi-m2.dtb
imx95-15x15-evk-boe-wxga-lvds1-panel.dtb
imx95-15x15-evk-boe-wxga-lvds0-panel.dtb
imx95-15x15-evk-boe-wxga-lvds-two-panels.dtb
imx95-15x15-evk-aud-hat.dtb
imx95-15x15-evk-ap1302_rpi.dtbo
imx95-15x15-evk-ap1302_rpi.dtb
imx95-15x15-evk-ap1302.dtbo
imx95-15x15-evk-ap1302.dtb
imx95-15x15-evk-adv7535.dtbo
imx95-15x15-evk-adv7535.dtb
imx95-15x15-evk-adv7535-ap1302_rpi.dtb
imx95-15x15-evk-adv7535-ap1302.dtb
imx95-15x15-evk-8mic-reve.dtb
imx95-15x15-ab2.dtb
imx-image-full-imx95evk.wic
imx-image-full-imx95evk.tar.zst
imx-image-full-imx95evk.manifest
imx-image-full-imx95-19x19-verdin.wic
imx-image-full-imx95-19x19-verdin.tar.zst
imx-image-full-imx95-19x19-verdin.manifest
```

aaa-063539

Figure 32. Full image for IMX95LPD5EVK-19

4. Boot the IMX95LPD5EVK-19 with the flashed SD card while connected to the serial console terminal on your PC.

7 Note about the source code in the document

Example code shown in this document has the following copyright and GPL-2.0 or (at your option) any later version license:

Copyright 2026 NXP

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; either version 2 of the License, or (at your option) any later version.

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, see <https://www.gnu.org/licenses/>.

8 Revision history

Table 3 summarizes the revisions to this document.

Table 3. Revision history

Document ID	Release date	Description
AN14853 v.1.1	28 January 2026	Legal information updated for disclaimers : • t001dis102 • t001dis110
AN14853 v.1.0	20 January 2026	Initial public 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.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

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.

Contents

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