Using a Camera on iMX Developer's Kits

Embedded Artists

Embedded Artists AB
Jörgen Ankersgatan 12
SE-211 45 Malmö
Sweden

http://www.EmbeddedArtists.com

Disclaimer

Embedded Artists AB makes no representation or warranties with respect to the contents hereof and specifically disclaim any implied warranties or merchantability or fitness for any particular purpose. Information in this publication is subject to change without notice and does not represent a commitment on the part of Embedded Artists AB.

Feedback

We appreciate any feedback you may have for improvements on this document. Send your comments by using the contact form: www.embeddedartists.com/contact.

Trademarks

All brand and product names mentioned herein are trademarks, services marks, registered trademarks, or registered service marks of their respective owners and should be treated as such.

Table of Contents

1 Document Revision History

Revision	*Date*	*Description*
A	2018-04-12	First release

2 Introduction

This document describes how to connect a camera to Embedded Artists i.MX 6/7 based Developer's Kits. Three different interfaces are described: MIPI/CSI-2, Parallel, and USB. Not all boards support all interfaces.

Additional documentation you might need is.

- The *Getting Started* document for the board you are using.
- The *Adding Displays to iMX Developer's Kits* document about displays and how to use them

2.1 Conventions

A number of conventions have been used throughout to help the reader better understand the content of the document.

`Constant width text` – is used for file system paths and command, utility and tool names.

```
$ This field illustrates user input in a terminal running on the
development workstation, i.e., on the workstation where you edit,
configure and build Linux
```

```
# This field illustrates user input on the target hardware, i.e.,
input given to the terminal attached to the COM Board
```

```
This field is used to illustrate example code or excerpt from a
document.
```

This fieild is used for important notes and highlights

3 Before getting started

3.1 Linux distributions

The instructions in this document have been tested on Embedded Artists **4.9.11** distribution.

3.2 Yocto configuration

GStreamer will be used to capture video feed from a camera and **v4l-utils** will be used to query a camera interface.

These tools have been included In Embedded Artists 4.9.11 distributions published **2018-04-12** or later. You can add support in your own Yocto build by adding the below packages to your `conf/local.conf` file.

```
IMAGE_INSTALL_append = " \
    v4l-utils \
    packagegroup-fsl-gstreamer1.0 \
    packagegroup-fsl-gstreamer1.0-full \
"
```

4 USB interface

4.1 Camera

A **Logitech QuickCam Pro 9000** camera was used when testing a camera via the USB interface.

Figure 1 - Logitech QuickCam Pro 9000

4.2 Supported boards

The following boards support USB based cameras.

- iMX6 SoloX COM
- iMX6 Quad COM
- iMX6 DualLite COM
- iMX6 UltraLite COM
- iMX7 Dual COM
- iMX7 Dual uCOM

4.3 Instructions

1. Connect the camera to the USB host port on the carrier board as shown in Figure 2.

Figure 2 - USB Host connector

2. The camera will be detected and you will see output in the console similar to below.

```
usb 1-1.1: USB disconnect, device number 3
usb 1-1.3: new high-speed USB device number 4 using ci_hdrc
uvcvideo: Found UVC 1.00 device <unnamed> (046d:0990)
input: UVC Camera (046d:0990) as /devices/soc0/soc/2100000.aips-
bus/2184200.usb/ci_hdrc.1/usb1/1-1/1-1.3/1-1.3:1.0/input/input1
usbcore: registered new interface driver uvcvideo
USB Video Class driver (1.1.1)
usb 1-1.3: Warning! Unlikely big volume range (=3072), cval->res
is probably wrong.
usb 1-1.3: [5] FU [Mic Capture Volume] ch = 1, val = 4608/7680/1
usbcore: registered new interface driver snd-usb-audio
```

3. Use `v4l2-ctl` to find out on which video interface the camera is attached. In this example four interfaces are available and the USB camera (UVC Camera) is attached to `/dev/video3`.

```
# v4l2-ctl --list-devices
i.MX6S_CSI (platform:2214000.csi):
        /dev/video1

i.MX6S_CSI (platform:221c000.csi):
        /dev/video2

pxp (pxp_v4l2_out):
        /dev/video0

UVC Camera (046d:0990) (usb-ci_hdrc.1-1.3):
        /dev/video3
```

4. Use the command below to start capturing a video stream and show it on a display. If you don't have a display, go to step 5 and take a snapshot instead.

```
# gst-launch-1.0 -v imxv4l2src device=/dev/video3 ! capsfilter
caps="video/x-raw, width=640, height=480, framerate=30/1" ! queue
! imxv4l2sink
```

5. Use the command below to take a snapshot from the camera. The snapshot is stored as a jpeg file.

```
# gst-launch-1.0 imxv4l2src device="/dev/video3" num-buffers=1 !
capsfilter caps="video/x-raw, width=640, height=480,
framerate=30/1" ! videoconvert ! jpegenc ! filesink
location=snapshot.jpeg
```

6. In the above examples a resolution of 640x480 and frame rate of 30 fps was used. You can query the camera to see which resolutions and frame rates it supports for different pixel formats. Only a portion of the output from the command is shown below.

```
# v4l2-ctl --list-formats-ext -d /dev/video3
ioctl: VIDIOC_ENUM_FMT
        Index       : 0
        Type        : Video Capture
        Pixel Format: 'MJPG' (compressed)
        Name        : Motion-JPEG
                Size: Discrete 160x120
```

```
                        Interval: Discrete 0.033s (30.000 fps)
                        Interval: Discrete 0.040s (25.000 fps)
                        Interval: Discrete 0.050s (20.000 fps)
...
```

5 MIPI/CSI-2 interface

5.1 Camera

The **Pcam 5C** camera from Digilent was used when testing a camera with MIPI interface. This camera is using the Omnivision **OV5640** image sensor. Two data lanes are connected. The camera comes with a (about) 200 mm long, 20-pos, 1 mm pitch flat cable.

> Do not extend the cable length more than this. The total length of the MIPI/CSI-2 trace length should be kept less than 200 mm. This is due to the high frequency signals on the MIPI/CSI-2 interface.

Figure 3 - Digilent Pcam 5C (OV5640)

5.2 Supported boards

The following boards support cameras with MIPI/CSI-2 interface.

- iMX6 Quad COM
- iMX6 DualLite COM
- iMX7 Dual COM
- iMX7 Dual uCOM

5.3 Instructions

1. Connect the camera to the J20 connector as shown in Figure 4 and Figure 5. The exposed pads on the cable shall be mounted towards the nearby PCB edge. Figure 4 illustrates the exposed pads. Figure 5 illustrates the side of the cable without exposed pads.

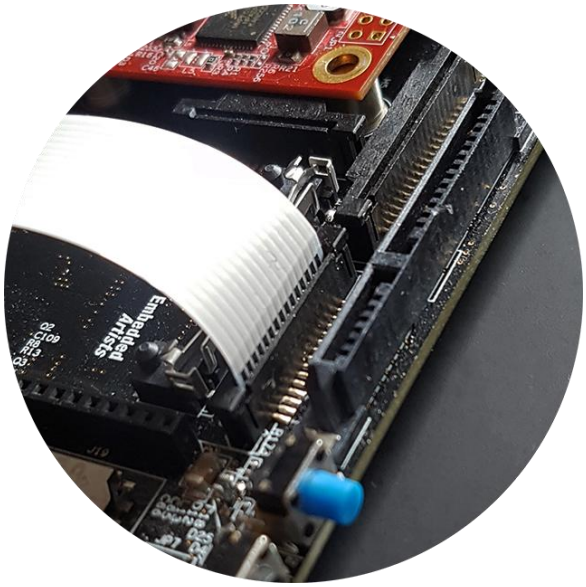

Figure 4 - Cable inserted into J20 (back)

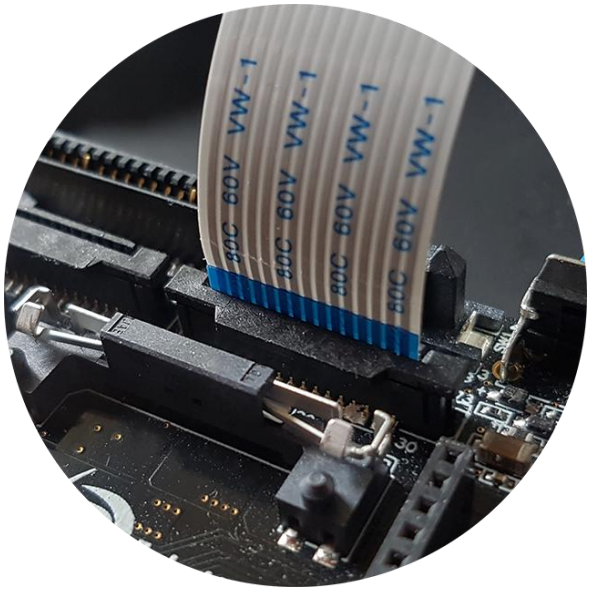

Figure 5 - Cable inserted into J20 (front)

2. Boot into the u-boot. We need to change to a different dtb (device tree) file before booting into Linux. This new dtb file enables the MIPI/CSI-2 interface and adds the ov5640 camera to the I2C bus. In this example we are using an iMX6 Quad COM board. For other boards the name of the dtb file will be different. Change "imx6qea-com-kit" to the name of the board you are using, for example, "imx7dea-com-kit" if you are using the iMX7 Dual COM board.

```
=> setenv fdt_file imx6qea-com-kit-ov5640.dtb
=> saveenv
```

If you want to see how the device file looks like you can find it on our GitHub repository.

https://github.com/embeddedartists/linux-imx/blob/ea_4.9.11_1.0.0/arch/arm/boot/dts/imx6qea-com-kit-ov5640.dts

3. Reset the board and boot into Linux. When you have logged into Linux run the command below to see if the camera has been found. You should see the message "camera ov5640_mipi is found".

```
# dmesg | grep ov5640
camera ov5640_mipi is found
```

4. Use `v4l2-ctl` to list available video interfaces. In this example three interfaces are listed, but no detailed information about each interface. The camera is in this case available on `/dev/video0`.

```
# v4l2-ctl --list-devices
():
        /dev/video0
        /dev/video16
        /dev/video17
```

5. Use the command below to start capturing a video stream and show it on a display. If you don't have a display, go to step 6 and take a snapshot instead.

```
# gst-launch-1.0 -v imxv4l2src device=/dev/video0 ! capsfilter
caps="video/x-raw, width=640, height=480, framerate=30/1" ! queue
! imxv4l2sink
```

6. Use the command below to take a snapshot from the camera. The snapshot is stored as a jpeg file.

```
# gst-launch-1.0 imxv4l2src device="/dev/video0" num-buffers=1 !
capsfilter caps="video/x-raw, width=640, height=480,
framerate=30/1" ! videoconvert ! jpegenc ! filesink
location=snapshot.jpeg
```

6 Parallel interface

6.1 Camera

The **Waveshare OV5640 Camera Board** was used when testing a camera with parallel interface. This camera is using the Omnivision **OV5640** image sensor. This sensor supports both the MIPI/CSI-2 and parallel camera interface. On the Waveshare board, the parallel interface is connected.

Figure 6 – Waveshare OV5640 Camera Board

Please note that this board cannot be connected directly to J21 connector on Embedded Artists COM Carrier board. You need to make an adapter between the camera board and the COM Carrier board.

Also note that the camera must be supplied with a 24 MHz clock with 2.8V voltage level. A 3.3V logic level clock will create disturbances in the camera. A local 24 MHz oscillator, powered by 2.8V, can be placed on the adapter between the Waveshare module and COM Carrier board.

Note that the parallel camera interface cannot be extended more than 100 mm from the MXM3 connector. Depending on pixel clock frequency, it can even be shorter than this. In general, Embedded Artists recommend using the MIPI/CSI-2 interface when connecting a camera.

6.2 Supported boards

The following boards support cameras with parallel interface.

- iMX6 SoloXCOM

6.3 Instructions

1. Connect the camera to the J21 connector as shown in Figure 7 and Figure 8.

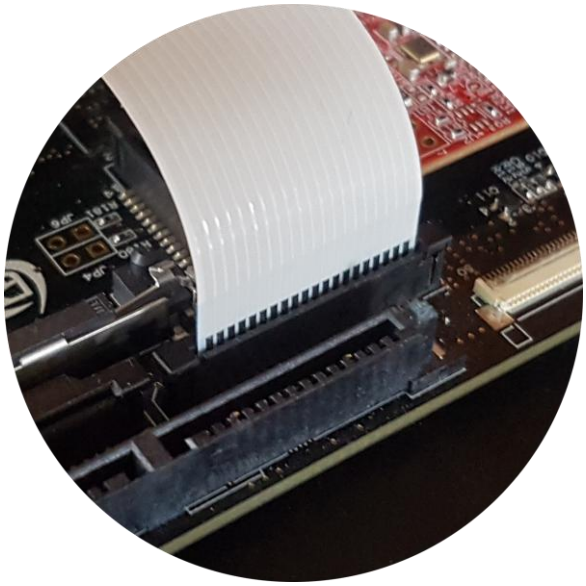

Figure 7 - Cable inserted into J21 (back)

Figure 8 - Cable inserted into J20 (front)

2. Boot into the u-boot. We need to change to a different dtb (device tree) file before booting into Linux. This new dtb file enables the parallel interface and adds the ov5640 camera to the I2C bus. In this example we are using an iMX6 SoloX COM board. For other boards the name of the dtb file will be different. Change "imx6sxea-com-kit" to the name of the board you are using, for example, "imx6qea-com-kit" if you are using the iMX6 Quad COM board.

```
=> setenv fdt_file imx6sxea-com-kit-ov5640-pl.dtb
=> saveenv
```

If you want to see how the device file looks like you can find it on our GitHub repository.

https://github.com/embeddedartists/linux-imx/blob/ea_4.9.11_1.0.0/arch/arm/boot/dts/imx6sxea-com-kit-ov5640-pl.dts

3. Reset the board and boot into Linux. When you have logged into Linux run the command below to see if the camera has been found. You should see the message "camera ov5640, is found".

```
# dmesg | grep ov5640
camera ov5640, is found
```

4. Use `v4l2-ctl` to list available video interfaces. In this example three interfaces are listed, but no detailed information about each interface. The camera is in this case available on `/dev/video1`.

```
# v4l2-ctl --list-devices
i.MX6S_CSI (platform:2214000.csi):
        /dev/video1

i.MX6S_CSI (platform:221c000.csi):
        /dev/video2

pxp (pxp_v4l2_out):
        /dev/video0
```

5. Use the command below to start capturing a video stream and show it on a display. If you don't have a display, go to step 6 and take a snapshot instead.

```
# gst-launch-1.0 -v imxv4l2src device=/dev/video1 ! capsfilter
caps="video/x-raw, width=640, height=480, framerate=30/1" ! queue
! imxv4l2sink
```

6. Use the command below to take a snapshot from the camera. The snapshot is stored as a jpeg file.

```
# gst-launch-1.0 imxv4l2src device="/dev/video1" num-buffers=1 !
capsfilter caps="video/x-raw, width=640, height=480,
framerate=30/1" ! videoconvert ! jpegenc ! filesink
location=snapshot.jpeg
```