

Application Note

How to use phyCAM camera modules with phyCORE-i.MX 8M SBC on phyBOARD Polaris

Revision History

Version	Changes	Author	Date
A0	Initial Release	H. Fendrich	05.02.2020
A1	Section 4, added reboot command	H. Fendrich	07.05.2020
A2	added phyCAM-M cable note	H. Fendrich	17.06.2020
A3	added support PD 21.1.0	H. Fendrich	24.08.2021

Content

1	Overview.....	2
2	Camera Connectors on the Polaris - Carrier Boards	4
3	Supported Camera Boards (YOCTO Linux PD19.1.0 and higher).....	5
4	Supported Camera Boards (YOCTO Linux PD21.1.0 and higher).....	6
4.1	Supported Camera Types in Standard Vision Image (phytec-vision-image-phyboard-polaris-imx8m-3).....	6
4.2	Change the Camera Types or Connected Port in Standard Vision Image (phytec-vision-image-phyboard-polaris-imx8m-3)	7
4.2.1	How to Change the Device Tree.....	7
5	Demo Scripts	8
5.1	GStreamer Scripts	8
5.2	Scripts to Call C/C++ files Based on v4l2 Interface	10
5.3	OpenCV Scripts	10
6	De-Bayering (demosacking) with NEON CoProcessor	11

1 Overview

The i.MX 8M Microcontroller supported 2 MIPI CSI-2 camera interfaces (see figure 1).

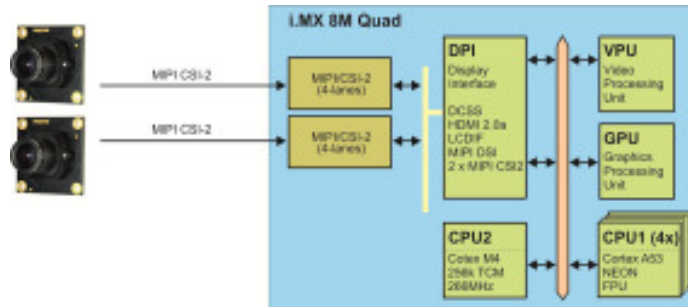


figure 1: Block Diagram Camera Interfaces of i.MX 8M Controller (Quad)

On the phyCORE-i.MX 8M the CSI1/CAM1 and CSI2/CAM2 camera path go out as CSI-2 MIPI signal. (see figure 2)

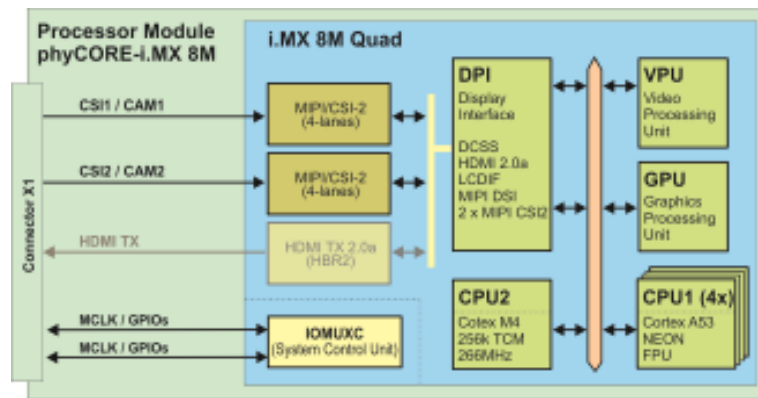


figure 2: CAM_1 (CSI1) and CAM_2 (CSI2) are present on connector X1

On the PHYTEC or customer carrier boards can the interfaces be led out as phyCAM-M. For more information to phyCAM-M see the phyCAM-Manuals on PHYTEC Homepage.
 On the phyBOARD-Polaris baseboard are both camera interfaces led out as phyCAM-M (MIPI CSI-2) interfaces (see figure 3). Here you can connect different phyCAM-M camera modules.

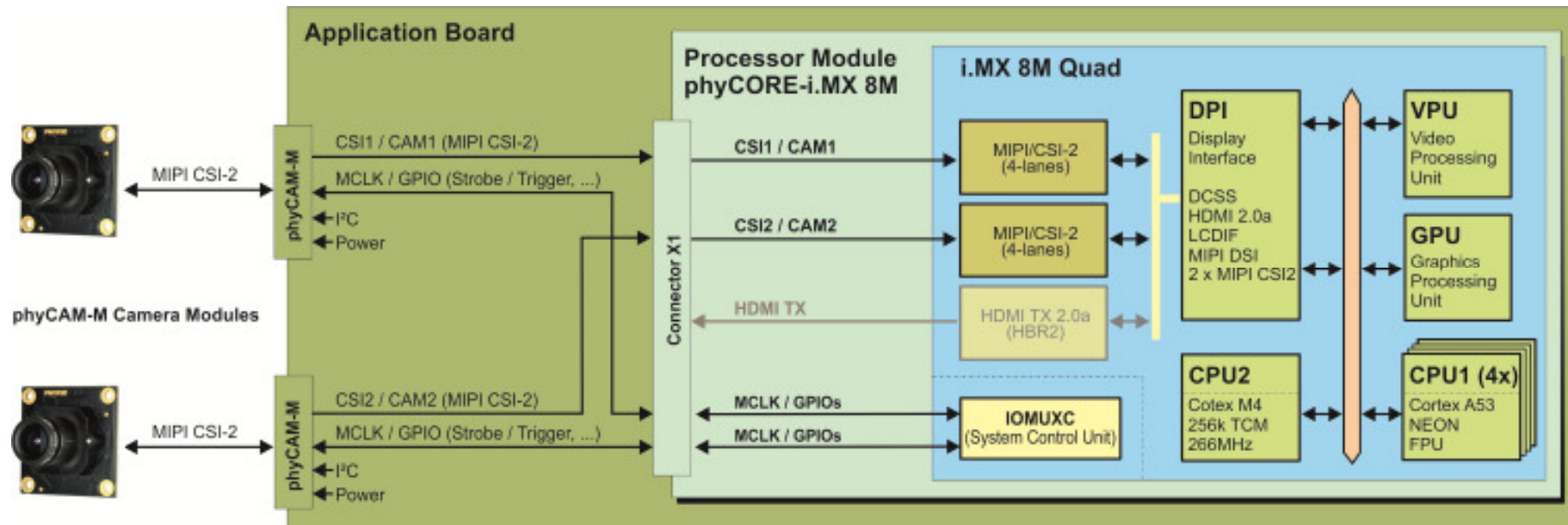


figure 3: Block Diagram of phyCAM-M Camera Interfaces of phyCORE-i.MX 8M (Quad) and the go out on the phyBOARD-Polaris-i.MX 8M - SBC

The BSP shipped with the Kit includes already the software drivers for the supported phyCAM-M camera modules. The drivers are compatible with v4l2. Also GStreamer scripts are included for the evaluation of the camera modules. If you need the camera interface to connecting your own camera module, is an adapter to phyCAM-M necessary.

Note that the both camera channels are independent. Therefore it is possible to work with 2 cameras parallel.

2 Camera Connectors on the Polaris - Carrier Boards

The development kits for the phyBOARD-Polaris-i.MX 8M contain:

- one carrier board (Polaris)
- one phyCORE-i.MX 8M modul SOM

The phyCORE-i.MX 8M is direct soldering on carrier board.

On the base board Polaris (PBA-CD-12) we convert the both MIPI CSI-2 interfaces in phyCAM-M standard.

- phyCAM-M camera interface 1 (support on connector X10) / phyCAM-M camera interface 2 (support on connector X11)

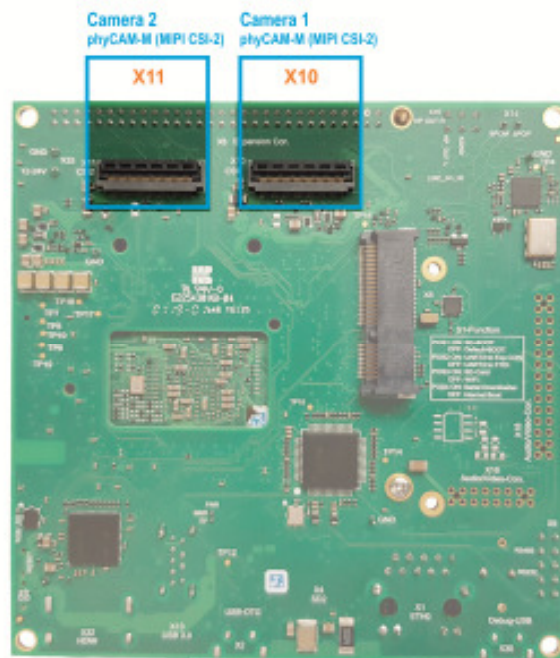


figure 4: Camera Interfaces on phyBOARD-Polaris up PCB Version PL1501.1 for the phyCORE-i.MX 8M SOM

Note: If you use the phyCAM-M interface, use an FFC cable that is especially suitable for FH41 connectors (e.g. Phytex WF271). Standard FFC cables can cause a short circuit.

3 Supported Camera Boards (YOCTO Linux PD19.1.0 and higher)

The cameras and the camera interface supported as a sub-devices. To configure the v4l2 framework is the handler "v4l2-ctl" used.

For more information see on FTP:

ftp://ftp.phytec.de/pub/ImageProcessing/phyBOARD-Polaris-i.MX8M_linux_PD19.1.0/

In the PD19.1.0 are only the camera VM-016 (sensor AR0144) is supported. For use other cameras a cameras specific device tree is necessary.

Supported camera types. Changing the setting of the config-expansions parameters is not necessary.

- VM-016-BW-M series (VM-016-BW-M, VM-016-BW-M-M12, VM-016-BW-M-H, ...) based on camera sensor "AR0144"
- VM-016-COL-M series (VM-016-COL-M, VM-016-COL-M-M12, VM-016-COL-M-H, ...) based on camera sensor "AR0144"

After login, change into the directory: `\gststreamer_examples\.`
`cd gststreamer_examples <ENTER>`.

Now you can start working with the GStreamer demo-scripts. (see chapter 5.1)

Hardware Configuration		Possible Parameters				default I ² C-address (default jumper on camera and boards)
phyCAM camera model (part number)	connected to	csi_port	cam_bus_type	cam_type	cam_i2c_address	
VM-016-BW-M (-M12 / -H)	X10 on Polaris board	1	phyCAM-M	VM-016	0x10, 0x18	0x10
	X11 on Polaris board	2	phyCAM-M	VM-016	0x10, 0x18	0x18
VM-016-COL-M (-M12 / -H)	X10 on Polaris board	1	phyCAM-M	VM-016	0x10, 0x18	0x10
	X11 on Polaris board	2	phyCAM-M	VM-016	0x10, 0x18	0x18

Notes:

- I²C addresses of the camera are set by hardware configuration (jumper setting on the camera and / or on the baseboard. Please refer to the phyCAM-manual L-748 and the hardware manual of the kit.
- If two cameras are used at the same time, csi[X]_cam address must be different for CAM1 and CAM2.

4 Supported Camera Boards (YOCTO Linux PD21.1.0 and higher)

The cameras and the camera interface are supported as sub-devices. To configure the v4l2 framework is the handler "v4l2-ctl" used.

4.1 Supported Camera Types in Standard Vision Image (phytec-vision-image-phyboard-polaris-imx8m-3)

Per default the following Camera types at X10 are supported in the image:

- VM-016-BW-M series (VM-016-BW-M, VM-016-BW-M-M12, VM-016-BW-M-H, ...) based on camera sensor "AR0144"
- VM-016-COL-M series (VM-016-COL-M, VM-016-COL-M-M12, VM-016-COL-M-H, ...) based on camera sensor "AR0144"

After login, change into the directory: `\gstreamer_examples\.`
`cd gstreamer_examples <ENTER>`.

Now you can start working with the GStreamer demo-scripts. (see chapter 5.1)

Hardware Configuration		Possible Parameters			default I ² C-address (default jumper on camera and boards)
phyCAM camera model (part number)	connected to	csi_port	cam_bus_type	cam_i2c_address	
VM-016-__-M (-M12 / -H)	X10 on Polaris board	1	phyCAM-M	0x10, 0x18	0x10

Notes:

- I²C addresses of the camera are set by hardware configuration (jumper setting on the camera and / or on the baseboard. Please refer to the phyCAM-manual the [L-1018e.A0 phyCAM-M VM-016 1/4" 1,0 MPixel Global Shutter Camera Module Guide](#) and the hardware manual of the kit.

4.2 Change the Camera Types or Connected Port in Standard Vision Image (phytec-vision-image-phyboard-polaris-imx8m-3)

The following camera types are interchangeable in the image:

- VM-016-BW-M series (VM-016-BW-M, VM-016-BW-M-M12, VM-016-BW-M-H, ...) based on camera sensor "AR0144"
- VM-016-COL-M series (VM-016-COL-M, VM-016-COL-M-M12, VM-016-COL-M-H, ...) based on camera sensor "AR0144"
- VM-116-BW-M series (VM-116-BW-M, VM-116-BW-M-M12, ...) based on camera sensor "AR0144"
- VM-116-COL-M series (VM-116-COL-M, VM-116-COL-M-M12, ...) based on camera sensor "AR0144"

To register the cameras you need an adapted device tree.

4.2.1 How to Change the Device Tree

- During the boot sequence, press a key (use a terminal program e.g. "Putty") to arrive the bootloader
- u-boot=> edit fdt_file
- Edit: <type here your device tree> (e.g. "oftree")

PHYTEC supports same device trees, see the right device tree in following list:

Hardware Configuration		Possible Parameters			default I ² C-address (default jumper on camera and boards)	Device Tree
phyCAM camera model (part number)	connected to	csi_port	cam_bus_type	cam_i2c_address		Name of device tree
VM-016-__-M (-M12 / -H)	X10 on Polaris	1	phyCAM-M	0x10, 0x18	0x10	oftree
VM-016-__-M (-M12 / -H)	X11 on Polaris	2	phyCAM-M	0x10, 0x18	0x10	imx8mq-phyboard-polaris-rdk-emmc-vm016-csi2-no18vsdcard.dtb
VM-016-__-M (-M12 / -H) and VM-016-__-M (-M12 / -H)	X10 on Polaris	1	phyCAM-M	0x10, 0x18	0x10	imx8mq-phyboard-polaris-rdk-emmc-dual-vm016-no18vsdcard.dtb
	X11 on Polaris	2	phyCAM-M	0x10, 0x18	0x10	

5 Demo Scripts

If you want to see the live images, we recommend using a HDMI monitor on the Pollis Board.

There are 3 subdirectories with demo scripts for the cameras:

- gstreamer-examples
- v4l2_c-examples
- opencv-examples

Note: Remove the *qt-demo* at first startup with the `\gstreamer-examples\remove_qt_demo.sh` script. Otherwise, the *qt-demo* is always present on the display.

5.1 GStreamer Scripts

After login, change into the directory: `\gstreamer-examples\.`
`cd gstreamer_examples <ENTER>`.

First deactivate the wayland windowmanager. “`./turn_off_on_wayland.sh`”

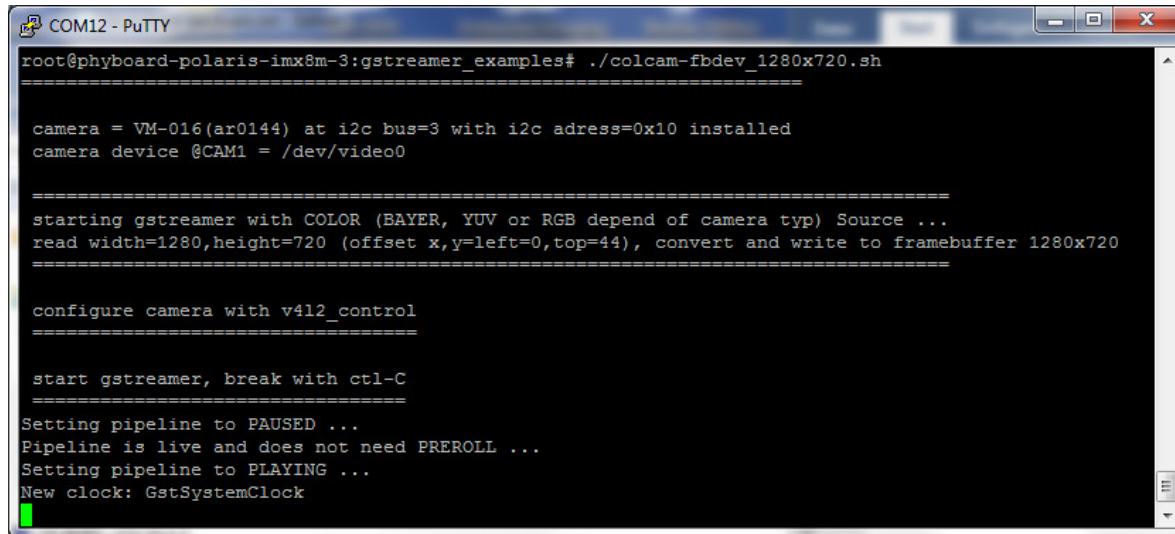
Now you can start working with the GStreamer demo scripts.

Start the scripts with the word phrase "col" or "bw" depending on the connected camera color type.

- `_cam-fbdev_1280x720.sh`: scripts show a live stream on display
- `_cam-save_jpg_full_res.sh`: scripts save a JPG File in this directory
- `_cam-save_raw_full_res.sh`: scripts save a RAW File in this directory
- `sh`: scripts detect the camera type and define the parameter for the scripts
- `sh`: scripts remove the qt-demo from autostart
- `sh`: scripts turn on/off if Wayland necessary or not

Subdirectories:

- *more_ar0144_scripts*: contain more scripts for the VM-016 camera series (v4l-ctrl_ar0144.txt list the v4l2-controls for this camera)
- *tools*: contain scripts to get and set the camera register direct via i2c access
- *phytec_usb_cam*: scripts for use the PHYTEC USB-cameras



```
COM12 - PuTTY
root@phyboard-polaris-imx8m-3:gstreamer_examples# ./colcam-fbdev_1280x720.sh
=====
camera = VM-016(ar0144) at i2c bus=3 with i2c adress=0x10 installed
camera device @CAM1 = /dev/video0
=====
starting gstreamer with COLOR (BAYER, YUV or RGB depend of camera typ) Source ...
read width=1280,height=720 (offset x,y=left=0,top=44), convert and write to framebuffer 1280x720
=====
configure camera with v4l2_control
=====
start gstreamer, break with ctl-C
=====
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

Call of “colcam-fbdev_1280x720.sh” for VM-016-COL (AR0144) series (A live image should show on the display)

All camera/video components get a separate “dev/video[x]” device.

The v4l2 capabilities are shown if you type: “v4l2-ctl -d [device] -L” e.g. “v4l2-ctl -d /dev/video0 -L”.

Note: For the first use, the camera and controller-camera interface must be configured with the tool v4l2-ctl.

5.2 Scripts to Call C/C++ files Based on v4l2 Interface

After login, change into the directory: `\v4l2_c-examples\.`
`cd v4l2_c-examples <ENTER>`.

Now you can start working with the demo scripts.

Start the scripts with the word phrase "col" or "bw" depending on the connected camera color type.

- `ar0144_col/bw_full_save-raw`: scripts save a raw image from VM-016 in full resolution (8-Bit Formats are possible)

To save the image, we use the program *Yavta*. *Yavta* stands for "Yet Another V4L2 Test Application". This is a test application based on V4L2 Linux interface.

Another way to use in your C-program is the direct access call to the v4l2 interface. For example:

```
v4l2-ctl -d0 --stream-mmap --stream-count 1 --stream-to=raw_image1.raw
```

Note: For the first use, the camera and controller-camera interface must be configured with the tool *v4l2-ctl*.

5.3 OpenCV Scripts

PHYTEC i.MX 8M BSP PD19.1.0 includes OpenCV3.3. For use OpenCV with image output we recommend windows manager Wayland. To turn/off/on wayland use the script in the path "`.../gststreamer_examples/turn_off_on_wayland.sh`"

PHYTEC tested openCV with pyCAM cameras and the windows manager Wayland. Examples of scripts can be found on our ftp:

- ftp://ftp.phytec.de/pub/ImageProcessing/phyBOARD-Polaris-i.MX8M_linux_PD19.1.0/Software/OpenCV/

If you use the PHYTEC phyCAM - camera modules the camera and the camera interface driver have to configure with the v4l2-ctl tool. Examples are in the scripts at upper link.

PHYTEC i.MX 8M BSP PD21.1.0 includes OpenCV4.4. To use OpenCV with image output, we use window manager Wayland. To turn/off/on wayland use the script in the path "`.../gststreamer_examples/turn_off_on_wayland.sh`". PHYTEC tested OpenCV with PYTHON programming language. Examples of scripts can be found in the path "opencv-examples":

- `/opencv-examples/python/python3 gst_phycam_video_v4l2.py`: show the live image from the camera

Note: For the first use, the camera and controller-camera interface must be configured with the tool *v4l2-ctl*.

6 De-Bayering (demosaicking) with NEON CoProcessor

Most of CMOS color chips provide the image in the bayer mosaicing (bayer raw) format. For get a color image in RGB format is it necessary to convert the bayer raw image.

- https://en.wikipedia.org/wiki/Bayer_filter
- <https://de.wikipedia.org/wiki/Bayer-Sensor>

There are exist different algorithm for converting.

- <https://en.wikipedia.org/wiki/Demosaicing>

If the microprocessor does not include debayering hardware, have to do the converting via software. For this you need additional processing power and the framerate goes down. It is better to use the NEON coprocessor of the i.MX 8M. For this support PHYTEC a special function.

It is present as GStreamer plugin "bayer2rgbneon" and in sources for use in an own C-program. We support a simple bilinear algorithm.

For use in GStreamer take "bayer2rgbneon" plugin. For more information to "bayer2rgbneon" parameters type:

- gst-inspect-1.0 bayer2rgbneon

The source are located at:

- <https://git.phytec.de/bayer2rgb-neon/>